THE DATASHEET OF

89H32NT8AG2ZCHLG8

www.win-source.net 0086-755-83957316

IDT® 89HPES32NT8xG2 PCI Express® Switch

User Manual

January 2013

6024 Silver Creek Valley Road, San Jose, California 95138
Telephone: (800) 345-7015 • (408) 284-8200 • FAX: (408) 284-2775
Printed in U.S.A.
©2013 Integrated Device Technology, Inc.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance and to supply the best possible product. IDT does not assume any responsibility for use of any circuitry described other than the circuitry embodied in an IDT product. The Company makes no representations that circuitry described herein is free from patent infringement or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, patent rights or other rights, of Integrated Device Technology, Inc.

CODE DISCLAIMER

Code examples provided by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of the code examples below is completely at your own risk. IDT MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE NONINFRINGEMENT, QUALITY, SAFETY OR SUITABILITY OF THE CODE, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FURTHER, IDT MAKES NO REPRESENTATIONS OR WARRANTIES AS TO THE TRUTH, ACCURACY OR COMPLETENESS OF ANY STATEMENTS, INFORMATION OR MATERIALS CONCERNING CODE EXAMPLES CONTAINED IN ANY IDT PUBLICATION OR PUBLIC DISCLOSURE OR THAT IS CONTAINED ON ANY IDT INTERNET SITE. IN NO EVENT WILL IDT BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE OR SPECIAL DAMAGES, HOWEVER THEY MAY ARISE, AND EVEN IF IDT HAS BEEN PREVIOUSLY ADVISED ABOUT THE POSSIBILITY OF SUCH DAMAGES. The code examples also may be subject to United States export control laws and may be subject to the export or import laws of other countries and it is your responsibility to comply with any applicable laws or regulations.

LIFE SUPPORT POLICY

Integrated Device Technology's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of IDT.

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any components of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

IDT, the IDT logo, and Integrated Device Technology are trademarks or registered trademarks of Integrated Device Technology, Inc.

About This Manual

Notes

Overview

This user manual includes hardware and software information on the 89HPES32NT8xG2, a member of IDT's PRECISE™ family of PCI Express® switching solutions offering the next-generation I/O interconnect standard. The part number PES32NT8xG2 which is used extensively throughout this manual in fact covers two distinct switch devices: the PES32NT8AG2 and the PES32NT8BG2. The information in this manual applies equally to both devices except where noted in occasional notes and footnotes in various chapters.

Finding Additional Information

Information not included in this manual such as mechanicals, package pin-outs, and electrical characteristics can be found in the data sheet for this device, which is available from the IDT website **(www.idt.com)** as well as through your local IDT sales representative.

Content Summary

Chapter 1, "PES32NT8xG2 Device Overview," provides an introduction to the performance capabilities of the 89HPES32NT8xG2 and a high level architectural overview of the device.

Chapter 2, "Clocking," provides a description of the PES32NT8xG2 clocking architecture.

Chapter 3, "Reset and Initialization," describes the PES32NT8xG2 reset operations and initialization procedure.

Chapter 4, "Switch Core," provides a description of the PES32NT8xG2 switch core.

Chapter 5, "Switch Partitions," describes how the PES32NT8xG2 supports up to 16 active switch partitions.

Chapter 6, "Failover," provides a description of the flexible failover mechanism that allows the construction of highly-available systems.

Chapter 7, "Link Operation," describes the operation of the link feature including polarity inversion, link width negotiation, and lane reversal.

Chapter 8, "SerDes," describes basic functionality and controllability associated with the Serializer-Deserializer (SerDes) block in PES32NT8xG2 ports.

Chapter 9, "Power Management," describes the power management capability structure located in the configuration space of each PCI-to-PCI bridge in the PES32NT8xG2.

Chapter 10, "Transparent Operation," describes the device-specific architectural features for the transparent switch associated with each PES32NT8xG2 partition (i.e., the PCI-to-PCI bridge functions and their interaction in the switch).

Chapter 11, "Hot-Plug and Hot-Swap," describes the behavior of the hot-plug and hot-swap features in the PES32NT8xG2.

Chapter 12, "SMBus Interfaces," describes the operation of the 2 SMBus interfaces on the PES32NT8xG2.

Chapter 13, "General Purpose I/O," describes how the 9 General Purpose I/O (GPIO) pins may be individually configured as general purpose inputs, general purpose outputs, or alternate functions.

Chapter 14, "Non-Transparent Operation," describes how a non-transparent bridge in the PES32NT8xG2 allows two roots or PCI Express trees (i.e., hierarchies) to be interconnected with one or more shared address windows between them.

Notes

Chapter 15, "DMA Controller," describes how the PES32NT8xG2 supports two direct memory access controller (DMA) functions.

Chapter 16, "Switch Events," describes mechanisms provided by the PES32NT8xG2 to facilitate communication between roots associated with different partitions as well as for communication between these roots and a management agent.

Chapter 17, "Multicast," describes how the multicast capability enables a single TLP to be forwarded to multiple destinations.

Chapter 18, "Temperature Sensor," provides a description of the on-chip temperature sensor with three programmable temperature thresholds and a temperature history capability.

Chapter 19, "Register Organization," describes the organization of all the software visible registers in the PES32NT8xG2 and provides the address space for those registers.

Chapter 20, "PCI to PCI Bridge and Proprietary Port Specific Registers," lists the Type 1 configuration header registers in the PES32NT8xG2 and provides a description of each bit in those registers.

Chapter 21, "Proprietary Registers," lists the proprietary registers in the PES32NT8xG2 and provides a description of each bit in those registers.

Chapter 22, "NT Endpoint Registers," lists the NT Endpoint registers in the PES32NT8xG2 and provides a description of each bit in those registers.

Chapter 23, "DMA Registers," lists the DMA registers in the PES32NT8xG2 and provides a description of each bit in those registers.

Chapter 24, "Switch Control Registers," lists the switch control and status registers in the PES32NT8xG2 and provides a description of each bit in those registers.

Chapter 25, "JTAG Boundary Scan," discusses an enhanced JTAG interface, including a system logic TAP controller, signal definitions, a test data register, an instruction register, and usage considerations.

Chapter 26, "Usage Models," describes possible configurations of the PES32NT8xG2 switch and presents some important system usage models.

Signal Nomenclature

To avoid confusion when dealing with a mixture of "active-low" and "active-high" signals, the terms assertion and negation are used. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

To define the active polarity of a signal, a suffix will be used. Signals ending with an 'N' should be interpreted as being active, or asserted, when at a logic zero (low) level. All other signals (including clocks, buses and select lines) will be interpreted as being active, or asserted when at a logic one (high) level.

To define buses, the most significant bit (MSB) will be on the left and least significant bit (LSB) will be on the right. No leading zeros will be included.

Throughout this manual, when describing signal transitions, the following terminology is used. Rising edge indicates a low-to-high (0 to 1) transition. Falling edge indicates a high-to-low (1 to 0) transition. These terms are illustrated in Figure 1.

Notes

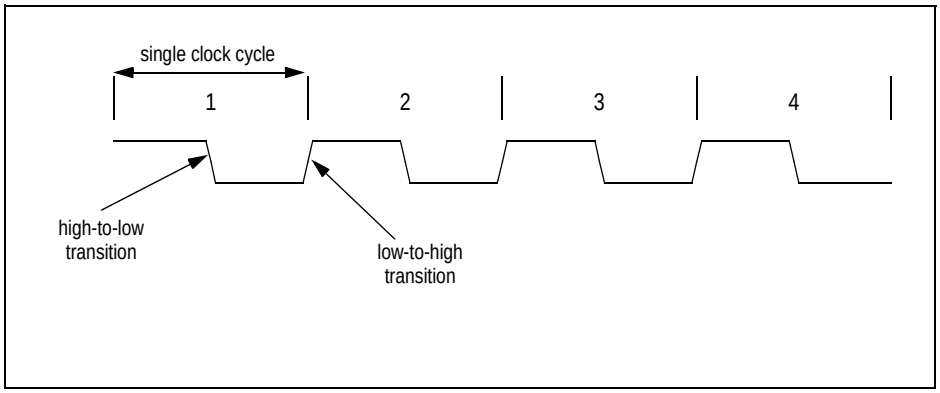

Figure 1 Signal Transitions

Numeric Representations

To represent numerical values, either decimal, binary, or hexadecimal formats will be used. The binary format is as follows: 0bDDD, where “D” represents either 0 or 1; the hexadecimal format is as follows: 0xDD, where “D” represents the hexadecimal digit(s); otherwise, it is decimal.

The compressed notation ABC[x|y|z]D refers to ABCxD, ABCyD, and ABCzD.

The compressed notation ABC[y:x]D refers to ABCxD, ABC(x+1)D, ABC(x+2)D,... ABCyD.

Data Units

The following data unit terminology is used in this document.

Term	Words	Bytes	Bits
Byte	1/2	1	8
Word	1	2	16
Doubleword (DWord)	2	4	32
Quadword (QWord)	4	8	64

Table 1 Data Unit Terminology

In quadwords, bit 63 is always the most significant bit and bit 0 is the least significant bit. In doublewords, bit 31 is always the most significant bit and bit 0 is the least significant bit. In words, bit 15 is always the most significant bit and bit 0 is the least significant bit. In bytes, bit 7 is always the most significant bit and bit 0 is the least significant bit.

The ordering of bytes within words is referred to as either “big endian” or “little endian.” Big endian systems label byte zero as the most significant (leftmost) byte of a word. Little endian systems label byte zero as the least significant (rightmost) byte of a word. See Figure 2.

Notes

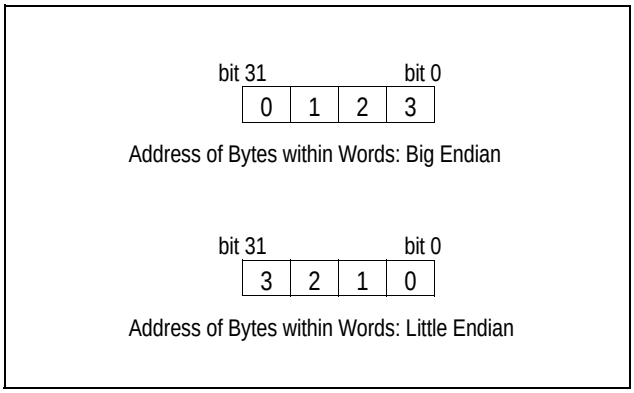

Figure 2 Example of Byte Ordering for "Big Endian" or "Little Endian" System Definition

Register Terminology

Software in the context of this register terminology refers to modifications made by PCI Express root configuration writes, writes to registers made through the slave SMBus interface, or serial EEPROM register initialization. See Table 2.

Type	Abbreviation	Description
Hardware Initialized	HWINIT	Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. (System firmware hardware initialization is only allowed for system integrated devices.) Bits are read-only after initialization and can only be reset (for write-once by firmware) with reset.
Read Only and Clear	RC	Software can read the register/bits with this attribute. Reading the value will automatically cause the register/bit to be reset to zero. Writing to a RC location has no effect.
Read Clear and Write	RCW	Software can read the register/bits with this attribute. Reading the value will automatically cause the register/bits to be reset to zero. Writes cause the register/bits to be modified.
Reserved	Reserved	The value read from a reserved register/bit is undefined. Thus, software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. In addition to reserved registers, some valid register fields have encodings marked as reserved. Such register fields must never be written with a value corresponding to an encoding marked as reserved. Violating this rule produces undefined operation in the device.
Read Only	RO	Software can only read registers/bits with this attribute. Contents are hardwired to a constant value or are status bits that may be set and cleared by hardware. Writing to a RO location has no effect.

Table 2 Register Terminology (Part 1 of 2)

Notes

Type	Abbreviation	Description
Read and Write	RW	Software can both read and write bits with this attribute.
Read and Write Clear	RW1C	Software can read and write to registers/bits with this attribute. However, writing a value of zero to a bit with this attribute has no effect. A RW1C bit can only be set to a value of 1 by a hardware event. To clear a RW1C bit (i.e., change its value to zero) a value of one must be written to the location. An RW1C bit is never cleared by hardware.
Read and Write when Unlocked		Software can read the register/bits with this attribute. Writing to register/bits with this attribute will only cause the value to be modified if the REGUNLOCK bit in the SWCTL register is set. When the REGUNLOCK bit is cleared, writes are ignored and the register/bits are effectively read-only.
Sticky	Sticky	Register/bits with this designation take on their initial value as a result of a switch fundamental reset or partition fundamental reset. Other resets have no effect.
Switch Sticky	SWSticky	Register/bits with this designation take on their default value as a result of a switch fundamental reset. Other resets have no effect.
Modified Switch Sticky	MSWSticky	A MSWSticky register is a Switch Sticky register that in addition to taking on its default value as a result of a switch fundamental reset, it takes on its default value when the event(s) defined in the register description occur, unless the register has been written-to by software/firmware before the occurrence of the event. If the value of an MSWSticky register has been written by software/firmware, it preserves the value across all events until written again or until a switch fundamental reset is applied to the device. After a switch fundamental reset, the MSWSticky register will return to taking on the value as defined in the register description.

Table 2 Register Terminology (Part 2 of 2)

Use of Hypertext

In Chapter 19, Tables 19.2, 19.5, 19.6, 19.10, and 19.11 contain register names and page numbers highlighted in blue under the Register Definition column. In pdf files, users can jump from this source table directly to the registers by clicking on the register name in the source table. Each register name in the table is linked directly to the appropriate register in Chapters 20 through 24. To return to the source table after having jumped to the register section, click on the same register name (in blue) in the register section.

Reference Documents

PCI Express Base Specification Revision 2.1., March 4, 2009, PCI-SIG.

PCI Local Bus Specification Revision 3.0., February 3, 2004, PCI-SIG.

PCI-to-PCI Bridge Architecture Specification Revision 1.2., June 9, 2003, PCI-SIG.

Address Translation Services Specification, March 8, 2007, PCI-SIG.

PCI Bus Power Management Interface Specification, Revision 1.2., March 3, 2004, PCI-SIG

SMBus Specification, Version 2.0, August 3, 2000, SBS Implementers Forum.

Revision History

July 8, 2009: Initial publication of preliminary user manual.

Notes

July 14, 2009: Includes changes in several chapters based on recent updates in the functional specification.

July 30, 2009: Includes changes in several chapters based on recent updates in the functional specification.

August 28, 2009: Added Chapter 27, Usage Models.

September 18, 2009: In Chapter 1, added explanation of differences between PES32NT8AG2 and PES32NT8BG2 devices. In Chapter 2, added Table 2.5. In Chapter 4, added new sections Packet Routing Classes and Proprietary Weighted Round Robin (WRR) Arbitration, and revised Figure 4.8. Made numerous revisions in Chapter 8. In Chapter 10, made changes to the Action Taken column in Table 10.14. In Chapter 12, updated the I/O Expander tables. In Chapter 15, made changes to Table 15.7 and added text to DMA Multicast section. In Chapter 25, made numerous changes in SerDes x Transmitter Lane Control 0 and 1 registers.

October 6, 2009: In Chapter 3, added text to section Switch Fundamental Rest and moved this section in front of Boot Configuration Vector section, and added text to Switch Modes section. In Chapter 5, added text to sections Switch Partitioning and Non-Transparent Operations. In Chapter 15, modified sections Data Transfer and Addressing, Source Address Expired Error, and Destination Address Expired Error. In Chapter 16, added text to section Switch Signals. In Chapter 21, modified description for bit EIS in the PCIESSTS register. In Chapter 24, modified description of RUN bit in the DMAC[1:0]CTL register.

October 13, 2009: In Chapter 8, modified Tables 8.1 through 8.4 and modified text in the examples used in section Programmable Voltage Margining and De-Emphasis.

October 27, 2009: In Chapter 12, modified Table 12.13 to clarify slave SMBus addresses for both the PES32NT8AG2 and PES32NT8BG2 devices.

November 4, 2009: In Chapter 2, added new section Support for Spread Spectrum Clocking (SCC) with updated tables and modified Limitations column in Table 2.6, Clock Frequency Limitations. In Chapter 5, added three new sections: Partition State Change Latency, Port Operating Mode Change Latency, and Partition Reconfiguration Latency. In Chapter 8, deleted all references to Slew Rate. In Chapter 10, title for Table 10.11 was changed to Unexpected Completions instead of Unsupported Requests, and a new bullet was added at the top of section Address Routed TLPs. In Chapter 14, modified text in Overview section and in section Unsupported Request (UR) Error. In Chapter 15, modified text in section Reception of a Request TLP That is Unsupported. In Chapter 19, added reference to junction temperature in the Overview section. In Chapter 20, added new section Configuration Register Side-Effects under Overview. In Chapter 24, DMA Channel Error Mask register, de-featured bits 2 and 17. In Chapter 25, SMBus Control register, de-featured bit 16 (MSMBIOM).

November 23, 2009: In Chapter 4, corrected port numbers for Stack 1 in Figure 4.1. In Chapter 7, modified text in Port Merging section to clarify merging process.

December 4, 2009: In Chapter 10, modified text in section Error Emulation Control in the PCI-to-PCI Bridge Function and added new section Error Emulation Usage and Limitations. In Chapter 14, modified text in section Error Emulation Control in the NT Function and added new section Error Emulation Usage and Limitations.

January 6, 2010: In Chapter 17, corrected references to NT Multicast Control register and NT Multicast Transmit Enable bit in the following sections: NT Multicast TLP Routing, and Usage Restrictions. In Chapter 20, corrected Figure 20.5. In Chapter 23, NTIERRORMSK0 register, changed bits 29 and 30 to reserved. In Chapter 24, DMAIERRORMSK0 register, changed bits 29 and 30 to reserved. In Chapter 25, changed default values for several bits in the TMPADJA and TSSLOPE registers.

January 12, 2010: In the Stack Configuration section of Chapter 3, corrected reference to possible configurations to say either one x8 or two x4 ports. No other configuration is possible. In section I/O Expanders of Chapter 12, changed number of I/O Expanders supported from 22 to 10; in Table 12.4, removed I/O Expanders 4 through 11; in Tables 12.5 and 12.6, changed title to read “I/O Expanders 0 to 3” instead of 0 to 11.

Notes

March 8, 2010: Removed references to OUTDBELLCLR and OUTSBELLDBELLCLR registers in Chapter 14, Non-Transparent Switch Operation.

March 17, 2010: In Chapter 8, updated Tables 8.6, 8.7, 8.8, 8.11, and 8.12. In Chapter 13, deleted reference to multiple GPIOAFSEL registers; there is only one register. In Chapter 24, deleted “other” from ECRC Error name for bit 31 in the DMAC[1:0]ERRSTS register.

May 10, 2010: In Chapter 21, PCI Bridge Registers, the ACSCAP register offset address was corrected to 0x324. In Chapter 23, NT Endpoint Register, revised the Description for MODE field in BARSETUP0 register and LADDR field in BARLIMIT0 register.

May 21, 2010: In Chapter 23, NT Endpoint Registers, revised Description for INDEX field in LUTOFFSET register to read that if BAR4 is selected, the INDEX field must only be set to values 0 to 11 (instead of 12 to 23).

June 21, 2010: In Chapter 23, NT Endpoint Registers, revised Bit Field column in NTMTBLDATA register.

August 27, 2010: In Chapter 4, revised text in sections Internal Errors and Reporting of Port AER Errors as Internal Errors and updated Figures 4.7 and 4.8. In Chapter 5, revised text in Reset Mode Change Behavior. In Chapter 7, revised text in Link Width Negotiation in the Presence of Bad Lanes section and Crosslink section. In Chapter 11, corrected reference to DLLLASC in Hot Plug Events section. In Chapter 12, revised description for BYTECNT in Tables 12.19 and 12.21. In Chapter 14, added Note at end of section NT Mapping Table. In Chapter 15, deleted section DMA Channel Errors and revised text in Descriptor Prefetching, ECRC Errors, and Completion Timeout sections. In Chapter 16, revised text in section Port AER Errors. In Chapter 17, changed reference from NTMTC to NTMCC in NT Multicast TLP Routing section.

In Chapter 21, made the following changes: revised description for MAXLNKSPD bit in PCI Express Link Capabilities register (also applies to same bit in same register in Chapters 23 and 24), revised description for bits in PCI Express Slot Control register.

In Chapter 22, made the following changes: added text under section Physical Layer Control and Status Registers, revised description for bits in PCI Express Slot Control Initial Value register, deleted Port AER Status register, revised Port AER Mask register, added bit 10 to bit 9 as Reserved and revised description for ILSCC bit in Phy Link Configuration 0 register.

In Chapter 23, made the following changes: revised PCI Express Device Capabilities 2 and PCI Express Device Control 2 registers, revised Description for REG and EREG bits in ECFGADDR register, added bits 31:16 row in AER Correctable Error Status register, added text in Description of NXTPTR in SNUMCAP register, added text in Description of NXTPTR in PCIEVCECAP register, revised information for fields PARBC and PATBLOFF in VCR0CAP register, revised information for fields LPAT and PARBSEL in VCR0CTL register, revised Description for PATS in VCR0STS register, added text in Description of NXTPTR in ACSECAPH register, added text in Description of NXTPTR in MCCAPH register, changed default value for bits 30:29 from 0x1 to 0x3 in NTIERRORMSK0 register, changed bit 6 in the NTINTMSK register to Reserved, revised description for bits SIZE and MODE in Bar 0 Setup register, revised information in LADDR field in BARLIMIT0 register, revised text under register title for BAR 1 Limit Address and changed Default Value for Reserved and LADDR and Description for LADDR, revised text under register title for BARLIMIT3 and changed Default Value for Reserved and LADDR and Description for LADDR, revised Default Value and Description for LADDR field in BARLIMIT4 register, revised text under register title for BARLIMIT5 and changed Default Value and Description for LADDR, revised description for INDEX in LUTOFFSET register.

In Chapter 24, made the following changes: revised bits 4 to 21 in PCIEDCAP2 register, revised bits 4 to 15 in PCIEDCTL2 register, revised bits 21 and 22 and added bits 24 and 25 in AERUES register, revised bit 22 and added bits 24 and 25 in AERUEM register, revised bit 22 and added bits 24 and 25 in AERUESV register, changed Default Value for CIE bit in AERCES register, changed Type and Default Value for ECRCGC and ECRCCC bits in AERCTL register, changed bit 24 (HEC) to reserved in DMAIERRORMSK1 register, de-featured bits 0 and 6 through 9 in the DMAC[1:0]ERRSTS register, de-featured bits 0 and 6 through 7 and changed name of bit 31 to ECRCE in DMAC[1:0]ERRMSK register.

Notes

In Chapter 25, made the following changes: revised SESTS register, revised description for COUNT field in FCAP[3:0]TIMER register, added bits 20 and 21 and revised default value and/or description for bits 22 to 25 and changed name/value/description of bit 29 in SMBUSSTS register, removed default value for TEMP field in TMPSTS register.

September 27, 2010: In Chapter 22, changed bit 16 in the IERRORSTS0 register from ULD to Reserved.

October 22, 2010: In Chapter 15, added footnote to Table 15.7. In Chapter 25, re-arranged bits 24:28 in TMPCTL register.

December 21, 2010: In Chapter 2, revised header in Table 2.5 to read "Initial Port Clock Mode." In Chapter 5, added new footnote #1 in section Port Operating Mode Change. In Chapter 15, deleted reference to DATCT bit in Completion Timeout section. In Chapter 23, added text to SUBVID and SUBID registers. In Chapter 24, changed bit 20 (DATCT bit) in the DMAC[1:0]ERRSTS and DMAC[1:0]ERRMSK registers to Reserved and added text to SUBVID and SUBID registers. In Chapter 26, deleted PERSTN, GLK1, and SMODE from Table 26.1.

March 11, 2011: In Chapter 26, revised Usage Considerations section to remove reference to JTAG_TCK being driven to a known value.

March 25, 2011: In Chapter 22, added PHYLSTATE0 register with FLRET bit description.

May 20, 2011: In Chapter 1, added ZC silicon to Table 1.2.

June 21, 2011: In Chapter 5, section Reset Mode Change Behavior, changed fourth bullet to read "The port remains in a Reset state for at least 250 µs."

June 24, 2011: In Chapter 25, added bit BDISCARD to the Switch Control register.

July 15, 2011: In Chapter 1, revised section Switch Events and removed "and Signals" from the section title. In Chapter 5, revised the following sections: Downstream Switch Port, Port Operating Mode Change Latency, and System Notification of Partition Reconfiguration. In Chapter 8, revised section Programmable De-emphasis Adjustment. In Chapter 16, removed "and Signals" from title and revised section Global Signals and deleted Signals section. In Chapter 21, MCBLKALLH register, changed lower 32 to upper 32 in description of MCBLKALL bit. In Chapters 22 and 23, deleted references to SSIGNAL field. In Chapter 25, added section Internal Switch Timers with 4 new registers and deleted SSIGNAL register. Updated Figure 20.5 and Table 20.11, Switch Configuration and Status, in Chapter 20 to account for new registers.

July 27, 2011: In Chapter 22, added bits 7:0 (RCVD_OVRD) in SERDESCFG register.

August 23, 2011: In Chapter 24, DMACxCFG register, changed 0x2 in DPREFETCH field to Reserved.

September 12, 2011: In Chapter 8, added additional reference in last paragraph of section Driver Voltage Level and Amplitude Boost.

October 24, 2011: In Chapter 22, added Port Control Register. In Chapter 20, added reference to Port Control register in Table 20.5.

November 7, 2011: In Chapter 2, section Local Port Clocked Mode, added recommendation to tie unused port clock pins to ground.

December 4, 2011: In Chapter 25, revised Description for AFSEL0 field in the GPIOAFSEL register.

January 11, 2012: Removed Hardware Error Containment chapter. Deleted references to SWFRST bit.

February 8, 2012: In Chapter 12, added footnote for RERR and WERR bits in Table 12.16.

February 23, 2012: Added paragraph after Table 12.19 to explain use of DWord addresses.

March 14, 2012: In Chapter 2, changed "single" to "two" differential global reference clock pairs.

May 1, 2012: In Chapter 2, Clocking, made text changes to state that unused port clock pins should be connected to Vss on the board. In Chapter 12, SMBus Interfaces, added new section Setting Up I2C Commands for Block Transactions.

Notes

June 27, 2012: In Chapter 12, changed BYTCNT=7 to BYTCNT=4 in Figure 12.14. In Chapter 24, changed type and default values for bits 16 and 20 in Switch Control register.

January 30, 2013: In Figure 12.12, changed No-ack to Ack between DATALM and DATAUM.

Notes

Table of Contents

Notes

About This Manual

Overview 1
Content Summary 1
Signal Nomenclature 2
Numeric Representations 3
Data Units 3
Register Terminology 4
Use of Hypertext 5
Reference Documents 5
Revision History 5

PES32NT8xG2 Device Overview

Overview 1-1
System Identification 1-1
- Vendor ID 1-1
- Device ID 1-1
- Revision ID 1-1
- JTAG ID 1-2
- SSID/SSVID 1-2
- Device Serial Number Enhanced Capability 1-2

Architectural Overview 1-2
Port Operating Modes 1-3
Switch Partitioning 1-6
Non-Transparent Operation 1-7
DMA Operation 1-12
Dynamic Reconfiguration and Failover 1-15
Switch Events 1-16
Multicasting and Non-Transparent Multicasting 1-17

Clocking

Overview 2-1
Port Clocking Modes 2-3
- Global Clocked Mode 2-3
- Local Port Clocked Mode 2-4
- Support for Spread Spectrum Clocking (SSC) 2-5
- Port Clocking Mode Selection 2-6

System Clocking Configurations 2-8

Reset and Initialization

Overview 3-1
Switch Fundamental Reset 3-2
Boot Configuration Vector 3-4
Stack Configuration 3-5
- Static Configuration of a Stack 3-8
- Dynamic Reconfiguration of a Stack via EEPROM / SMBus 3-8
- Switch Modes 3-8

Partition Resets 3-10

Notes

Partition Fundamental Reset 3-10
Partition Hot Reset 3-11
Partition Upstream Secondary Bus Reset 3-12
Partition Downstream Secondary Bus Reset 3-12
Port Mode Change Reset 3-13

Switch Core

Overview 4-1
Switch Core Architecture 4-1
Ingress Buffer 4-3
Egress Buffer 4-3
Crossbar Interconnect 4-4
Virtual Channel Support 4-5
Packet Routing Classes 4-5
Packet Ordering 4-6
Arbitration 4-6
Port Arbitration 4-6
Cut-Through Routing 4-9
Request Metering 4-11
Operation 4-13
Completion Size Estimation 4-14
Internal Errors 4-16
Switch Core Time-Outs 4-17
Memory SECDED ECC Protection 4-18
End-to-End Data Path Parity Protection 4-18
Reporting of Port AER Errors as Internal Errors 4-19

Switch Partition and Port Configuration

Overview 5-1
Switch Partitions 5-1
Partition Configuration 5-2
Partition State 5-3
Partition State Change 5-4
Switch Ports 5-5
Switch Port Mode 5-5
Port Operating Mode Change 5-13
Common Operating Mode Change Behavior 5-15
No Action Mode Change Behavior 5-21
Reset Mode Change Behavior 5-21
Partition Reconfiguration and Failover 5-21
Partition Reconfiguration Latency 5-23
System Notification of Partition Reconfiguration 5-23

Failover

Overview 6-1
Failover Initiation 6-1
Software Initiated Failover 6-2
Signal Initiated Failover 6-2
Watchdog Timer Initiated Failover 6-2

Notes

Link Operation

Overview 7-1
Port Merging 7-1
Port Maximum Link Width 7-2
Polarity Inversion 7-2
Lane Reversal 7-2
Link Width Negotiation 7-4
Link Width Negotiation in the Presence of Bad Lanes 7-5
Dynamic Link Width Reconfiguration 7-5
Dynamic Link Width Reconfiguration in the PES32NT8xG2 7-6
Link Speed Negotiation 7-6
Link Speed Negotiation in the PES32NT8xG2 7-7
Software Management of Link Speed 7-8
Link Retraining 7-9
Link States 7-9
Link Down Handling 7-10
Slot Power Limit Support 7-11
Upstream Port 7-11
Downstream Switch Port 7-12
Link Active State Power Management (ASPM) 7-12
L0s ASPM 7-12
L1 ASPM 7-13
Link Status 7-16
De-emphasis Negotiation 7-16
Crosslink 7-17
Hot Reset Operation on a Crosslink 7-17
Link Disable Operation on a Crosslink 7-18
Gen 1 Compatibility Mode 7-18

SerDes

Overview 8-1
SerDes Numbering and Port Association 8-1
SerDes Transmitter Controls 8-3
Driver Voltage Level and Amplitude Boost 8-3
De-emphasis 8-3
PCI Express Low-Swing Mode 8-4
Receiver Equalization 8-4
Programming of SerDes Controls 8-4
Programmable Voltage Margining and De-Emphasis 8-4
SerDes Transmitter Control Registers 8-5
Transmit Margining Using the PCI Express Link Control 2 Register 8-11
Low-Swing Transmitter Voltage Mode 8-12
Receiver Equalization Controls 8-14
SerDes Power Management 8-14

Power Management

Overview 9-1
Power Management Event (PME) Messages 9-4
PCI Express Power Management Fence Protocol 9-4
Upstream Switch Port or Downstream Switch Port Mode 9-4
NT Function Mode or NT with DMA Function Mode 9-5
Upstream Switch Port with NT and/or DMA Function Mode 9-5

Notes

Transparent Switch Operation

Overview 10-1
Transaction Routing 10-1
Virtual Channel Support 10-2
Maximum Payload Size 10-2
Upstream Port Device Number 10-2
Bus Locking 10-2
Interrupts 10-4
- Downstream Port Interrupts 10-4
- Upstream Port Interrupts 10-4
- Legacy Interrupt Aggregation 10-5

Access Control Services 10-6
ECRC Support 10-10
Error Detection and Handling by the PCI-to-PCI Bridge Function 10-11
- Physical Layer Errors 10-11
- Data Link Layer Errors 10-12
- Transaction Layer Errors 10-13
- Routing Errors 10-23
- Error Emulation Control in the PCI-to-PCI Bridge Function 10-24

Hot-Plug and Hot-Swap

Overview 11-1
Hot-Plug Signals 11-3
Port Reset Outputs 11-5
- Power Enable Controlled Reset Output 11-5
- Power Good Controlled Reset Output 11-6

Hot-Plug Events 11-7
Legacy System Hot-Plug Support 11-7
Hot-Swap 11-8

SMBus Interfaces

Overview 12-1
Master SMBus Interface 12-1
- Initialization and I^2C Reset 12-1
- Serial EEPROM 12-2
- Initialization from Serial EEPROM 12-3
- Programming the Serial EEPROM 12-10
- I/O Expanders 12-11

Slave SMBus Interface 12-19
- Initialization 12-19
- SMBus Transactions 12-20

Setting Up I2C Commands for Block Transactions 12-26
- CSR Register Read or Write Operation 12-26
- SMBus Transactions 12-27
- Examples of Setting Up the I2C CSR Byte Sequence for a CSR Register Read 12-29
- Examples of Setting Up the I2C CSR Byte Sequence for a CSR Register Write 12-32

General Purpose I/O

Overview 13-1
GPIO Configuration 13-1
- Input 13-1
- Output 13-1
- Alternate Function 13-1

Notes

Non-Transparent Switch Operation

Overview 14-1
Base Address Registers (BARs) 14-1
BAR Limit 14-2
Mapping NT Configuration Space to BAR 0 14-4
TLP Translation 14-4
Direct Address Translation 14-4
Lookup Table Address Translation 14-5
ID Translation 14-9
NT Mapping Table 14-9
Request ID Translation 14-11
Completion ID Translation 14-13
Requester ID Capture Register 14-14
TLP Attribute Processing 14-14
No Snoop Processing 14-14
Address Type Processing 14-15
NT Multicast 14-15
Inter-Domain Communications 14-15
Doorbell Registers 14-16
Message Registers 14-17
Punch-Through Configuration Requests 14-18
Re-programming the Bus Number of the NT Function 14-19
Interrupts 14-20
Virtual Channel Support 14-21
Maximum Payload Size 14-22
Power Management 14-22
Bus Locking 14-22
ECRC Support 14-22
Access Control Services (ACS) 14-23
Error Detection and Handling by the NT Function 14-25
Physical Layer Errors 14-26
Data Link Layer Errors 14-26
Transaction Layer Errors 14-26
NTB Inter-Partition Error Propagation 14-31
Error Emulation Control in the NT Function 14-39
Non Transparent Operation Restrictions 14-40

DMA Controller

Overview 15-1
Base Address Registers 15-1
DMA Channel Functional Description 15-1
Data Transfer and Addressing 15-2
DMA Descriptors 15-6
DMA Descriptor Processing 15-15
TLP Attribute and Traffic Class Control 15-20
Channel Interrupts 15-21
DMA Outstanding Requests 15-21
Descriptor Prefetching 15-22
DMA Request Rate Control 15-22

Notes

DMA Multicast 15-23
Interrupts 15-24
Virtual Channel (VC) Support 15-25
Access Control Services (ACS) Support 15-25
Power Management 15-27
Bus Locking 15-27
ECRC Support 15-27
Error Handling 15-27
PCI Express Error Handling by the DMA Function 15-28
DMA Limitations and Usage Restrictions 15-36

Switch Events

Overview 16-1
Switch Events 16-1
Link Up 16-2
Link Down 16-3
Fundamental Reset 16-3
Hot Reset 16-3
Failover 16-3
Global Signals 16-4
Port AER Errors 16-5

Multicast

Overview 17-1
Transparent Multicast Operation 17-1
Addressing and Routing 17-1
Usage Restrictions 17-6
Non-Transparent Multicast Operation 17-6
NT Multicast Configuration 17-7
NT Multicast TLP Determination 17-8
NT Multicast TLP Routing 17-8
NT Multicast Egress Processing 17-9
Usage Restrictions 17-11

Temperature Sensor

Overview 18-1

Register Organization

Overview 19-1
Partial-Byte Access to Word and DWord Registers 19-3
Configuration Register Side-Effects 19-3
Address Maps 19-3
PCI-to-PCI Bridge Function Registers 19-3
Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function 19-11
NT Function Registers 19-14
DMA Function Registers 19-23
Switch Configuration and Status Registers 19-29

Notes

PCI-to-PCI Bridge Registers

Type 1 Configuration Header Registers32NT8 20-1
PCI Express Capability Structure 20-13
PCI Power Management Capability Structure 20-35
Message Signaled Interrupt Capability Structure 20-37
Subsystem ID and Subsystem Vendor ID 20-38
Extended Configuration Space Access Registers 20-39
Advanced Error Reporting (AER) Extended Capability 20-40
Device Serial Number Extended Capability 20-50
PCI Express Virtual Channel Capability 20-51
ACS Extended Capability 20-54
Multicast Extended Capability 20-59

Proprietary Port Specific Registers

Port Control Register 21-1
Upstream PCI-to-PCI Bridge Interrupt and Signaling 21-1
Port AER Mask Register 21-3
Port Slot Control 21-5
Internal Error Control and Status Registers 21-7
Physical Layer Control and Status Registers 21-23
Request Metering 21-28
WRR Port Arbitration Counts 21-29
Non-Transparent Multicast Overlay 21-32
AER Error Emulation 21-34
Global Address Space Access Registers 21-37

NT Endpoint Registers

Type 0 Configuration Header Registers 22-1
PCI Express Capability Structure 22-13
PCI Power Management Capability Structure 22-27
Message Signaled Interrupt Capability Structure 22-28
Subsystem ID and Subsystem Vendor ID 22-30
Extended Configuration Space Access Registers 22-30
Advanced Error Reporting (AER) Extended Capability 22-32
Device Serial Number Extended Capability 22-42
PCI Express Virtual Channel Capability 22-43
ACS Extended Capability 22-47
Multicast Extended Capability 22-49
NT Registers 22-52
- NT Control & Status 22-52
- NT Interrupt and Signaling 22-53
- Internal Error Reporting Masks 22-55
- Doorbell Registers 22-61
- Message Registers 22-62
- BAR Configuration 22-64
- Mapping Table 22-82

Lookup Table 22-85
AER Error Emulation 22-86
Punch-Through Configuration Registers 22-89
NT Multicast 22-91
Global Address Space Access Registers 22-92

Notes

DMA Function Registers

Type 0 Configuration Header Registers 23-1
PCI Express Capability Structure 23-9
PCI Power Management Capability Structure 23-21
Message Signaled Interrupt Capability Structure 23-23
Extended Configuration Space Access Registers 23-24
Advanced Error Reporting (AER) Extended Capability 23-26
ACS Extended Capability 23-36
DMA Registers 23-38
- BAR Configuration 23-38
- DMA AER Error Emulation 23-39
- Internal Error Reporting Masks 23-41
- DMA Multicast Control 23-47
- DMA Channel Registers 23-48

Global Address Space Access Registers 23-57

Switch Configuration and Status Registers

Switch Control and Status Registers 24-1
Internal Switch Timers 24-5
Switch Partition and Port Registers 24-7
Failover Capability Registers 24-13
Protection 24-15
Switch Event Registers 24-16
Global Doorbells and Message Registers 24-23
SerDes Control and Status Registers 24-24
General Purpose I/O Registers 24-31
Hot-Plug and SMBus Interface Registers 24-33
Temperature Sensor Registers 24-44

JTAG Boundary Scan

Introduction 25-1
Test Access Point 25-1
Signal Definitions 25-1
Boundary Scan Chain 25-3
Test Data Register (DR) 25-4
- Boundary Scan Registers 25-5

Instruction Register (IR) 25-6
- EXTEST 25-7
- SAMPLE/PRELOAD 25-7
- BYPASS 25-7
- CLAMP 25-8
- IDCODE 25-8
- VALIDATE 25-8
- EXTEST_TRAIN 25-8
- EXTEST_PULSE 25-9
- RESERVED 25-9
- Usage Considerations 25-9

Usage Models

Introduction 26-1
Boot-time Stack Reconfiguration 26-1
Port Clocking Configuration 26-2
Boot-time Switch Partitioning 26-3

Notes

Switch Partitioning via serial EEPROM 26-4
Switch Partitioning via PCI Express Configuration Requests 26-5
Dynamic Port and Partition Reconfiguration 26-8
I/O Load Balancing: Downstream Port Migration 26-8
Non-Transparent Bridge (NTB) Usage Models 26-11
PES32NT8xG2 as a Multiprocessor System Interconnect 26-11
NT Crosslink & NT Punch-Through 26-15
DMA Usage Models 26-17
High-Performance Multiprocessor System 26-17
Immediate Descriptor Usage 26-20
Failover 26-20
Active / Passive Failover Configuration 26-20
Active / Active Failover Configuration 26-23
Failover with Two Crosslinked PES32NT8xG2 Switches 26-27
NT Multicasting 26-30

Notes

List of Tables

Notes

Table 1.1 PES32NT8xG2 Device IDs 1-1
Table 1.2 PES32NT8xG2 Revision ID 1-2
Table 1.3 Operating Modes Supported by Each Port 1-6
Table 2.1 PxCLK Usage When a Port Operates in Local Port Clocked Mode 2-4
Table 2.2 GCLK and PxCLK frequencies when PxCLK has SSC 2-6
Table 2.3 Port Clocking Mode Requirements 2-6
Table 2.4 Initial Port Clocking Mode and Slot Clock Configuration State 2-7
Table 2.5 Clock Frequency Limitations when Modifying a Port's Clock Mode 2-7
Table 2.6 Valid PES32NT8xG2 System Clocking Configurations 2-8
Table 3.1 PES32NT8xG2 Reset Precedence 3-1
Table 3.2 Boot Configuration Vector Signals 3-5
Table 3.3 Ports in Each Stack 3-6
Table 3.4 Possible Configurations for Stack 0 3-6
Table 3.5 Possible Configurations for Stack 1 3-6
Table 3.6 Possible Configurations for Stack 2 3-7
Table 3.7 Possible Configurations for Stack 3 3-7
Table 3.8 Normal Switch Modes 3-8
Table 3.9 Switch Mode Dependent Register Initialization 3-9
Table 4.1 IFB Buffer Sizes 4-3
Table 4.2 EFB Buffer Sizes 4-4
Table 4.3 Replay Buffer Storage Limit 4-4
Table 4.4 Packet Ordering Rules in the PES32NT8xG2 4-6
Table 4.5 Conditions for Cut-Through Transfers 4-10
Table 4.6 Request Metering Decrement Value 4-14
Table 5.1 Port Functions for Each Port Operating Mode 5-7
Table 5.2 Port Operating Mode Changes Supported by the Switch 5-14
Table 7.1 Crosslink Port Groups 7-17
Table 7.2 Gen 1 Compatibility Mode: bits cleared in training sets 7-18
Table 8.1 SerDes / Port Association for Ports in Stack 0 8-2
Table 8.2 SerDes / Port Association for Ports in Stack 1 8-2
Table 8.3 SerDes / Port Association for Ports in Stack 2 8-2
Table 8.4 SerDes / Port Association for Ports in Stack 3 8-2
Table 8.5 SerDes Transmit Level Controls in the S[x]TXLCTL0 and S[x]TXLCTL1 Registers 8-6
Table 8.6 SerDes Transmit Driver Settings in Gen 1 Mode with -3.5 dB de-emphasis 8-6
Table 8.7 SerDes Transmit Driver Settings in Gen 2 Mode with -3.5 dB de-emphasis 8-7
Table 8.8 SerDes Transmit Driver Settings in Gen 2 Mode with -6.0 dB de-emphasis 8-9
Table 8.9 PCI Express Transmit Margining Levels Supported by the PES32NT8xG2 8-12
Table 8.10 SerDes Transmit Drive Swing in Low Swing Mode at Gen 1 speed 8-13
Table 8.11 SerDes Transmit Drive Swing in Low Swing Mode at Gen 2 Speed 8-13
Table 9.1 PES32NT8xG2 Power Management State Transition Diagram 9-2
Table 10.1 Switch Routing Methods 10-1
Table 10.2 PCI-to-PCI Bridge Function Interrupts 10-4
Table 10.3 Downstream to Upstream Port Interrupt Routing Based on Device Number 10-6
Table 10.4 Prioritization of ACS Checks for Request TLPs 10-9
Table 10.5 Prioritization of ACS Checks for Completion TLPs 10-10
Table 10.6 TLP Types Affected by ACS Checks 10-10
Table 10.7 Physical Layer Errors 10-12
Table 10.8 Data Link Layer Errors 10-12
Table 10.9 Transaction Layer Errors Associated with the PCI-to-PCI Bridge Function 10-14

Notes

Table 10.10 Conditions Handled as Unsupported Requests (UR) by the PCI-to-PCI Bridge Function 10-15
Table 10.11 Conditions Handled as Unexpected Completions (UC) by the PCI-to-PCI Bridge Function 10-16
Table 10.12 Ingress TLP Formation Checks associated with the PCI-to-PCI Bridge Function 10-17
Table 10.13 Egress Malformed TLP Error Checks 10-18
Table 10.14 ACS Violations for Ports Operating in Downstream Switch Port Mode 10-19
Table 10.15 Prioritization of Transaction Layer Errors 10-20
Table 11.1 Port Hot Plug Signals 11-3
Table 11.2 Negated Value of Unused Hot-Plug Output Signals 11-4
Table 12.1 Serial EEPROM SMBus Address 12-2
Table 12.2 PES32NT8xG2 Compatible Serial EEPROMs 12-3
Table 12.3 Serial EEPROM Initialization Errors 12-10
Table 12.4 I/O Expander Functionality Allocation 12-11
Table 12.5 Pin Mapping for I/O Expanders 0 through 3 12-14
Table 12.6 I/O Expander 0 through 3 Port Mapping 12-15
Table 12.7 Pin Mapping I/O Expander 12 12-15
Table 12.8 Pin Mapping I/O Expander 14 12-16
Table 12.9 Pin Mapping of I/O Expander 17 12-16
Table 12.10 Pin Mapping of I/O Expander 18 12-17
Table 12.11 Pin Mapping of I/O Expander 19 12-18
Table 12.12 Pin Mapping of I/O Expander 20 12-18
Table 12.13 Slave SMBus Address 12-19
Table 12.14 Slave SMBus Command Code Fields 12-20
Table 12.15 CSR Register Read or Write Operation Byte Sequence 12-21
Table 12.16 CSR Register Read or Write CMD Field Description 12-22
Table 12.17 Serial EEPROM Read or Write Operation Byte Sequence 12-23
Table 12.18 Serial EEPROM Read or Write CMD Field Description 12-23
Table 12.19 CSR Register Read or Write Operation Byte Sequence 12-27
Table 12.20 Slave SMBus Command Code Fields 12-28
Table 12.21 CSR Register Read or Write CMD Field Description 12-28
Table 12.22 Constants Used in Examples 12-29
Table 12.23 I2C Command Byte Array Indices 12-30
Table 12.24 I2C Command Byte Array Indices 12-31
Table 12.25 I2C Command Byte Array Indices 12-32
Table 12.26 I2C Command Byte Array Indices 12-33
Table 12.27 I2C Command Byte Array Indices 12-34
Table 12.28 I2C Command Byte Array Indice 12-35
Table 13.1 GPIO Pin Configuration 13-1
Table 13.2 GPIO Alternate Function Pin Assignment 13-2
Table 13.3 GPIO Alternate Function Pins 13-2
Table 14.1 NT Endpoint BARs 14-2
Table 14.2 12-Entry Lookup Table Parameters 14-7
Table 14.3 24-Entry Lookup Table Parameters 14-8
Table 14.4 NT Mapping Table Field Description 14-9
Table 14.1 NT Endpoint Interrupts 14-21
Table 14.2 ACS Checks Performed by the NT Function in a Port Operating in Multi-function Mode 14-24
Table 14.3 TLP Types Affected by ACS Checks 14-24
Table 14.4 Transaction Layer Errors Associated with the NT Function 14-27
Table 14.5 Conditions Handled as Unsupported Requests (UR) by the NT Function 14-29
Table 14.6 Conditions Handled as Unexpected Completion (UC) by the NT Function 14-30
Table 14.7 Error Logging at Each Function for UR Example # 1 14-34
Table 14.8 Error Logging at Each Function for UR Example # 2 14-35
Table 14.9 Error Logging at Each Function for Poisoned TLP Example 14-36
Table 14.10 Error Logging at Each Function for Poisoned TLP Example 14-38

Notes

Table 15.1 DMA Channel Addressing Parameters 15-4
Table 15.2 Linear Addressing DMA Example 15-5
Table 15.3 Constant Addressing DMA Example 15-6
Table 15.4 Stride Control DMA Descriptor Fields 15-8
Table 15.5 Data Transfer DMA Descriptor Fields 15-10
Table 15.6 Immediate Data Transfer DMA Descriptor Fields 15-13
Table 15.7 DMA Chaining Disabling 15-17
Table 15.8 DMA Channel Control (DMACxCTL) Register Action Summary 15-19
Table 15.9 Downstream Switch Port Interrupts 15-25
Table 15.10 ACS Checks Performed by the DMA Function 15-26
Table 15.11 TLP Types Affected by ACS Checks 15-26
Table 15.12 PCI Express Errors Detected by the DMA Function's Transaction Layer 15-30
Table 15.13 Prioritization of Transaction Layer Errors 15-35
Table 19.1 Global Address Space Layout 19-1
Table 19.2 PCI-to-PCI Bridge Function Configuration Space Registers 19-6
Table 19.3 Default Linkage of Capability Structures for a PCI-to-PCI Bridge Function in the Upstream Switch Port Mode 19-10
Table 19.4 Default Linkage of Capability Structures for a PCI-to-PCI Bridge Function in a Downstream or Unattached Port 19-10
Table 19.5 Proprietary Port Specific Registers 19-13
Table 19.6 NT Function Registers 19-17
Table 19.7 Default Linkage of Capability Structures for the NT Function When Operating as Function 0 of the Port 19-22
Table 19.8 Default Linkage of Capability Structures for the NT Function When Operating as Function 1 of the Port 19-23
Table 19.9 Default Linkage of Capability Structures for the DMA Function 19-24
Table 19.10 DMA Function Registers 19-26
Table 19.11 Switch Configuration and Status 19-31
Table 25.1 JTAG Pin Descriptions 25-2
Table 25.2 Boundary Scan Chain 25-3
Table 25.3 Instructions Supported by the JTAG Boundary Scan 25-7
Table 25.4 System Controller Device Identification Register 25-8

Notes

List of Figures

Notes

Figure 1.1 PES32NT8xG2 Block Diagram 1-3
Figure 1.2 Logical Representation of a Port with PCI-to-PCI bridge, NT, and DMA Functions 1-5
Figure 1.3 Transparent PCI Express Switch 1-6
Figure 1.4 Partitionable PCI Express Switch 1-7
Figure 1.5 Non-Transparent Bridge 1-8
Figure 1.6 Generalized Multi-Port Non-Transparent Interconnect 1-9
Figure 1.7 Architectural Approaches for Integrating Non-Transparency into a PCI Express Switch .1-10
Figure 1.8 Non-Transparent Switch with Non-Transparency Between Partitions 1-11
Figure 1.9 Non-Transparent Switch with Non-Transparent Ports 1-11
Figure 1.10 Non-Transparent Switch with Non-Transparent Ports 1-12
Figure 1.11 Non-Transparent Switch with Non-Transparent Ports 1-12
Figure 1.12 Switch Partition with DMA function 1-13
Figure 1.13 Two Switch Partitions Interconnected by an NTB, with DMA in One Partition 1-14
Figure 1.14 Two Switch Partitions Interconnected by an NTB, with DMA in Both Partitions 1-15
Figure 1.15 Non-Transparent Switch Failover Usage 1-16
Figure 1.16 Example of Switch Event Mechanism 1-17
Figure 1.17 Example of Transparent Multicast 1-18
Figure 1.18 Example of Non Transparent Multicast 1-18
Figure 2.1 Logical Representation of PES32NT8AG2 Clocking Architecture 2-2
Figure 2.2 Logical Representation of PES32NT8BG2 Clocking Architecture 2-2
Figure 2.3 Clocking Connection for a Port in Global Clocked Mode, with a Common Clocked Configuration 2-3
Figure 2.4 Clocking Connection for a Port in Global Clocked Mode, Non-Common Clocked Configuration 2-3
Figure 2.5 Clocking Connection for a Port in Local Port Clocked Mode, in a Common Clocked Configuration 2-4
Figure 2.6 Clocking Connection for a Port in Local Port Clocked Mode, in a Non-Common Clocked Configuration 2-5
Figure 3.1 Switch Fundamental Reset with Serial EEPROM Initialization 3-3
Figure 3.2 Fundamental Reset Using RSTHALT to Keep Device in Quasi-Reset State 3-4
Figure 4.1 High Level Diagram of Switch Core 4-2
Figure 4.2 Architectural Model of Arbitration 4-7
Figure 4.3 PCI Express Switch Static Rate Mismatch 4-12
Figure 4.4 PCI Express Switch Static Rate Mismatch 4-13
Figure 4.5 Request Metering Counter Decrement Operation 4-14
Figure 4.6 Non-Posted Read Request Completion Size Estimate Computation 4-15
Figure 4.7 Internal Error Logic in Each PES32NT8xG2 Port 4-17
Figure 4.8 Reporting of Port AER Errors as Internal Errors 4-21
Figure 5.1 Allowable Partition State Transitions 5-4
Figure 5.2 Logical Representation of a Port with PCI-to-PCI bridge, NT, and DMA Functions 5-6
Figure 6.1 Failover Policy vs. Failover Reconfiguration 6-1
Figure 7.1 Lane Reversal for Highest Achievable Link Width of x2 7-2
Figure 7.2 Lane Reversal for Highest Achievable Link Width of x4 7-3
Figure 7.3 Lane Reversal for Highest Achievable Link Width of x8 7-4
Figure 7.4 PES32NT8xG2 ASPM Link State Transitions 7-10
Figure 8.1 Relationship Between Coarse and Fine De-emphasis Controls 8-10
Figure 8.2 Effect of Fine de-emphasis Control at Gen 2 with -6.0 dB Nominal de-emphasis 8-11
Figure 9.1 PES32NT8xG2 Power Management State Transition Diagram 9-2
Figure 10.1 Logical Representation of INTx Aggregation 10-5

Notes

Figure 10.2 ACS Source Validation Example 10-7
Figure 10.3 ACS Peer-to-Peer Request Re-direct at a Downstream Switch Port 10-8
Figure 10.4 ACS Upstream Forwarding Example 10-8
Figure 10.5 ACS Peer-to-Peer Request Re-direct by an Upstream PCI-to-PCI Bridge Function 10-9
Figure 10.6 Error Checking and Logging on a Received TLP 10-21
Figure 11.1 Hot-Plug on Switch Downstream Slots Application 11-1
Figure 11.2 Hot-Plug with Switch on Add-In Card Application 11-2
Figure 11.3 Hot-Plug with Carrier Card Application 11-2
Figure 11.4 Power Enable Controlled Reset Output Mode Operation 11-6
Figure 11.5 Power Good Controlled Reset Output Mode Operation 11-6
Figure 12.1 Split SMBus Interface Configuration 12-1
Figure 12.2 Single Double-word Initialization Sequence Format 12-4
Figure 12.3 Sequential Double-word Initialization Sequence Format 12-5
Figure 12.4 Jump Configuration Block 12-5
Figure 12.5 Execution of a Jump Configuration Block 12-6
Figure 12.6 Example of Multiple Configuration Images in Serial EEPROM 12-7
Figure 12.7 Wait Configuration Block 12-8
Figure 12.8 Configuration Done Sequence Format 12-9
Figure 12.9 Slave SMBus Command Code Format 12-20
Figure 12.10 CSR Register Read or Write CMD Field Format 12-21
Figure 12.11 Serial EEPROM Read or Write CMD Field Format 12-23
Figure 12.12 CSR Register Read Using SMBus Block Write/Read Transactions with PEC Disabled 12-24
Figure 12.13 Serial EEPROM Read Using SMBus Block Write/Read Transactions with PEC Disabled 12-24
Figure 12.14 CSR Register Write Using SMBus Block Write Transactions with PEC Disabled 12-25
Figure 12.15 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Disabled 12-25
Figure 12.16 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Enabled 12-25
Figure 12.17 CSR Register Read Using SMBus Read and Write Transactions with PEC Disabled 12-26
Figure 14.1 BAR Limit Operation 14-3
Figure 14.2 Direct Address Translation 14-4
Figure 14.3 Lookup Table Translation 14-5
Figure 14.4 Lookup Table Entry Format 14-6
Figure 14.5 NT Mapping Table 14-9
Figure 14.6 NT Table Partitioning 14-11
Figure 14.7 Request TLP Requester ID Translation 14-12
Figure 14.8 Request TLP Requester ID Translation 14-13
Figure 14.9 Logical Representation of Doorbell Operation 14-17
Figure 14.10 Logical Representation of Message Register Operation 14-18
Figure 14.11 Example of a Rootless PCI Express Hierarchy with Bus Number Reprogramming 14-20
Figure 14.12 Example of ACS Peer-to-Peer Request Re-direct Applied by the NT Function 14-25
Figure 14.13 Basic Non-Transparent PES32NT8xG2 Configuration 14-31
Figure 14.14 Unsupported Request Example # 1 14-33
Figure 14.15 Unsupported Request Example # 2 14-34
Figure 14.16 Poisoned TLP Error Propagation Example 14-36
Figure 14.17 Example of Combined Transaction Layer Error Handling 14-38
Figure 15.1 DMA Data Transfer 15-2
Figure 15.2 Linear Addressing 15-3
Figure 15.3 Linear Addressing Operations 15-3
Figure 15.4 DMA Channel Addressing 15-4
Figure 15.5 Constant Addressing Example 15-6
Figure 15.6 DMA Descriptor List 15-6
Figure 15.7 General DMA Descriptor Format 15-7
Figure 15.8 Stride Control DMA Descriptor Format 15-8
Figure 15.9 Data Transfer DMA Descriptor Format 15-10

Notes

Figure 15.10 Immediate Data Transfer DMA Descriptor Format 15-13
Figure 15.11 DMA Chaining Example 15-17
Figure 15.12 Path Taken by a TLP Emitted by the DMA When it is Multicasted 15-24
Figure 15.13 Path Taken by a TLP Emitted by the DMA When it is NT Multicasted 15-24
Figure 15.14 Example of ACS Peer-to-Peer Request Redirect Applied by the DMA Function 15-27
Figure 15.15 DMA Function's Error Checking and Logging on a Received TLP 15-36
Figure 16.1 Switch Event Detection and Signaling Mechanism 16-2
Figure 16.2 Global Signaling Mechanism 16-4
Figure 17.1 Multicast Group Address Ranges 17-3
Figure 17.2 Multicast Group Address Region Determination 17-4
Figure 17.3 Transparent and Non-Transparent Multicast 17-7
Figure 19.1 PCI-to-PCI Bridge Configuration Space Organization 19-5
Figure 19.2 Proprietary Port Specific Register Organization 19-12
Figure 19.3 NT Function Configuration Space Organization 19-16
Figure 19.4 DMA Function Configuration Space Organization 19-25
Figure 19.5 Switch Configuration and Status Space Organization 19-30
Figure 25.1 Diagram of the JTAG Logic 25-1
Figure 25.2 State Diagram of the TAP Controller 25-2
Figure 25.3 Diagram of Observe-only Input Cell 25-5
Figure 25.4 Diagram of Output Cell 25-5
Figure 25.5 Diagram of Bidirectional Cell 25-6
Figure 25.6 Device ID Register Format 25-8
Figure 26.1 PES24NT24AG2 with One x8 port and Sixteen x1 Ports 26-1
Figure 26.2 PES24NT6AG2 with Ports Operating in Different Clock Modes 26-3
Figure 26.3 PES16NT8BG2 with Two Partitions Configured via Serial EEPROM 26-4
Figure 26.4 PES16NT8BG2 with Two Partitions Configured via a Switch Manager Root Complex 26-6
Figure 26.5 I/O Load Balancing Example: Initial Switch Configuration 26-8
Figure 26.6 I/O Load Balancing Example: Switch Configuration after Port Migration 26-11
Figure 26.7 Multiprocessor System Interconnection Using the PES32NT8xG2 26-12
Figure 26.8 System Configuration immediately after Switch Fundamental Reset 26-15
Figure 26.9 System Configuration after Serial EEPROM Initialization 26-16
Figure 26.10 System Configuration Immediately after Switch Fundamental Reset 26-18
Figure 26.11 Target System Configuration 26-19
Figure 26.12 Active/Passive System Configuration Before Failover Event 26-21
Figure 26.13 Active/Passive System Configuration after Failover Event 26-23
Figure 26.14 Active/Active System Configuration Before Failover Event 26-24
Figure 26.15 Active/Active System Configuration Before Failover Event 26-26
Figure 26.16 High Availability System Configuration with Redundant PCI Express Switches 26-27
Figure 26.17 System Configuration after RC2 Modifies Port 8 in Switch #2 to Downstream Switch Port Mode in Partition 0 26-29
Figure 26.18 System Configuration after RC2 Modifies Port 8 in Switch #1 to Upstream Switch Port Mode in Partition 0 26-30
Figure 26.19 PES32NT8xG2 with Port 0 Configured in NT Function with DMA Mode and Ports 4, 8, and 16 in NT Function Mode 26-31
Figure 26.20 PES32NT8xG2 with Port 0 Configured in NT Function with DMA Mode and Ports 4, 8, and 16 in NT Function Mode 26-32

Notes

Notes

ACSCAP - ACS Capability (0x324) 22-47
ACSCAP - ACS Capability (0x324) 23-37
ACSCAP - ACS Capability Register (0x324) 20-55
ACSCTL - ACS Control (0x326) 22-48
ACSCTL - ACS Control (0x326) 23-37
ACSCTL - ACS Control Register (0x326) 20-57
ACSECAPH - ACS Extended Capability Header (0x320) 20-54
ACSECAPH - ACS Extended Capability Header (0x320) 22-47
ACSECAPH - ACS Extended Capability Header (0x320) 23-36
ACSECV - ACS Egress Control Vector (0x328) 20-58
AERCAP - AER Capabilities (0x100) 20-40
AERCAP - AER Capabilities (0x100) 22-32
AERCAP - AER Capabilities (0x100) 23-26
AERCEM - AER Correctable Error Mask (0x114) 20-47
AERCEM - AER Correctable Error Mask (0x114) 22-39
AERCEM - AER Correctable Error Mask (0x114) 23-33
AERCES - AER Correctable Error Status (0x110) 20-46
AERCES - AER Correctable Error Status (0x110) 22-38
AERCES - AER Correctable Error Status (0x110) 23-32
AERCTL - AER Capabilities and Control (0x118) 20-49
AERCTL - AER Control (0x118) 22-41
AERCTL - AER Control (0x118) 23-35
AERHL1DW - AER Header Log 1st Doubleword (0x11C) 20-49
AERHL1DW - AER Header Log 1st Doubleword (0x11C) 22-42
AERHL1DW - AER Header Log 1st Doubleword (0x11C) 23-36
AERHL2DW - AER Header Log 2nd Doubleword (0x120) 20-49
AERHL2DW - AER Header Log 2nd Doubleword (0x120) 22-42
AERHL2DW - AER Header Log 2nd Doubleword (0x120) 23-36
AERHL3DW - AER Header Log 3rd Doubleword (0x124) 20-50
AERHL3DW - AER Header Log 3rd Doubleword (0x124) 22-42
AERHL3DW - AER Header Log 3rd Doubleword (0x124) 23-36
AERHL4DW - AER Header Log 4th Doubleword (0x128) 20-50
AERHL4DW - AER Header Log 4th Doubleword (0x128) 22-42
AERHL4DW - AER Header Log 4th Doubleword (0x128) 23-36
AERUEM - AER Uncorrectable Error Mask (0x108) 20-42
AERUEM - AER Uncorrectable Error Mask (0x108) 22-33
AERUEM - AER Uncorrectable Error Mask (0x108) 23-27
AERUES - AER Uncorrectable Error Status (0x104) 20-40
AERUES - AER Uncorrectable Error Status (0x104) 22-32
AERUES - AER Uncorrectable Error Status (0x104) 23-26
AERUESV - AER Uncorrectable Error Severity (0x10C) 20-44
AERUESV - AER Uncorrectable Error Severity (0x10C) 22-36
AERUESV - AER Uncorrectable Error Severity (0x10C) 23-30
BAR0 - Base Address Register 0 (0x010) 20-5
BAR0 - Base Address Register 0 (0x010) 22-5
BAR0 - Base Address Register 0 (0x010) 23-5
BAR1 - Base Address Register (0x014) 20-6
BAR1 - Base Address Register 1 (0x014) 22-6
BAR1 - Base Address Register 1 (0x014) 23-6

Notes

BAR2 - Base Address Register 2 (0x018) 22-7
BAR2 - Base Address Register 2 (0x018) 23-6
BAR3 - Base Address Register 3 (0x01C) 22-8
BAR3 - Base Address Register 3 (0x01C) 23-6
BAR4 - Base Address Register 4 (0x020) 22-9
BAR4 - Base Address Register 4 (0x020) 23-6
BAR5 - Base Address Register 5 (0x024) 22-10
BAR5 - Base Address Register 5 (0x024) 23-7
BARLIMIT0 - BAR 0 Limit Address (0x474) 22-66
BARLIMIT1 - BAR 1 Limit Address (0x484) 22-69
BARLIMIT2 - BAR 2 Limit Address (0x494) 22-72
BARLIMIT3 - BAR 3 Limit Address (0x4A4) 22-75
BARLIMIT4 - BAR 4 Limit Address (0x4B4) 22-78
BARLIMIT5 - BAR 5 Limit Address (0x4C4) 22-81
BARLTBASE0 - BAR 0 Lower Translated Base Address (0x478) 22-66
BARLTBASE1 - BAR 1 Lower Translated Base Address (0x488) 22-70
BARLTBASE2 - BAR 2 Lower Translated Base Address (0x498) 22-72
BARLTBASE3 - BAR 3 Lower Translated Base Address (0x4A8) 22-75
BARLTBASE4 - BAR 4 Lower Translated Base Address (0x4B8) 22-78
BARLTBASE5 - BAR 5 Lower Translated Base Address (0x4C8) 22-82
BARSETUP0 - BAR 0 Setup (0x400) 23-38
BARSETUP0 - BAR 0 Setup (0x470) 22-64
BARSETUP1 - BAR 1 Setup (0x480) 22-67
BARSETUP2 - BAR 2 Setup (0x490) 22-70
BARSETUP3 - BAR 3 Setup (0x4A0) 22-73
BARSETUP4 - BAR 4 Setup (0x4B0) 22-76
BARSETUP5 - BAR 5 Setup (0x4C0) 22-79
BARUTBASE0 - BAR 0 Upper Translated Base Address (0x47C) 22-67
BARUTBASE1 - BAR 1 Upper Translated Base Address (0x48C) 22-70
BARUTBASE2 - BAR 2 Upper Translated Base Address (0x49C) 22-73
BARUTBASE3 - BAR 3 Upper Translated Base Address (0x4AC) 22-76
BARUTBASE4 - BAR 4 Upper Translated Base Address (0x4BC) 22-79
BARUTBASE5 - BAR 5 Upper Translated Base Address (0x4CC) 22-82
BCTL - Bridge Control Register (0x03E) 20-12
BCVSTS - Boot Configuration Vector Status (0x0004) 24-2
BIST - Built-in Self Test Register (0x00F) 20-5
BIST - Built-in Self Test Register (0x00F) 22-5
BIST - Built-in Self Test Register (0x00F) 23-5
CAPPTR - Capabilities Pointer (0x034) 22-11
CAPPTR - Capabilities Pointer (0x034) 23-8
CAPPTR - Capabilities Pointer Register (0x034) 20-10
CCISPTR - CardBus CIS Pointer (0x028) 22-11
CCISPTR - CardBus CIS Pointer (0x028) 23-7
CCODE - Class Code (0x009) 22-4
CCODE - Class Code (0x009) 23-4
CCODE - Class Code Register (0x009) 20-4
CLS - Cache Line Size (0x00C) 22-4
CLS - Cache Line Size (0x00C) 23-4
CLS - Cache Line Size Register (0x00C) 20-5
DID - Device Identification (0x002) 22-1
DID - Device Identification (0x002) 23-1
DID - Device Identification Register (0x002) 20-1
DMAC[1:0]CFG - DMA Channel Configuration (0x504/604) 23-48
DMAC[1:0]CTL - DMA Channel Control (0x500/600) 23-48
DMAC[1:0]DPTRH - DMA Channel Descriptor Pointer High (0x52C/62C) 23-56

Notes

DMAC[1:0]DPTRL - DMA Channel Descriptor Pointer Low (0x528/628) 23-56
DMAC[1:0]DSCTL - DMA Channel Destination Stride Control (0x520/620) 23-55
DMAC[1:0]ERRMSK - DMA Channel Error Mask (0x514/614) 23-53
DMAC[1:0]ERRSTS - DMA Channel Error Status (0x510/610) 23-52
DMAC[1:0]MSK - DMA Channel Status Mask (0x50C/60C) 23-51
DMAC[1:0]NDPTRH - DMA Channel Next Descriptor Pointer High (0x534/634) 23-57
DMAC[1:0]NDPTRL - DMA Channel Next Descriptor Pointer Low (0x530/630) 23-57
DMAC[1:0]RRCTL - DMA Channel Request Rate Control (0x524/624) 23-55
DMAC[1:0]SSCTL - DMA Channel Source Stride Control (0x51C/61C) 23-55
DMAC[1:0]SSIZE - DMA Channel Stride Size (0x518/618) 23-54
DMAC[1:0]STS - DMA Channel Status (0x508/608) 23-50
DMACEEM - DMA Correctable Error Emulation (0x40C) 23-40
DMAIERRORMSK0 - Internal Error Reporting Mask 0 (0x410) 23-41
DMAIERRORMSK1 - Internal Error Reporting Mask 1 (0x414) 23-45
DMAUEEM - DMA Uncorrectable Error Emulation (0x408) 23-39
ECFGADDR - Extended Configuration Space Access Address (0x0F8) 20-39
ECFGADDR - Extended Configuration Space Access Address (0x0F8) 22-30
ECFGADDR - Extended Configuration Space Access Address (0x0F8) 23-24
ECFGDATA - Extended Configuration Space Access Data (0x0FC) 20-40
ECFGDATA - Extended Configuration Space Access Data (0x0FC) 22-31
ECFGDATA - Extended Configuration Space Access Data (0x0FC) 23-25
EEPROMINTF - Serial EEPROM Interface (0x1190) 24-40
EROMBASE - Expansion ROM Base (0x030) 22-11
EROMBASE - Expansion ROM Base (0x030) 23-7
EROMBASE - Expansion ROM Base Address Register (0x038) 20-10
FCAP[3:0]CTL - Failover Capability x Control 24-13
FCAP[3:0]STS - Failover Capability x Status 24-14
FCAP[3:0]TIMER - Failover Capability x Watchdog Timer 24-14
GASAADDR - Global Address Space Access Address (0xFF8) 21-37
GASAADDR - Global Address Space Access Address (0xFF8) 22-92
GASAADDR - Global Address Space Access Address (0xFF8) 23-57
GASADATA - Global Address Space Access Data (0xFFC) 21-37
GASADATA - Global Address Space Access Data (0xFFC) 22-92
GASADATA - Global Address Space Access Data (0xFFC) 23-58
GASAPROT - Global Address Space Access Protection (0x0700) 24-15
GDBELLSTS - NT Global Doorbell Status (0x0C3C) 24-23
GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] 24-23
GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] 24-23
GPECTL - General Purpose Event Control (0x11B0) 24-43
GPESTS - General Purpose Event Status (0x11B4) 24-43
GPIOAFSEL - General Purpose I/O Alternate Function Select (0x1170) 24-32
GPIOCFG - General Purpose I/O Configuration (0x1174) 24-32
GPIOD - General Purpose I/O Data (0x1178) 24-33
GPIOFUNC - General Purpose I/O Function (0x116C) 24-31
HDR - Header Type (0x00E) 22-4
HDR - Header Type (0x00E) 23-5
HDR - Header Type Register (0x00E) 20-5
HPCFGCTL - Hot-Plug Configuration Control (0x117C) 24-33
IERRORCTL - Internal Error Reporting Control (0x480) 21-7
IERRORSEV0 - Internal Error Reporting Severity 0 (0x48C) 21-11
IERRORSEV1 - Internal Error Reporting Severity 1 (0x490) 21-14
IERRORSTS0 - Internal Error Reporting Status 0 (0x484) 21-7
IERRORSTS1 - Internal Error Reporting Status 1 (0x488) 21-10
IERRORTST0 - Internal Error Reporting Test 0 (0x494) 21-16
IERRORTST1 - Internal Error Reporting Test 1 (0x498) 21-18

Notes

INDBELLMSK - NT Inbound Doorbell Mask (0x42C) 22-61
INDBELLSTS - NT Inbound Doorbell Status (0x428) 22-61
INMSG[3:0] - Inbound Message [3:0] (0x440-44C) 22-62
INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C) 22-62
INTRLINE - Interrupt Line (0x03C) 22-12
INTRLINE - Interrupt Line (0x03C) 23-8
INTRLINE - Interrupt Line Register (0x03C) 20-11
INTRPIN - Interrupt PIN (0x03D) 22-12
INTRPIN - Interrupt PIN (0x03D) 23-8
INTRPIN - Interrupt PIN Register (0x03D) 20-11
IOBASE - I/O Base Register (0x01C) 20-7
IOBASEU - I/O Base Upper Register (0x030) 20-10
IOEXPADDR0 - SMBus I/O Expander Address 0 (0x1198) 24-40
IOEXPADDR1 - SMBus I/O Expander Address 1 (0x119C) 24-41
IOEXPADDR2 - SMBus I/O Expander Address 2 (0x11A0) 24-41
IOEXPADDR3 - SMBus I/O Expander Address 3 (0x11A4) 24-42
IOEXPADDR4 - SMBus I/O Expander Address 4 (0x11A8) 24-42
IOEXPADDR5 - SMBus I/O Expander Address 5 (0x11AC) 24-43
IOLIMIT - I/O Limit Register (0x01D) 20-7
IOLIMITU - I/O Limit Upper Register (0x032) 20-10
L1ASPMRTC - L1 ASPM Rejection Timer Control (0x710) 21-27
LANESTS0 - Lane Status 0 (0x51C) 21-24
LANESTS1 - Lane Status 1 (0x520) 21-25
LTIMER - Latency Time (0x00D) 22-4
LTIMER - Latency Timer (0x00D) 23-4
LUTLDATA - Lookup Table Lower Data (0x4E4) 22-85
LUTMDATA - Lookup Table Middle Data (0x4E8) 22-86
LUTOFFSET - Lookup Table Offset (0x4E0) 22-85
LUTUDATA - Lookup Table Upper Data (0x4EC) 22-86
MAXLAT - Maximum Latency (0x03F) 22-12
MAXLAT - Maximum Latency (0x03F) 23-9
MBASE - Memory Base Register (0x020) 20-8
MCBARH- Multicast Base Address High (0x33C) 20-61
MCBARH- Multicast Base Address High (0x33C) 22-50
MCBARL- Multicast Base Address Low (0x338) 20-60
MCBARL- Multicast Base Address Low (0x338) 22-50
MCBLKALLH- Multicast Block All High (0x34C) 20-62
MCBLKALLH- Multicast Block All High (0x34C) 22-51
MCBLKALLL- Multicast Block All Low (0x348) 20-62
MCBLKALLL- Multicast Block All Low (0x348) 22-51
MCBLKUTH - Multicast Block Untranslated High (0x354) 20-63
MCBLKUTH - Multicast Block Untranslated High (0x354) 22-52
MCBLKUTL- Multicast Block Untranslated Low (0x350) 20-62
MCBLKUTL- Multicast Block Untranslated Low (0x350) 22-52
MCCAP - Multicast Capability (0x334) 20-59
MCCAP - Multicast Capability (0x334) 22-49
MCCAPH - Multicast Extended Capability Header (0x330) 20-59
MCCAPH - Multicast Extended Capability Header (0x330) 22-49
MCCTL- Multicast Control (0x336) 20-60
MCCTL- Multicast Control (0x336) 22-49
MCOVRBARH- Multicast Overlay Base Address High (0x35C) 20-63
MCOVRBARL- Multicast Overlay Base Address Low (0x358) 20-63
MCRCVH- Multicast Receive High (0x344) 20-61
MCRCVH- Multicast Receive High (0x344) 22-51
MCRCVINT - Multicast Receive Interpretation (0x4FC) 23-47

Notes

MCRCVL- Multicast Receive Low (0x340) 20-61
MCRCVL- Multicast Receive Low (0x340) 22-51
MINGNT - Minimum Grant (0x03E) 22-12
MINGNT - Minimum Grant (0x03E) 23-8
MLIMIT - Memory Limit Register (0x022) 20-8
MSGSTS - Message Status (0x460) 22-62
MSGSTSMSK - Message Status Mask (0x464) 22-63
MSIADDR - Message Signaled Interrupt Address (0x0D4) 20-37
MSIADDR - Message Signaled Interrupt Address (0x0D4) 22-29
MSIADDR - Message Signaled Interrupt Address (0x0D4) 23-24
MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) 20-37
MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) 22-28
MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) 23-23
MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) 20-38
MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) 22-29
MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) 23-24
MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) 20-38
MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) 22-29
MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) 23-24
NTCEEM - NT Endpoint Correctable Error Emulation (0x4F4) 22-88
NTCTL - NT Endpoint Control (0x400) 22-52
NTGSIGNAL - NT Endpoint Global Signal (0x410) 22-55
NTIERRORMSK0 - Internal Error Reporting Mask 0 (0x414) 22-55
NTIERRORMSK1 - Internal Error Reporting Mask 1 (0x418) 22-59
NTINTMSK - NT Endpoint Interrupt Mask (0x408) 22-54
NTINTSTS - NT Endpoint Interrupt Status (0x404) 22-53
NTMCC - NT Multicast Control (0x900) 21-32
NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C) 22-91
NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High 21-34
NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low 21-34
NTMCOVR[3:0]C - NT Multicast Overlay x Configuration 21-33
NTMTBLADDR - NT Mapping Table Address (0x4D0) 22-82
NTMTBLDATA - NT Mapping Table Data (0x4D8) 22-83
NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection 24-15
NTMTBLSTS - NT Mapping Table Status (0x4D4) 22-83
NTSDATA - NT Endpoint Signal Data (0x40C) 22-54
NTUEEM - NT Endpoint Uncorrectable Error Emulation (0x4F0) 22-86
OUTDBELLSET - NT Outbound Doorbell Set (0x420) 22-61
OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C) 22-62
P2PCEEM - PCI-to-PCI Bridge Correctable Error Emulation (0xD94) 21-36
P2PGSIGNAL - PCI-to-PCI Bridge Global Signal (0x414) 21-3
P2PIERRORMSK0 - PCI-to-PCI Bridge Internal Error Reporting Mask 0 (0x4A0) 21-18
P2PIERRORMSK1 - PCI-to-PCI Bridge Internal Error Reporting Mask 1 (0x4A4) 21-22
P2PINTMSK - PCI-to-PCI Bridge Interrupt Mask (0x408) 21-2
P2PINTSTS - PCI-to-PCI Bridge Interrupt Status (0x404) 21-1
P2PSDATA - PCI-to-PCI Bridge Signal Data (0x410) 21-3
P2PUEEM - PCI-to-PCI Bridge Uncorrectable Error Emulation (0xD90) 21-34
PAERMSK - Port AER Mask (0x424) 21-3
PBUSN - Primary Bus Number Register (0x018) 20-6
PCICMD - PCI Command (0x004) 22-1
PCICMD - PCI Command (0x004) 23-1
PCICMD - PCI Command Register (0x004) 20-2
PCIECAP - PCI Express Capability (0x040) 20-13
PCIECAP - PCI Express Capability (0x040) 22-13
PCIECAP - PCI Express Capability (0x040) 23-9

Notes

PCIEDCAP - PCI Express Device Capabilities (0x044) 20-14
PCIEDCAP - PCI Express Device Capabilities (0x044) 22-13
PCIEDCAP - PCI Express Device Capabilities (0x044) 23-10
PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) 20-30
PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) 22-22
PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) 23-18
PCIEDCTL - PCI Express Device Control (0x048) 20-16
PCIEDCTL - PCI Express Device Control (0x048) 22-15
PCIEDCTL - PCI Express Device Control (0x048) 23-11
PCIEDCTL2 - PCI Express Device Control 2 (0x068) 20-31
PCIEDCTL2 - PCI Express Device Control 2 (0x068) 22-23
PCIEDCTL2 - PCI Express Device Control 2 (0x068) 23-19
PCIEDSTS - PCI Express Device Status (0x04A) 20-17
PCIEDSTS - PCI Express Device Status (0x04A) 22-17
PCIEDSTS - PCI Express Device Status (0x04A) 23-13
PCIEDSTS2 - PCI Express Device Status 2 (0x06A) 20-32
PCIEDSTS2 - PCI Express Device Status 2 (0x06A) 22-24
PCIEDSTS2 - PCI Express Device Status 2 (0x06A) 23-20
PCIELCAP - PCI Express Link Capabilities (0x04C) 20-18
PCIELCAP - PCI Express Link Capabilities (0x04C) 22-18
PCIELCAP - PCI Express Link Capabilities (0x04C) 23-14
PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) 20-32
PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) 22-24
PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) 23-20
PCIELCTL - PCI Express Link Control (0x050) 20-20
PCIELCTL - PCI Express Link Control (0x050) 22-19
PCIELCTL - PCI Express Link Control (0x050) 23-16
PCIELCTL2 - PCI Express Link Control 2 (0x070) 20-32
PCIELCTL2 - PCI Express Link Control 2 (0x070) 22-24
PCIELCTL2 - PCI Express Link Control 2 (0x070) 23-20
PCIELSTS - PCI Express Link Status (0x052) 20-22
PCIELSTS - PCI Express Link Status (0x052) 22-21
PCIELSTS - PCI Express Link Status (0x052) 23-17
PCIELSTS2 - PCI Express Link Status 2 (0x072) 20-34
PCIELSTS2 - PCI Express Link Status 2 (0x072) 22-26
PCIELSTS2 - PCI Express Link Status 2 (0x072) 23-21
PCIESCAP - PCI Express Slot Capabilities (0x054) 20-24
PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074) 20-34
PCIESCTL - PCI Express Slot Control (0x058) 20-26
PCIESCTL2 - PCI Express Slot Control 2 (0x078) 20-34
PCIESCTLIV - PCI Express Slot Control Initial Value (0x430) 21-5
PCIESSTS - PCI Express Slot Status (0x05A) 20-29
PCIESSTS2 - PCI Express Slot Status 2 (0x07A) 20-35
PCIEVCECAP - PCI Express VC Extended Capability Header (0x200) 20-51
PCIEVCECAP - PCI Express VC Extended Capability Header (0x200) 22-44
PCISTS - PCI Status (0x006) 22-2
PCISTS - PCI Status (0x006) 23-3
PCISTS - PCI Status Register (0x006) 20-3
PCLKMODE - Port Clocking Mode (0x0008) 24-3
PHYLCFG0 - Phy Link Configuration 0 (0x530) 21-25
PHYLSTATE0 - Phy Link State 0 (0x540) 21-26
PHYPRBS - Phy PRBS Seed (0x55C) 21-27
PLTIMER - Primary Latency Timer (0x00D) 20-5
PMBASE - Prefetchable Memory Base Register (0x024) 20-9
PMBASEU - Prefetchable Memory Base Upper Register (0x028) 20-9

Notes

PMCAP - PCI Power Management Capabilities (0x0C0) 20-35
PMCAP - PCI Power Management Capabilities (0x0C0) 22-27
PMCAP - PCI Power Management Capabilities (0x0C0) 23-21
PMCSR - PCI Power Management Control and Status (0x0C4) 20-36
PMCSR - PCI Power Management Control and Status (0x0C4) 22-27
PMCSR - PCI Power Management Control and Status (0x0C4) 23-22
PMLIMIT - Prefetchable Memory Limit Register (0x026) 20-9
PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) 20-10
POMCDELAY - Port Operating Mode Change Drain Delay (0x0084) 24-5
PORTCTL - Port Control (0x400) 21-1
PTCCTL0 - Punch-Through Configuration Control 0 (0x510) 22-89
PTCCTL1 - Punch-Through Configuration Control 1 (0x514) 22-90
PTCDATA - Punch-Through Data (0x518) 22-90
PTCSTS - Punch-Through Status (0x51C) 22-90
PVCCAP1- Port VC Capability 1 (0x204) 20-51
PVCCAP1- Port VC Capability 1 (0x204) 22-44
PVCCAP2- Port VC Capability 2 (0x208) 20-52
PVCCAP2- Port VC Capability 2 (0x208) 22-45
PVCCTL - Port VC Control (0x20C) 20-52
PVCCTL - Port VC Control (0x20C) 22-45
PVCSTS - Port VC Status (0x20E) 20-52
PVCSTS - Port VC Status (0x20E) 22-45
RDRAINDELAY - Reset Drain Delay (0x0080) 24-5
REQIDCAP - Requester ID Capture (0x4DC) 22-84
RID - Revision Identification (0x008) 22-4
RID - Revision Identification (0x008) 23-4
RID - Revision Identification Register (0x008) 20-4
RMCOUNT - Requester Metering Count (0x88C) 21-29
RMCTL - Requester Metering Control (0x880) 21-28
S[7:0]CTL- SerDes x Control 24-25
S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control 24-31
S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 24-26
S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 24-28
SBUSN - Secondary Bus Number Register (0x019) 20-6
SECSTS - Secondary Status Register (0x01E) 20-7
SEDELAY - Side Effect Delay (0x0088) 24-6
SEFOVRMSK - Switch Event Failover Mask (0x0C2C) 24-21
SEFRSTMSK - Switch Event Fundamental Reset Mask (0x0C20) 24-20
SEFRSTSTS - Switch Event Fundamental Reset Status (0x0C1C) 24-20
SEGSIGMSK - Switch Event Global Signal Mask (0x0C34) 24-22
SEGSIGSTS - Switch Event Global Signal Status (0x0C30) 24-22
SEHRSTMSK - Switch Event Hot Reset Mask (0x0C28) 24-21
SEHRSTSTS - Switch Event Hot Reset Status (0x0C24) 24-21
SELINKDNMSK - Switch Event Link Down Mask (0x0C18) 24-20
SELINKDNSTS - Switch Event Link Down Status (0x0C14) 24-20
SELINKUPMSK - Switch Event Link Up Mask (0x0C10) 24-19
SELINKUPSTS - Switch Event Link Up Status (0x0C0C) 24-19
SEMSK - Switch Event Mask (0x0C04) 24-17
SEPMSK - Switch Event Partition Mask (0x0C08) 24-19
SERDESCFG - SerDes Configuration (0x510) 21-24
SESTS - Switch Event Status (0x0C00) 24-16
SLTIMER - Secondary Latency Timer Register (0x01B) 20-6
SMBUSCBHL - SMBus Configuration Block Header Log (0x11E8) 24-39
SMBUSCTL - SMBus Control (0x118C) 24-37
SMBUSSTS - SMBus Status (0x1188) 24-35

Notes

SNUMCAP - Serial Number Capabilities (0x180) 20-50
SNUMCAP - Serial Number Capabilities (0x180) 22-42
SNUMLDW - Serial Number Lower Doubleword (0x184) 20-50
SNUMLDW - Serial Number Lower Doubleword (0x184) 22-43
SNUMUDW - Serial Number Upper Doubleword (0x188) 20-51
SNUMUDW - Serial Number Upper Doubleword (0x188) 22-43
SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) 20-39
SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) 22-30
SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) 20-38
SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) 22-30
STK0CFG - Stack Configuration (0x0010) 24-3
STK1CFG - Stack Configuration (0x0014) 24-4
STK2CFG - Stack Configuration (0x0018) 24-4
STK3CFG - Stack Configuration (0x001C) 24-4
SUBID - Subsystem ID Pointer (0x02E) 22-11
SUBID - Subsystem ID Pointer (0x02E) 23-7
SUBUSN - Subordinate Bus Number Register (0x01A) 20-6
SUBVID - Subsystem Vendor ID Pointer (0x02C) 22-11
SUBVID - Subsystem Vendor ID Pointer (0x02C) 23-7
SWCTL - Switch Control (0x0000) 24-1
SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] 24-24
SWPART[7:0]CTL - Switch Partition x Control 24-7
SWPART[7:0]FCTL - Switch Partition x Failover Control 24-9
SWPART[7:0]STS - Switch Partition x Status 24-8
SWPORTxCTL - Switch Port x Control 24-9
SWPORTxFCTL - Switch Port x Failover Control 24-12
SWPORTxSTS - Switch Port x Status 24-10
TLCNTCFG - Transaction Layer Countables Configuration (0x690) 21-27
TMPADJ - Temperature Sensor Adjustment (0x11E0) 24-48
TMPALARM - Temperature Sensor Alarm (0x11DC) 24-46
TMPCTL - Temperature Sensor Control (0x11D4) 24-44
TMPSTS - Temperature Sensor Status (0x11D8) 24-45
TSSLOPE - Temperature Sensor Slope (0x11E4) 24-48
USSBRDELAY - Upstream Secondary Bus Reset Delay (0x008C) 24-6
VC0PARBCI0 - VC0 Port Arbiter Counter Initialization 0 (0x890) 21-29
VC0PARBCI1 - VC0 Port Arbiter Counter Initialization 1 (0x894) 21-30
VC0PARBCI2 - VC0 Port Arbiter Counter Initialization 2 (0x898) 21-30
VC0PARBCI3 - VC0 Port Arbiter Counter Initialization 3 (0x89C) 21-30
VC0PARBCI4 - VC0 Port Arbiter Counter Initialization 4 (0x8A0) 21-31
VC0PARBCI5 - VC0 Port Arbiter Counter Initialization 5 (0x8A4) 21-31
VC0PARBCI6 - VC0 Port Arbiter Counter Initialization 6 (0x8A8) 21-31
VCR0CAP- VC Resource 0 Capability (0x210) 20-53
VCR0CAP- VC Resource 0 Capability (0x210) 22-45
VCR0CTL- VC Resource 0 Control (0x214) 20-53
VCR0CTL- VC Resource 0 Control (0x214) 22-46
VCR0STS - VC Resource 0 Status (0x218) 20-54
VCR0STS - VC Resource 0 Status (0x218) 22-46
VID - Vendor Identification (0x000) 22-1
VID - Vendor Identification (0x000) 23-1
VID - Vendor Identification Register (0x000) 20-1

PES32NT8xG2 Device Overview

Notes

Overview

The 89HPES32NT8xG2 is a member of the IDT family of PCI Express® switching solutions. The PES32NT8xG2 is a 32-lane, 8-port system interconnect switch optimized for PCI Express Gen2 packet switching in high-performance applications, supporting multiple simultaneous peer-to-peer traffic flows. Target applications include multi-host or intelligent I/O based systems where inter-domain communication is required, such as servers, storage, communications, and embedded systems.

With Non-Transparent Bridging functionality and innovative Switch Partitioning feature, the PES32NT8xG2 allows true multi-host or multi-processor communications in a single device. Integrated DMA controllers enable high-performance system design by off-loading data transfer operations across memories from the processors. Each lane is capable of 5 GT/s link speed in both directions and is fully compliant with PCI Express Base Specification 2.1.

A non-transparent bridge (NTB) is required when two PCI Express domains need to communicate to each other. The main function of the NTB block is to initialize and translate addresses and device IDs to allow data exchange across PCI Express domains.

Note: The part number PES32NT8xG2 covers two distinct switch devices: the PES32NT8AG2 and the PES32NT8BG2. The information in this manual applies equally to both devices except where noted in occasional notes and footnotes in various chapters. The differences between the two devices are summarized as follows:

- Port clocking: the PES32NT8AG2 supports port clocking on all 8 ports. The PES32NT8BG2 supports port clocking on ports 0, 2, and 4 only.
- Slave SMBus address pins: The PES32NT8AG2 has 2 address pins while the PES32NT8BG2 has only one pin.

System Identification

Vendor ID

All vendor IDs in the device are hardwired to 0x111D which corresponds to Integrated Device Technology, Inc.

Device ID

The PES32NT8xG2 device IDs are shown in Table 1.1.

Device	PCIe Device	Device ID
PES32NT8AG2	0x7	0x808F
PES32NT8BG2	0x0	0x8088

Table 1.1 PES32NT8xG2 Device IDs

Revision ID

The revision ID in the PES32NT8xG2 is set to the same value in all mode. The value of the revision ID is determined in one place and is easily modified during a metal mask change. The revision ID will start at 0x0 and will be incremented with each all-layer or metal mask change.

Notes

Revision ID	Description
0x0	Corresponds to ZA silicon
0x1	Corresponds to ZB silicon
0x2	Corresponds to ZC silicon

Table 1.2 PES32NT8xG2 Revision ID

JTAG ID

The JTAG ID is:

- Version: Same value as Revision ID. See Table 1.2
- Part number: Same value as base Device ID. See Table 1.1.
- Manufacture ID: 0x33
- LSB: 0x1

SSID/SSVID

The PES32NT8xG2 contains the mechanisms necessary to implement the PCI-to-PCI bridge Subsystem ID and Subsystem Vendor ID capability structure. However, in the default configuration the Subsystem ID and Subsystem Vendor ID capability structure is not enabled. To enable this capability, the SSID and SSVID fields in the Subsystem ID and Subsystem Vendor ID (SSIDSSVID) register must be initialized with the appropriate ID values. the Next Pointer (NXTPTR) field in one of the other enhanced capabilities should be initialized to point to this capability. Finally, the Next Pointer (NXTPTR) of this capability should be adjusted to point to the next capability if necessary.

Device Serial Number Enhanced Capability

The PES32NT8xG2 contains the mechanisms necessary to implement the PCI express device serial number enhanced capability. However, in the default configuration this capability structure is not enabled. To enable the device serial number enhanced capability, the Serial Number Lower Doubleword (SNUMLDW) and the Serial Number Upper Doubleword (SNUMUDW) registers should be initialized. The Next Pointer (NXTPTR) field in one of the other enhanced capabilities should be initialized to point to this capability. Finally, the Next Pointer (NXTPTR) of this capability should be adjusted to point to the next capability if necessary.

Architectural Overview

This section provides a high level architectural overview of the switch. An architectural block diagram of the switch is shown in Figure 1.1.

Notes

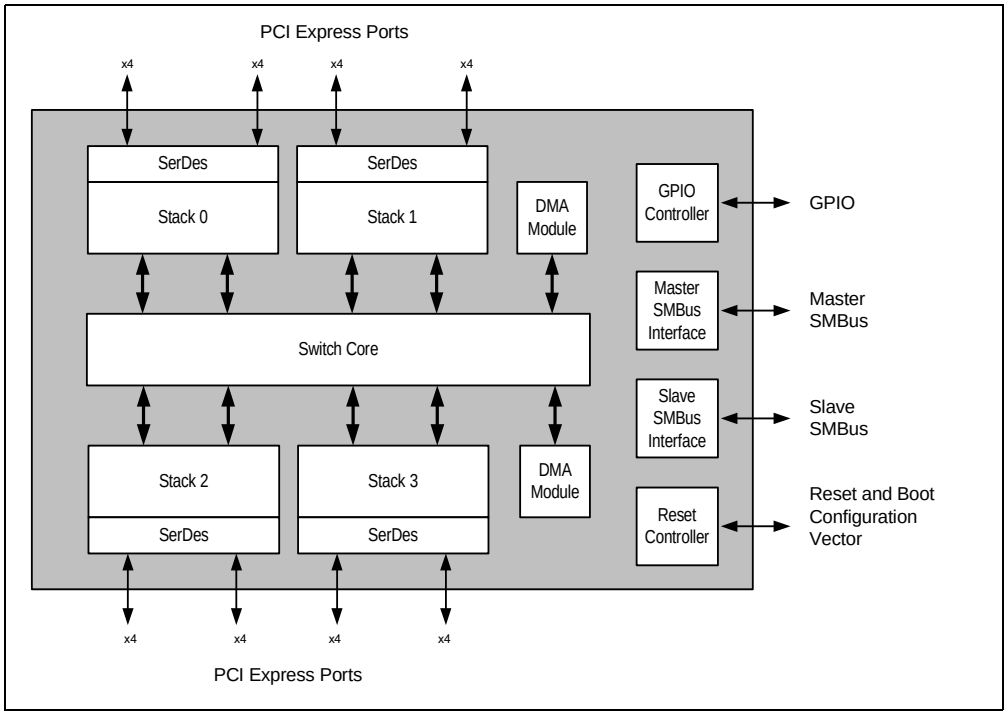

Figure 1.1 PES32NT8xG2 Block Diagram

The switch contains 8 ports labeled ports 0 , 2, 4, 6, 8, 12, 16, and 20. All ports support 2.5 GTps (e.g., Gen 1) and 5.0 GTps (e.g., Gen 2) operation.

At a high level, the switch consists of four PCI Express (PCIe) stacks, two DMA modules, a switch core, and peripheral blocks associated with SMBus functionality, GPIO functionality, reset, etc. A stack consists of a logic that performs functions associated with the physical, data link, and transactions layers described in the PCI Express Base Specification 2.1. In addition, a stack performs switch application layer functions such as Transaction Layer Packet (TLP) routing using route map tables, processing configuration read and write requests, non-transparent address translation, etc.

Each of the four stacks is composed of two x4 ports for a total of eight x4 ports. These stacks may be configured such that both x4 ports can be merged into one x8 port. Stack configurations are described in section Stack Configuration on page 3-5. The DMA modules contain the logic and state associated with DMA functionality. DMA functionality is introduced below and described in detail in Chapter 15, DMA Controller.

The switch core is responsible for transferring TLPs between stacks. Its main functions are input buffering, maintaining per port ingress and egress flow control information, port arbitration, scheduling, and forwarding TLPs between ports. Since the switch represents a single architecture optimized for both fan-out and system interconnect applications, its switch core is based on a non-blocking crossbar. Chapter 4 describes the switch core architecture and operation in detail.

Port Operating Modes

Ports operate independently from each other, even if the ports are in the same stack. Each port has several operational modes that determine the behavior of the port, the PCI functions (e.g., PCI-to-PCI bridge, Non-Transparent (NT) endpoint, and DMA endpoint) associated with the port, etc. Port operating modes are introduced below and described in detail in Chapter 5, Switch Partition and Port Configuration.

Notes

PES32NT8xG2 ports support the following port operating modes.

- Disabled
- Unattached
- Upstream switch port (i.e., upstream PCI-to-PCI bridge)
- Downstream switch port (i.e., downstream PCI-to-PCI bridge)
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function
- NT with DMA function

Figure 1.2 shows a logical diagram of a port. Depending on the port operating mode, the port may contain one, two, or up to three PCI Express functions (e.g., PCI-to-PCI bridge, NT, and DMA functions). The figure shows a port with all three functions. Multi-function ports always face upstream (i.e., they are considered upstream ports).

- For example, a port in upstream switch port mode contains only an upstream PCI-to-PCI bridge function. Similarly, a port in downstream switch port mode contains only a downstream PCI-to-PCI bridge function. The PCI-to-PCI bridge function serves as a bridge between the PCI Express link and the switch's virtual PCI bus.
- A port in NT function mode contains a single Non Transparent (NT) endpoint function facing upstream. This function serves as a non-transparent bridge between the port's PCI Express link and the switch's NT Interconnect. Refer to section Non-Transparent Operation on page 1-7 for details.
- A port in upstream switch port with NT and DMA functions mode contains three functions, an upstream PCI-to-PCI bridge function, an NT function, and a DMA function. The port faces upstream.
- A port in Disabled mode is disabled and it's PCI Express link is turned off.

Other modes are possible, as listed above. Refer to Chapter 5 for details.

When a port is configured with two or more functions, data transfers across the functions (i.e., inter-function transfers) are possible. For example, TLPs may be transferred from the PCI-to-PCI bridge function to the NT function and vice-versa. Similarly, TLPs may be transferred from the DMA function to the NT function and vice-versa. Finally, TLPs may be transferred from the DMA function to the PCI-to-PCI bridge function and vice-versa. Inter-function transfers occur within the port and are not emitted on the port's PCI Express link.

Note that a port's link width is not determined by the port's operating mode. Instead, the port's maximum link width is determined by the configuration of the stack associated with the port (e.g., a stack may be configured as one x8 port or eight x1 ports). The actual link width that the port achieves is determined during link training. Refer to Chapter 7, Link Operation, for details on link operation.

Notes

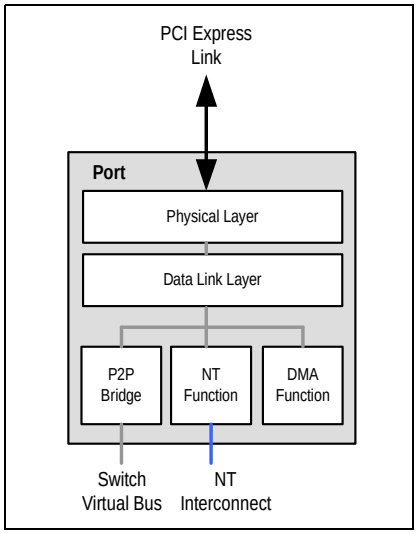

Figure 1.2 Logical Representation of a Port with PCI-to-PCI bridge, NT, and DMA Functions

Not all ports support all port operating modes. The following applies.

- All ports support the Disabled, Unattached, Upstream Switch Port, and Downstream Switch port mode.
- All ports support port operating modes associated with an NT function.
- Two ports support port operating modes associated with a DMA function. These are ports 0 and 8.
- Table 1.3 lists all the operating modes and their support by each port. Ports marked with a blue dot support the corresponding operating mode.

The operating modes listed above allow for highly flexible configurations of the switch. These operating modes are tightly associated with the topics of switch partitioning, non-transparent operation, and DMA operation introduced in the following sections. Port modes may be modified at boot-time (i.e., fundamental reset of the switch) or run-time (i.e., after fundamental reset).

Notes

Port Operating Mode	Port Support							
	0	2	4	6	8	12	16	20
Disabled	•	•	•	•	•	•	•	•
Unattached	•	•	•	•	•	•	•	•
Upstream switch port	•	•	•	•	•	•	•	•
Downstream switch port	•	•	•	•	•	•	•	•
Upstream switch port with DMA function	•				•			
Upstream switch port with NT function	•	•	•	•	•	•	•	•
Upstream switch port with NT and DMA functions	•				•			
NT function	•	•	•	•	•	•	•	•
NT with DMA function	•				•			

Table 1.3 Operating Modes Supported by Each Port

Switch Partitioning

The logical view of a PCI Express switch is shown in Figure 1.3. A PCI Express switch contains one upstream port and one or more downstream ports. Each port is associated with a PCI-to-PCI (P2P) bridge function. All PCI-to-PCI bridges associated with a PCI Express switch are interconnected by a virtual PCI bus.

- The primary side of the upstream port's PCI-to-PCI bridge is associated with the external link, while the secondary side connects to the virtual PCI bus.
- The primary side of a downstream port's PCI-to-PCI bridge is connected to the virtual PCI bus, while the secondary side is associated with the external link.

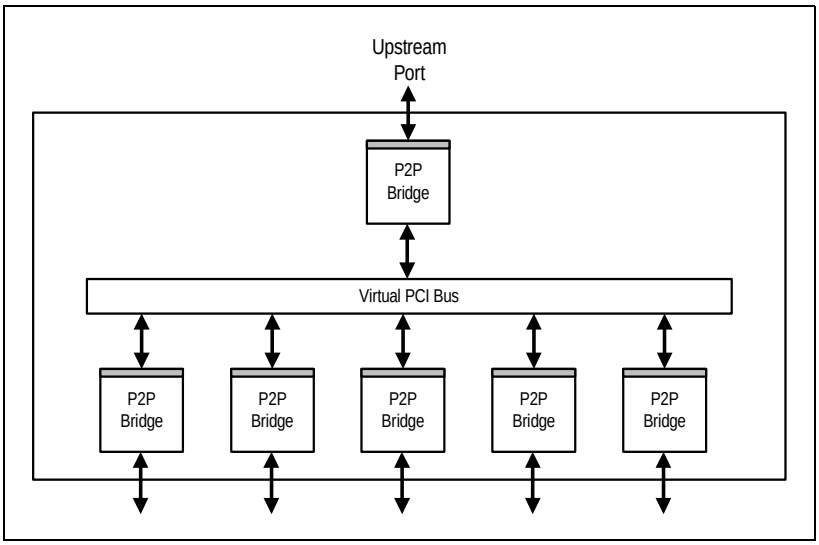

Figure 1.3 Transparent PCI Express Switch

The PES32NT8xG2 is a *partitionable* PCI Express switch. This means that in addition to operating as a standard PCI Express switch, PES32NT8xG2 ports may be partitioned into groups that logically operate as completely independent PCI Express switches. Figure 1.4 illustrates a three partition switch configuration.

Notes

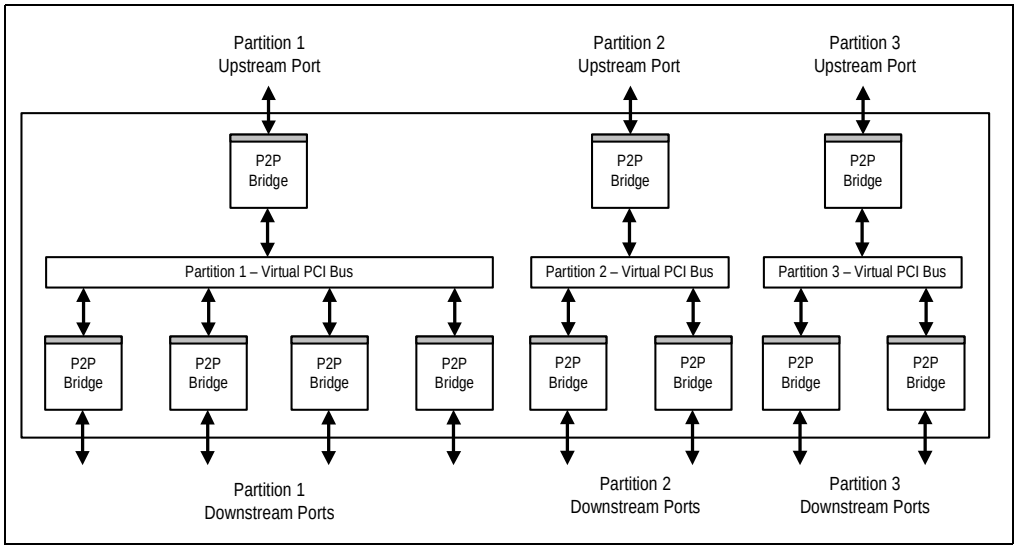

Figure 1.4 Partitionable PCI Express Switch

Each partition operates logically as a completely independent PCI Express switch that implements the behavior and capabilities outlined in the PCI Express Base Specification 2.1required of a switch. Conceptually, switch partitioning allows the logical division of the PCI Express switch into multiple partitions, each of which is composed of a configurable number of ports, and each of which connects to a separate PCIe domain[1]. Each switch partition is logically isolated from the other partitions. From the switch's perspective, a switch partition represents a logical container that contains switch ports associated with a PCIe domain. Any switch port can be configured to belong one partition.

The PES32NT8xG2 supports boot-time (i.e., at fundamental reset) and runtime (i.e., after fundamental reset) configuration of ports and partitions. Boot-time configuration creates an initial grouping of ports into partition and assigns the operating modes of the ports. Boot-time configuration may be performed via serial EEPROM, external SMBus master, or software executing on a root port (e.g., BIOS, OS, driver, or hypervisor).

Basic preconfigurations of the switch may be chosen using the Switch Mode (SWMODE[3:0]) pins at fundamental reset. When using these preconfigurations, boot-time configuration of the ports and partitions in the switch is not required, although it is still allowed. Refer to Chapter 3, Reset and Initialization.

Runtime reconfiguration allows the number of active partitions in the device and assignment of ports to partitions to be modified while the system is active. Runtime reconfiguration may be performed by an external SMBus master or by software executing on a root port. Runtime reconfiguration does not affect either a port or a partition whose configuration is not modified. Runtime reconfiguration of ports and partitions is further described in section Dynamic Reconfiguration and Failover on page 1-15. Switch partitioning is described in detail in Chapter 5.

Non-Transparent Operation

The PCI architecture defines a hierarchy of buses interconnected by PCI-to-PCI bridges. This hierarchy forms a tree and is referred to as a PCI domain.

- A PCI domain consists of a single memory address space, I/O address space, and ID address space.
- The PCI ID consists of a bus, device and function number that uniquely defines an element in the domain.

1. A PCIe domain is the collection of PCIe devices under a common processor/memory complex (i.e., root-complex), sharing common PCIe memory, I/O, and configuration spaces.

Notes

Although PCI Express switches support direct transfers between ports, the logical view seen by software remains that of a hierarchy of buses as defined by the PCI architecture and illustrated in Figure 1.3. The portion of a PCI domain emanating from a PCI Express root complex is referred to as the PCI Express domain.

In many applications, a need exists to interconnect two independent PCI domains. A Non-Transparent Bridge (NTB) enables this inter-domain communication. The architecture of an NTB is illustrated in Figure 1.5.

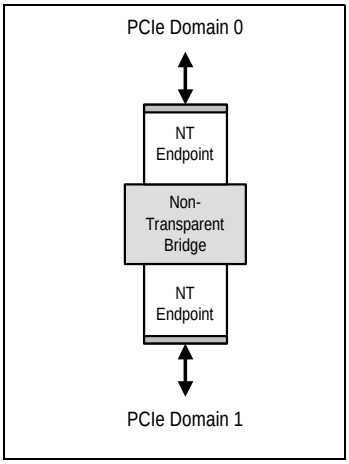

Figure 1.5 Non-Transparent Bridge

An NTB consists of two PCI functions each defined by a Type 0 PCI header that are interconnected by a bridging function. The two Type 0 PCI functions are referred two as Non-Transparent (NT) endpoints (a.k.a. NT functions). Each function advertises one or more memory windows using PCI Base Address Registers (BARs). Software executing on each hierarchy allocates PCI memory space to the BAR. Memory operations that target a memory window defined by an NT endpoint are routed within the PCI domain to that endpoint. When the non-transparent bridge receives a memory operation that targets a BAR used for mapping through the bridge, it translates the address of the transaction to a new address in the opposite domain and forwards the transaction to the other domain. Completions are handled in a similar manner.

The first non-transparent bridge was developed in 1997 Digital Semiconductor and called Drawbridge (a.k.a. 21554). Drawbridge has been widely used to construct PCI based multi-processors and intelligent I/O adapters. In 2004 PLX extended the Drawbridge NTB architecture to PCI Express by introducing Requester ID translation. The PLX approach limited the number of masters to 8 on one side of the bridge and 32 on the other side.

While maintaining the architectural concepts of the original Digital architecture, the switch extends non-transparent bridging to allow direct non-transparent switching between two or more domains, and between up to 64 masters. As shown in Figure 1.6, the switch allows two or more non-transparent endpoints to directly communicate over a *non-transparent (NT) interconnect*. This extension of non-transparent bridging from two ports to multiple ports parallels the evolution of two-port PCI bridges to multi-port PCI Express switches.

Non-transparent operation is related to the concept of switch partitioning in that the non-transparent interconnect allows switching between multiple switch partitions, each of which is associated with a separate PCIe domain.

Notes

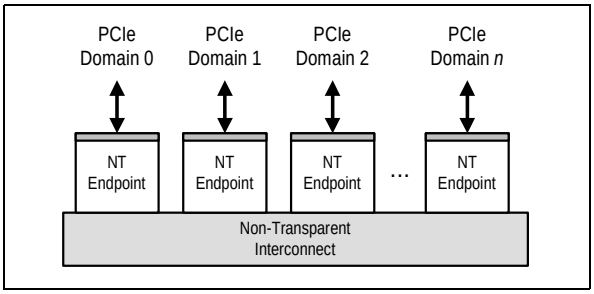

Figure 1.6 Generalized Multi-Port Non-Transparent Interconnect

There are numerous approaches for integrating a non-transparent bridge into a PCI Express switch. Figure 1.7 illustrates three approaches.

Figure 1.7(a) shows an architecture in which a non-transparent bridge is integrated below the PCI-to-PCI bridge associated with a downstream port. This architecture is used in IDT Gen 1 switches. A disadvantage of this approach is that it leads to complex implementations when extended to direct non-transparent switching.

Figure 1.7(b) illustrates an architecture in which a non-transparent bridge is integrated directly onto the virtual PCI bus. The advantage of this approach is that it is simple to implement since the PCI-to-PCI bridge associated with a downstream port may be replaced (or reconfigured) with a non-transparent bridge. The issue with this approach is that it violates the fundamental requirement outlined in the PCI Express base specification that endpoints (represented by type 0h headers) must not appear to configuration software on a switch's internal bus as peers of the virtual PCI-to-PCI bridges representing switch downstream ports.[1]

Figure 1.7(c) exhibits the architecture used in the switch. In this architecture, the upstream port is transformed into a multi-function device with two functions, one representing the PCI-to-PCI bridge associated with the upstream port, and the other representing the NT endpoint.

[1] Refer to Chapter 7 in the PCI Express Base Specification Revision 2.1.

Notes

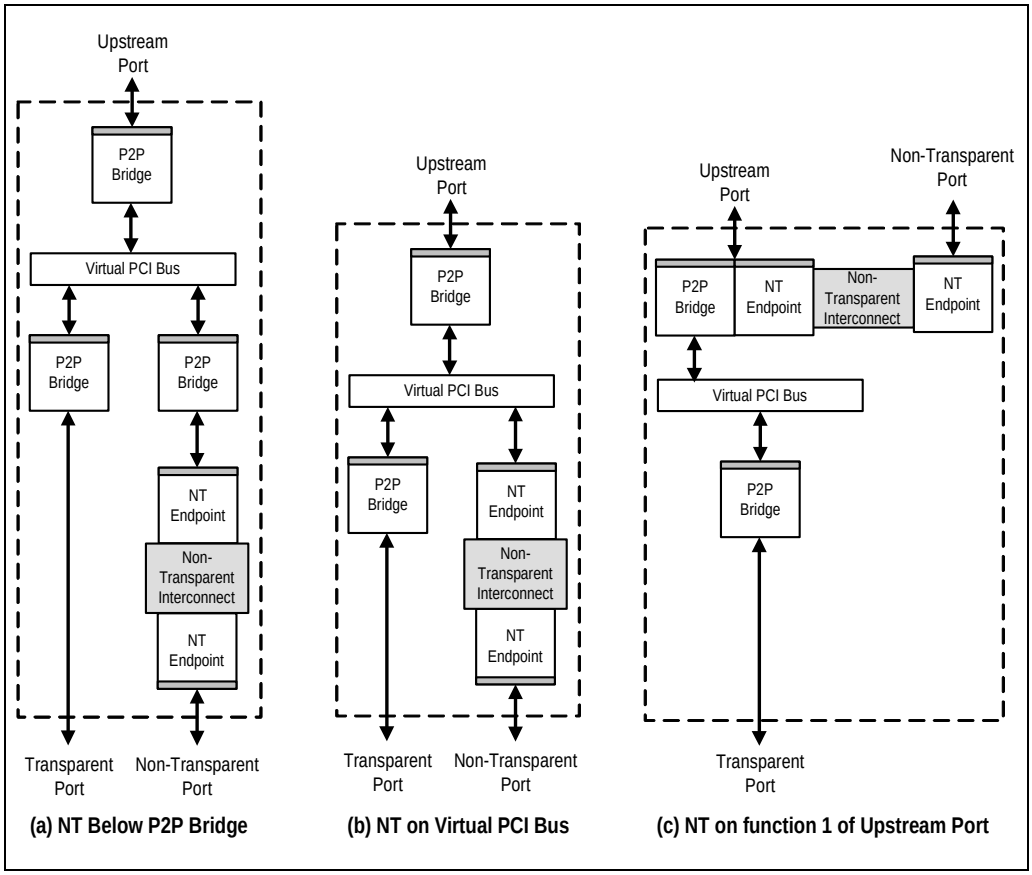

Figure 1.7 Architectural Approaches for Integrating Non-Transparency into a PCI Express Switch

As described in section Port Operating Modes on page 1-3, a switch port may be configured to operate with an NT endpoint function. The following port operating modes allow non-transparent operation on the port.

- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function port
- NT and DMA function port.

Figure 1.8 illustrates a basic non-transparent switch configuration. In this configuration, the switch ports are split into two partitions. Each partition represents a three-port transparent PCI Express switch. The upstream port of each partition is configured to operate as an *upstream switch port with NT endpoint*.

This configuration allows direct partition to partition communications without consuming external switch ports or links.

The NT endpoints in Figure 1.8 communicate using the NT interconnect. This allows PCI Express functions in either domain to communicate using the address windows presented by the NT endpoint BARs. Functions may be connected to the upstream port (e.g., the root) or to a downstream switch port. Upstream port TLPs flow directly to the corresponding NT endpoint. Downstream switch port TLPs flow through the corresponding three-port transparent switch and then back to the NT endpoint via the upstream port.

TLPs flowing from the secondary side of an upstream port's PCI-to-PCI bridge, through the bridge, to the NT endpoint stay entirely within the switch and are not transmitted on the upstream port's link. This is referred to as an inter-function transfer among functions (e.g., PCI-to-PCI bridge function and NT function) in the upstream port.

Notes

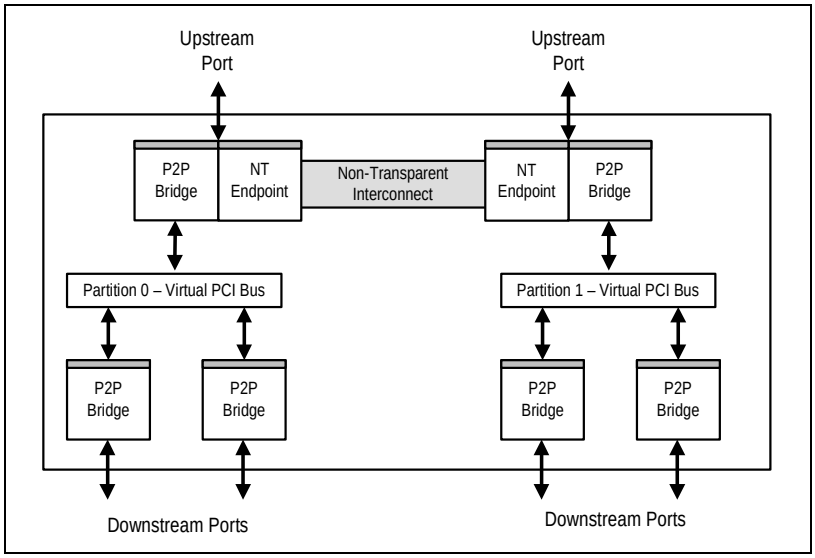

Figure 1.8 Non-Transparent Switch with Non-Transparency Between Partitions

Figure 1.9 illustrates a basic non-transparent switch configuration with NT ports. In this configuration, the switch ports are split into four partitions. The first partition, partition 0, represents a three-port transparent PCI Express switch. The remaining three partitions consist of the three NT endpoints[1]. Requesters in any of the NT domains may communicate using the address windows presented by the NT endpoint BARs.

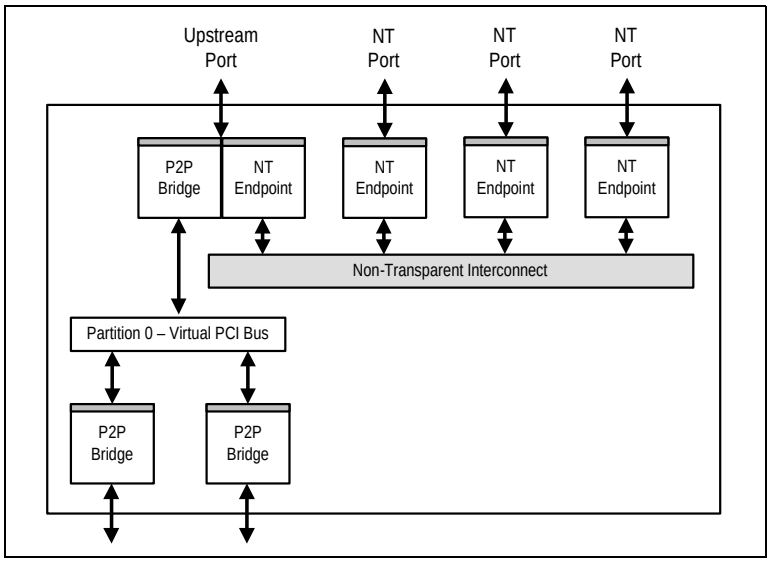

Figure 1.9 Non-Transparent Switch with Non-Transparent Ports

Figure 1.10 illustrates the switch configuration in which all ports are configured as NT endpoints. Such a configuration may be useful in bladed systems.

1. A port configured in NT function mode logically consists of only an NT endpoint and represents a switch partition.

Notes

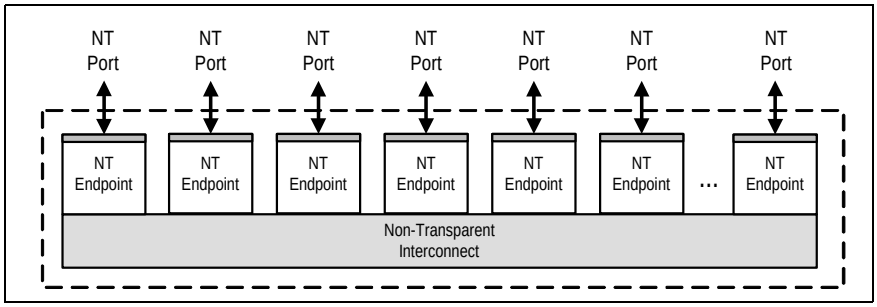

Figure 1.10 Non-Transparent Switch with Non-Transparent Ports

This section outlined several possible switch NTB configurations. The ability to configure ports to operate in a variety of modes together with support for switch partitioning provides the PES32NT8xG2 with the flexibility required for a wide variety of system applications.

Figure 1.11 illustrates a switch configuration with three transparent switch partitions and four NT port partitions. In this example, non-transparent communication is supported between all partitions except partition zero.

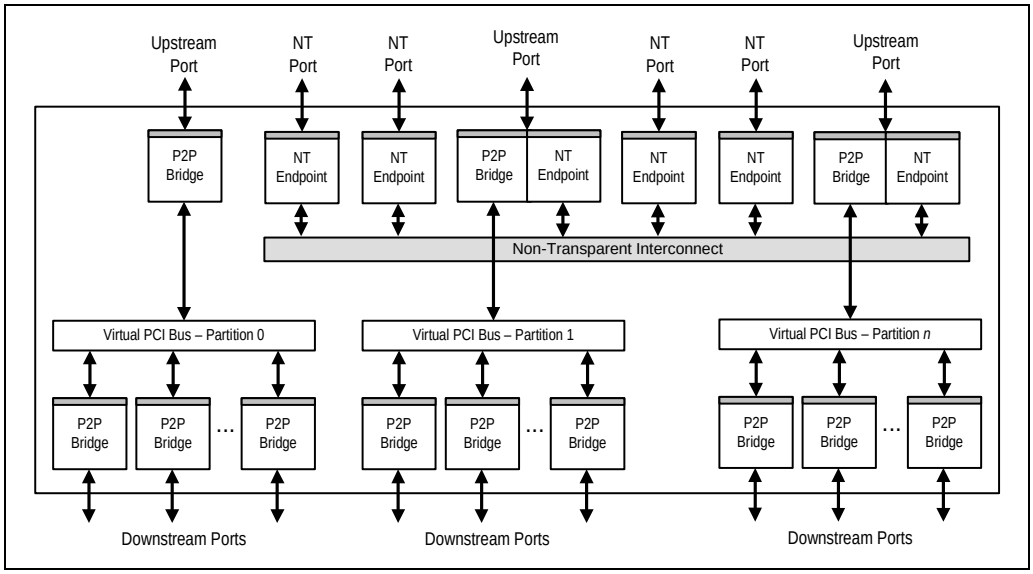

Figure 1.11 Non-Transparent Switch with Non-Transparent Ports

The switch's non-transparent operation is described in detail in Chapter 14.

DMA Operation

The PES32NT8xG2 supports two Direct Memory Access controller (DMA) functions. Each DMA function appears as a PCI Express endpoint in the PCI Express hierarchy, located in a switch partition's upstream port. In each partition, the operating mode of the switch's upstream port determines if this port contains a DMA function. The following port operating modes include a DMA function.

- Upstream switch port with DMA function
- Upstream switch port with NT and DMA functions
- NT with DMA function.

There can be at most one DMA function in a PES32NT8xG2 switch partition. Therefore, the PES32NT8xG2 allows up to two switch partitions to be configured to include a DMA function.

Notes

A DMA function is associated with two DMA channels. A DMA channel is an engine that can be programmed to transfer data between two PCI Express functions in the hierarchy, including transfers across the non-transparent bridge (see section Non-Transparent Operation on page 1-7). DMA channels act independently and operate by processing descriptors. DMA channels are programmed via configuration registers in the DMA function's configuration space. These configuration registers may be mapped to memory space using a Base Address Register (BAR) in the DMA function's configuration space.

Having two DMA channels per DMA function allows concurrent bi-directional data transfers among devices in the PCI Express hierarchy (e.g., one channel can be used to transfer data in one direction, while the other can be used to transfer data in another direction). Channels operate independently and can be programmed with different source and destination locations.

A DMA channel operates by fetching descriptors from a programmed memory address, and processing the descriptors. Descriptors may be organized as descriptor lists, which the DMA automatically processes until it reaches the end of the list. Descriptor processing typically involves reading data from a programmed memory address into the DMA function, converting the received completion TLPs into memory write TLPs, and issuing the memory write TLPs to write the data to another programmed memory address. The conversion step is done on the fly to minimize latency (i.e., the DMA need not read and buffer all the data prior to writing it to the target location). Once processing is completed, the DMA may be configured to issue an interrupt to the system.

Figure 1.12 shows the logical view of a switch partition with a DMA function in the upstream port. In this configuration, the DMA may be used to transfer data between devices connected (directly or indirectly via PCI Express) to any of the ports in the switch partition.

Memory read or write TLPs issued by the DMA function that are claimed by the upstream port's PCI-to-PCI bridge function (i.e., fall in the PCI-to-PCI bridge function base/limit memory windows) are routed across the bridge function into the switch partition's virtual PCI bus and sent towards the appropriate downstream port. Such TLPs are not emitted on the upstream link (i.e., there is a inter-function transfer between the DMA function and the PCI-to-PCI bridge function in the upstream port). If the TLP issued by the DMA function is not claimed by the PCI-to-PCI bridge function, then it is emitted on the upstream link.

Similarly, TLPs flowing from the secondary side of an upstream port's PCI-to-PCI bridge, through the bridge, to the DMA function stay entirely within the switch and are not transmitted on the upstream port's link.

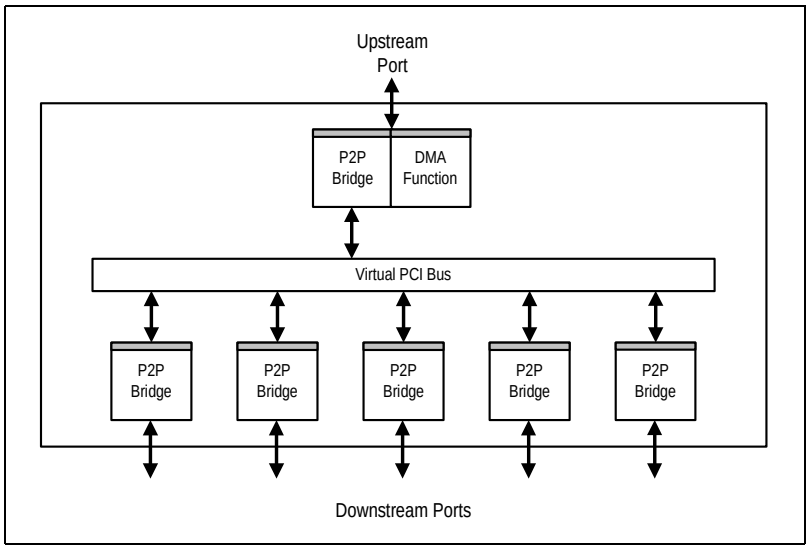

Figure 1.12 Switch Partition with DMA function

Figure 1.13 shows the logical view of two switch partitions interconnected via an NTB, with a DMA function in the upstream port of one partition. In this configuration, the DMA may be programmed to transfer data between the two partitions. That is, the DMA can be used to read data from a memory address in

Notes

either partition and write data to a memory address in the other partition. To read or write data from the partition across the NTB (i.e., partition 1 in this example), the DMA need only be programmed to issue the read/write transactions to addresses that map to one of the memory windows of the NT function in partition 0.

– Memory read or write request TLPs issued by the DMA function that are claimed by the PCI-to-PCI bridge function in the upstream port are routed as described in the previous example. TLPs issued by the DMA function that are claimed by the NT function (i.e., the TLP falls into one of the NT function's memory windows) are routed across the non-transparent interconnect and emitted by the NT function in the target partition.

In such a configuration, programming of the DMA would be typically done by an agent (e.g., the CPU) in the partition on which the DMA resides.[1] This would allow the programming agent to 'push' data from its partition to another partition, or 'pull' data from another partition into its partition. If symmetry is desired, the upstream port in both partitions could be programmed to have a DMA function, as shown in Figure 1.14. This allows agents in either partition to push or pull data from the other partition.

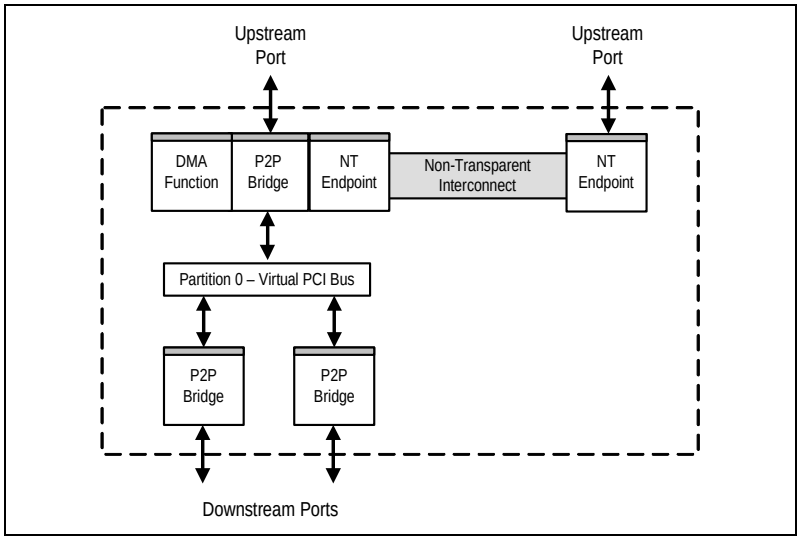

Figure 1.13 Two Switch Partitions Interconnected by an NTB, with DMA in One Partition

1. In the switch, the DMA may be programmed by agents that are not in the partition on which the DMA function resides. This is done by accessing the PES32NT8xG2's global address space (see Chapter 19 for details).

Notes

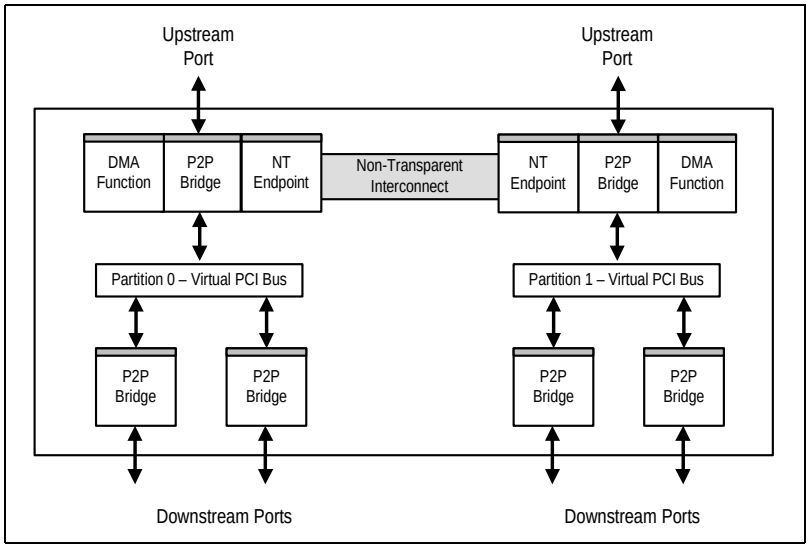

Figure 1.14 Two Switch Partitions Interconnected by an NTB, with DMA in Both Partitions

DMA transfers are always memory-mapped, and can therefore leverage the multicast feature offered by the PES32NT8xG2 (see section Multicasting and Non-Transparent Multicasting on page 1-17 for an introduction to Multicast support in the switch). The DMA function can be programmed to read data from a source memory-mapped location, and issue a multicast write operation to transfer the data to several memory-mapped destination locations in one shot. As described in section Multicasting and Non-Transparent Multicasting on page 1-17, the data can even be multicasted across switch partitions. The switch's DMA operation is described in detail in Chapter 15, DMA Controller.

Dynamic Reconfiguration and Failover

Dynamic reconfiguration refers to the modification of the PES32NT8xG2 switch configuration at runtime (i.e., after fundamental reset). The switch supports two forms of dynamic reconfiguration. The first is reconfiguration of the ports associated with a switch partition. The second is reconfiguration of the operating mode of a port. Partition and port reconfiguration may be initiated by software executing on a root complex or SMBus master, or initiated by hardware as the result of a failover event. The switch supports four failover configuration structures. Each configuration structure may be independently configured to initiate a failover event on:

- a configuration register write,
- watchdog timer time-out, or
- external device pin state transition.

An example failover operation is illustrated in Figure 1.15. Figure 1.15(a) illustrates a possible PES32NT8xG2 application with two partitions. Partition zero represents a transparent switch while partition one is an NT port with an alternate (secondary) root. The primary upstream port is able communicate with I/O device on downstream switch ports in a transparent manner. The primary upstream port is able to synchronize failover state information (e.g., recovery point data) with the secondary upstream port using the NT endpoints and non-transparent interconnect. Although not shown in Figure 1.15, it is possible to use the PES32NT8xG2's DMA function to off-load the root processor from this task. Downstream I/O devices are able to transfer data to the primary root and to the secondary root.

Notes

Consider an application that utilizes a watchdog timer to initiate failover. When the watchdog timer expires, a failover event is initiated. The failover event initiates the following actions to take place in hardware.

- The port associated with the primary upstream port is reconfigured to operate in NT function mode. The port's partition association is changed from partition 0 to partition 1.
- The port associated with the secondary upstream port is reconfigured to operate in upstream switch port with NT endpoint mode. The port's partition association is changed from partition 1 to partition 0.

Figure 1.15(b) illustrates the switch configuration following a failover event. The functionality previously associated with the primary upstream port is now associated with the secondary upstream port and vice-versa.

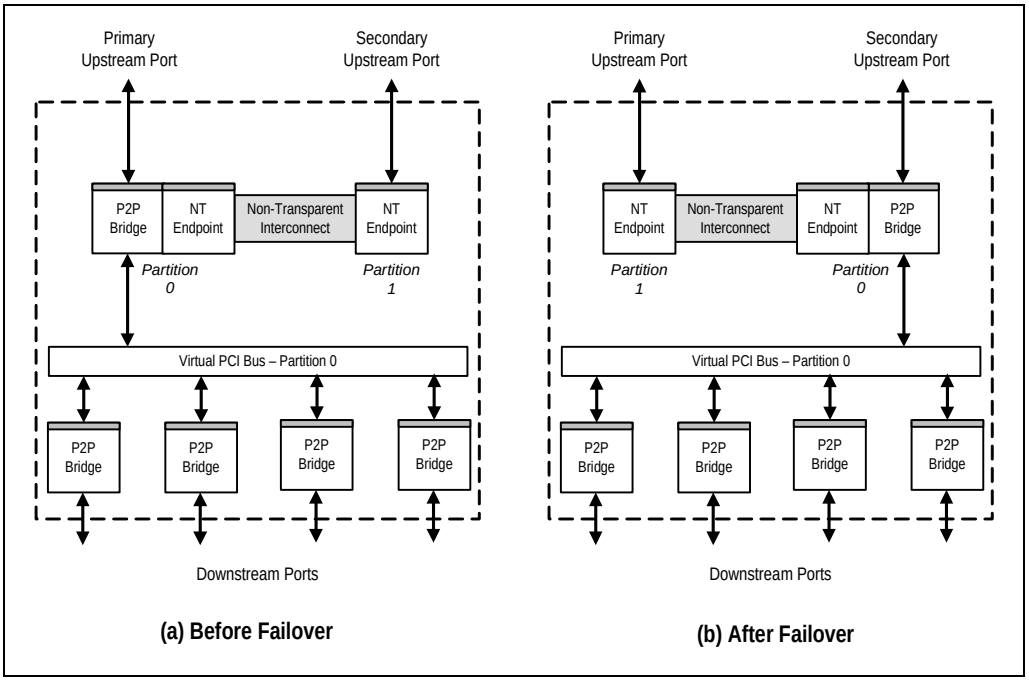

Figure 1.15 Non-Transparent Switch Failover Usage

Dynamic partition and port operating mode reconfiguration is described in section Port Operating Mode Change on page 5-13. Failover is described in is described in Chapter 6, Failover.

Switch Events

In a multi-partition switch, such as the PES32NT8xG2, a need may exist to signal the occurrence of certain events that occur within a partition to agents (e.g., a PCI Express function) in other partitions. For example, in a switch configuration with two or more partitions, the occurrence of a hot reset in a partition is an event that may be signaled to the root-complex in other partitions or to a switch management agent connected to yet another partition.

In this context, a switch management agent is a device in charge of managing the configuration and resources of the PES32NT8xG2 switch. A switch management agent may be a device connected to the switch via the SMBus interface or via a PCI Express link.

The PES32NT8xG2 contains a proprietary switch event mechanism that enables usage models where inter-partition event notification is desired. The switch event mechanism allows the notification of an event occurring in a partition to agents in other partitions. It is possible to configure which partitions are notified of the events. Notification is done via PCI Express interrupts (i.e., legacy interrupt or MSI) generated from the upstream port of the switch partition that received the notification.

Notes

The following switch events in a partition may be notified to other partitions:

- A switch port link going up (i.e., a transition from DL_Down to DL_Up)
- A switch port link going down (i.e., a transition from DL_Up to DL_Down)
- A switch port detecting an AER error
- A fundamental reset in a partition
- A hot reset in a switch partition
- Failover mode change initiated
- Failover mode change completed
- A global signal from a switch partition (see description of global signals below)

Figure 1.16 shows an example where a hot reset event in one partition is notified to other partitions via an interrupt. Note that event notifications are only issued to agents connected (directly or indirectly via PCI Express) to a switch partition. Thus, such notification is not possible to devices that connect to the PES32NT8xG2 switch via the SMBus interface, or that connect to a port that operates in disabled or unattached mode (i.e., such ports are not considered to belong to a switch partition).

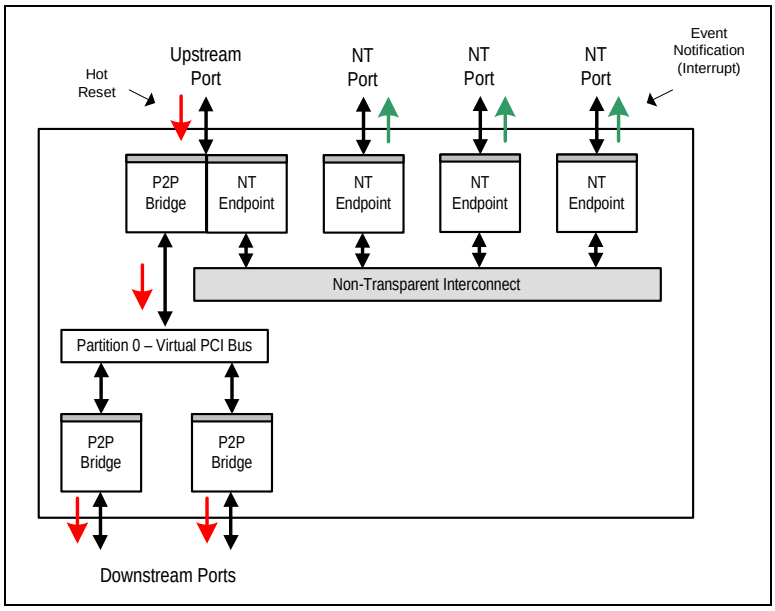

Figure 1.16 Example of Switch Event Mechanism

Global signal events allow an agent in a partition to issue a signal to agents in other partitions. A global signal is initiated when an agent in a partition writes to a specific register in the upstream port of the switch partition. This causes a switch event and it's corresponding notification to other partitions. In addition to the signaling, there are dedicated data registers that allow basic information passing between the agents (e.g., to indicate the reason behind the global signal event). Therefore, global signal events provide a basic form of inter-partition communication without involving Non-Transparent Bridging. Such communication may be used for coordinating switch reconfiguration actions between a switch management agent connected to a switch partition and agents in other partitions.

Switch event and signals are described in detail in Chapter 16, Switch Events.

Multicasting and Non-Transparent Multicasting

The PES32NT8xG2 implements multicast within switch partitions as defined by the PCI Express Base Specification 2.1. The term *transparent multicast* is used to refer to this type of multicast operation. In addition, the switch supports *non-transparent multicast*, using a proprietary implementation. This allows TLPs received by the NT endpoint in a partition to be multicasted to ports in other switch partitions. Transparent and non-transparent multicast may operate concurrently within a switch partition.

Notes

Using transparent multicast, a posted TLP (e.g., a memory write TLP) received by a port in a switch partition can be multicasted to other ports within that switch partition. Figure 1.17 shows an example of transparent multicast. In this example, a posted TLP received by the upstream port is multicasted to two of the downstream ports. Multicast is not restricted to upstream-to-downstream transfers. A TLP received on any port may be multicasted to other ports within that partition.

As defined in the PCI Express Base Specification 2.1, multicasting occurs when the received TLP falls within a programmed address window (i.e., the multicast BAR). Ports that serve as multicast egress ports may be grouped, and each group is associated with a segment of the multicast address window. The PES32NT8xG2 supports up to 64 multicast groups, the maximum allowed by the PCI Express Base Specification 2.1. In addition, the switch supports multicast address overlay, a feature defined as optional in the PCI Express spec, that allows re-mapping of the memory address in the multicast TLPs to a programmable address range. Address overlay is important as it allows multicast operation with non multicast-aware endpoints.

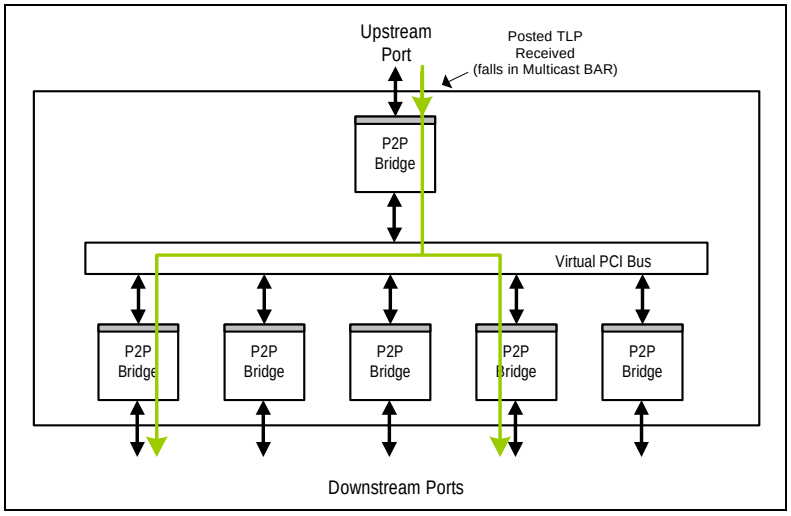

Figure 1.17 Example of Transparent Multicast

In addition to transparent multicast, the PES32NT8xG2 supports non-transparent multicast (a.k.a. NT multicast). NT multicast is a proprietary feature that allows TLPs received by the NT endpoint in a partition to be simultaneously transferred to one or more ports in other switch partitions. This improves performance in systems in which data in a switch partition needs to be distributed to other partitions.

Figure 1.18 shows an example of an NT multicast transfer. In this example, a TLP received by an NT endpoint is NT multicasted and transmitted by ports located in other partitions. Such a configuration may be found in multiprocessor systems in which multiple CPUs need to exchange data or state associated with a distributed computation. The switch's non-transparent interconnect can be used to interconnect the CPUs, and NT multicast improves the performance in sharing the data among the CPUs (i.e., the data need not be unicasted one destination at a time).

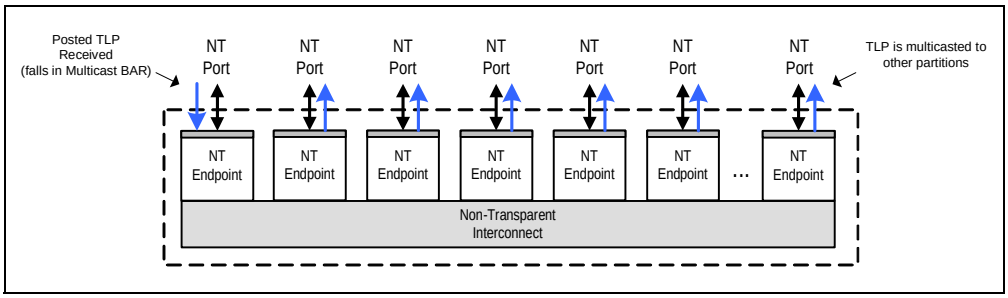

Figure 1.18 Example of Non Transparent Multicast

Notes

The programming model of NT multicast mimics that of transparent multicast, with a few exceptions. In particular, NT multicast has a proprietary address and requester ID overlay feature, that allows the TLP's address and requester ID to be modified when emitted by the egress ports. Such modifications are necessary to ensure that TLP is routed correctly in the targeted partitions.

Transparent and non-transparent multicast are described in detail in Chapter 17, Multicast.

Notes

Clocking

Notes

Overview

Figure 2.1 provides a logical representation of the PES32NT8xG2 clocking architecture. The switch has two differential global reference clock input (GCLK) pairs as well as several differential reference clock inputs (PxCLK) used for local port clocking.

The differential global reference clock input (GCLK) is driven into the device on the GCLKP[1:0] and GCLKN[1:0] pins. The nominal frequency of the global reference clock input may be selected by the Global Clock Frequency Select (GCLKFSEL) pin to be either 100 MHz or 125 MHz (+/- 300 ppm). Both global reference clock differential inputs should be driven with the same frequency. However, there are no skew requirements between the GCLKP[0]/GCLKN[0] and GCLKP[1]/GCLKN[1] inputs. Any constant phase difference is acceptable.

The global reference clock input is provided to each SerDes quad and to an on-chip PLL. The on-chip PLL uses this clock to generate a 250 MHz core clock that is used by internal switch logic (e.g., switch core, portion of a stack, etc.). The PLL within each SerDes quad generates a 5.0 GHz clock used by the SerDes analog portion (PMA) and a 250 MHz clock used by the digital portion (PCS).

Associated with some ports is a port reference clock input (PxCLK). Depending on the port clocking mode (see section Port Clocking Modes on page 2-3), a differential reference clock is driven into the device on the corresponding PxCLKP and PxCLKN pins.

Note: The nominal frequency of a port reference clock input (PxCLK) is 100 MHz (+/- 300 ppm), except in cases where the restrictions outlined in section Port Clocking Mode Selection on page 2-6 apply. The PxCLK supports SSC as described in section Support for Spread Spectrum Clocking (SSC) on page 2-5.

There are no skew requirements between the global clock input and a port reference clock input or between any of the port reference clock inputs. Any constant phase difference is acceptable.

The clocking architecture for the PES32NT8AG2 device is shown in Figure 2.1. This device has a port clock for each of the 8 ports. The clocking architecture for the PES32NT8BG2 device is shown in Figure 2.2. The “B” device supports on 3 port clocks (Ports 0, 2, and 4).

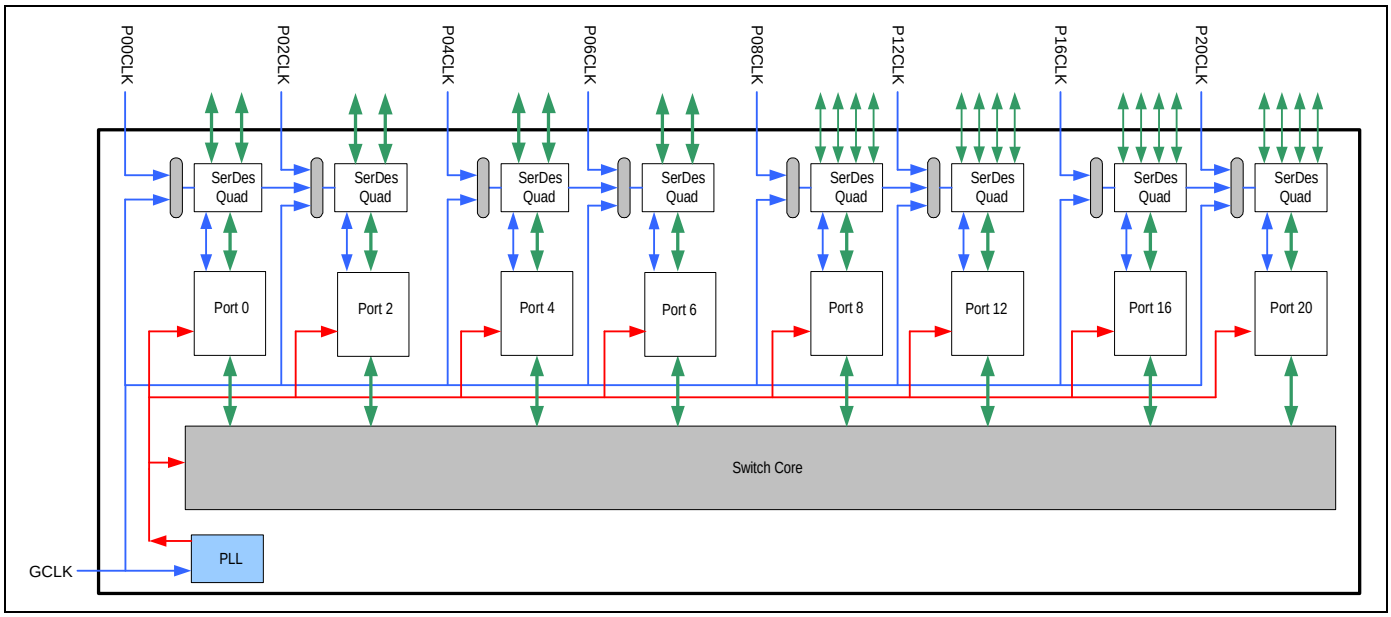

Figure 2.1 Logical Representation of PES32NT8AG2 Clocking Architecture

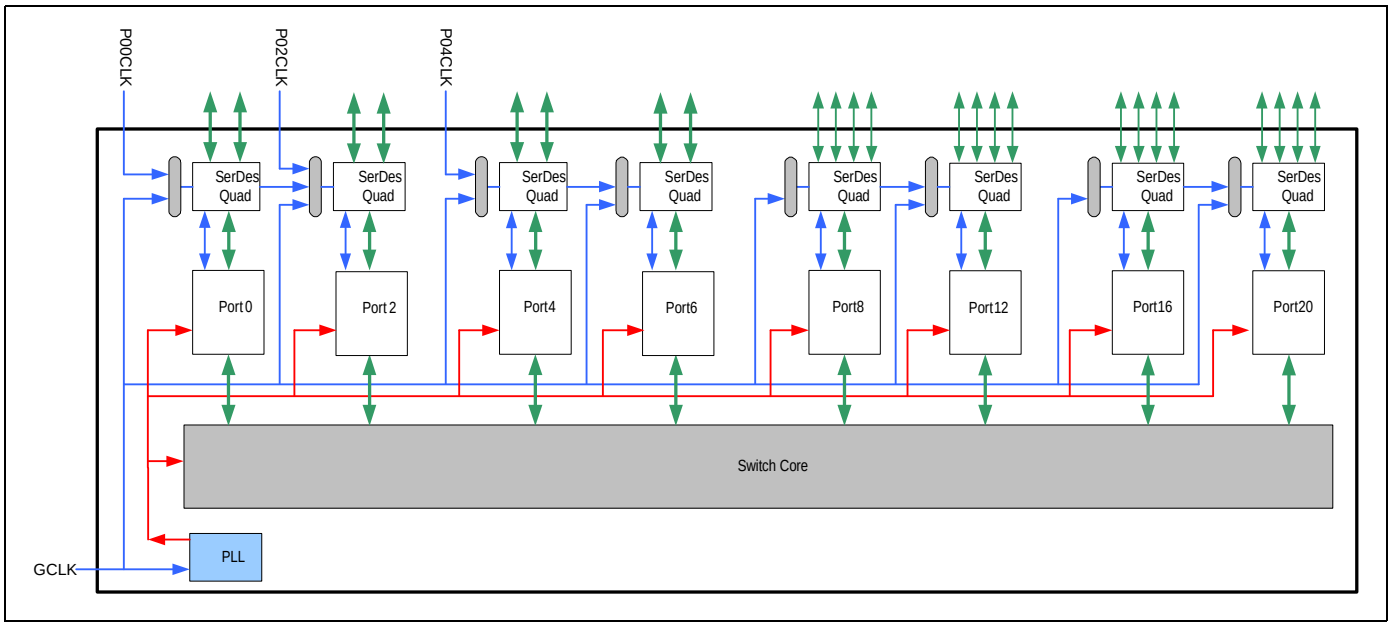

Figure 2.2 Logical Representation of PES32NT8BG2 Clocking Architecture

Notes

Port Clocking Modes

Port clocking refers to the clock that a port uses to receive and transmit serial data. The PES32NT8xG2 ports support two port clocking modes: Global Clocked and Local Port Clocked. These modes are described in section Global Clocked Mode on page 2-3 and section Local Port Clocked Mode on page 2-4.

Ports are not required to all operate in the same port clocking mode. Ports that do not share a SerDes quad may operate in different clocking modes.

Global Clocked Mode

A port in global clocked mode uses the global reference clock (GCLK) input for receiving and transmitting serial data. The port clock (PxCLK) associated with such a port (if any) is unused by the port. If no other port uses that same PxCLK, the PxCLK pins should be connected to Vss on the system board.

A port in this mode does not introduce any requirements on the global reference clock input beyond those imposed by PCI Express. Depending on the system configuration, a port in this mode may employ the common reference clock or separate (i.e., non-common) reference clock architectures defined by the PCI Express Base Specification 2.1.

Each port may independently be configured for common or non-common reference clock configuration. The grouping of ports shown in Table 2.1 above does not constrain this. Figure 2.3 shows the clock connection between a PES32NT8xG2 port and it's link partner, when the the switch port operates in global clocked mode with a common clock configuration.

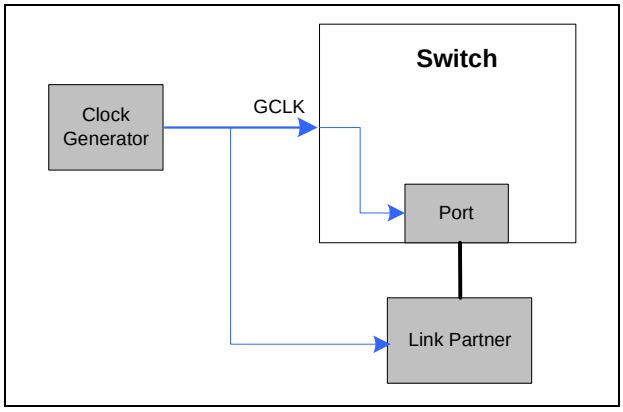

Figure 2.3 Clocking Connection for a Port in Global Clocked Mode, with a Common Clocked Configuration

Figure 2.4 shows the clock connection between a PES32NT8xG2 port and it's link partner, when the switch port operates in global clocked mode with a non-common clock configuration.

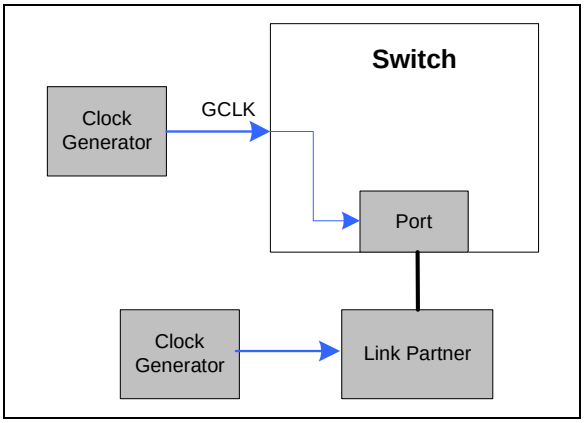

Figure 2.4 Clocking Connection for a Port in Global Clocked Mode, Non-Common Clocked Configuration

Notes

Local Port Clocked Mode

A port in local port clocked mode uses a dedicated port clock (PxCLK) input for receiving and transmitting serial data. Table 2.1 lists the ports and the PxCLK used by each.

Ports	PxCLK used when port operates in Port Clocked Mode[1]
0	P00CLK
2	P02CLK
4	P04CLK
6	P06CLK
8	P08CLK
12	P12CLK
16	P16CLK
20	P20CLK

Table 2.1 PxCLK Usage When a Port Operates in Local Port Clocked Mode

1. Note that in the PES32NT8BG2, the device supports port clocking only for ports 0, 2, and 4.

Depending on the system configuration, a port in this mode may employ the common reference clock or non-common reference clock architectures defined by the PCI Express Base Specification 2.1. Each port may independently be configured for common or non-common reference clock configuration. The grouping of ports shown in Table 2.1 above does not constrain this.

Local port clocked mode allows a port to use a reference clock that is separate from the global reference clock (GCLK) used by the switch or the reference clock used by other ports. As described in section Support for Spread Spectrum Clocking (SSC) on page 2-5, this separate reference clock can have Spread Spectrum Clocking (SSC). Therefore, local port clocked mode allows the use of the PES32NT8xG2 in system configurations where one or more switch ports operate with independent reference clocks, and SSC is desired on these clocks.

Figure 2.5 shows the clock connection between a PES32NT8xG2 port and it's link partner, when the switch port operates in local port clocked mode with a common clock configuration.

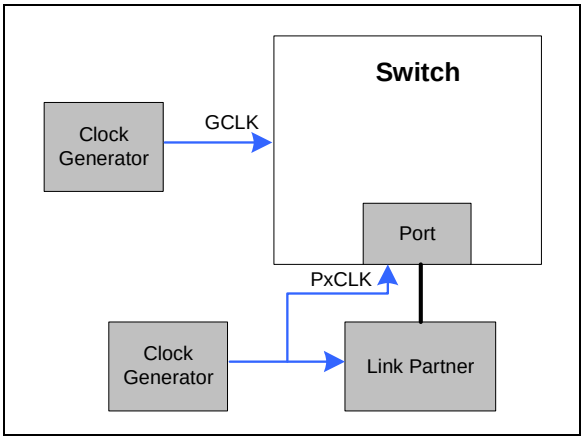

Figure 2.5 Clocking Connection for a Port in Local Port Clocked Mode, in a Common Clocked Configuration

Notes

Figure 2.6 shows the clock connection between a PES32NT8xG2 port and it's link partner, when the switch port operates in local port clocked mode with a non-common clock configuration.

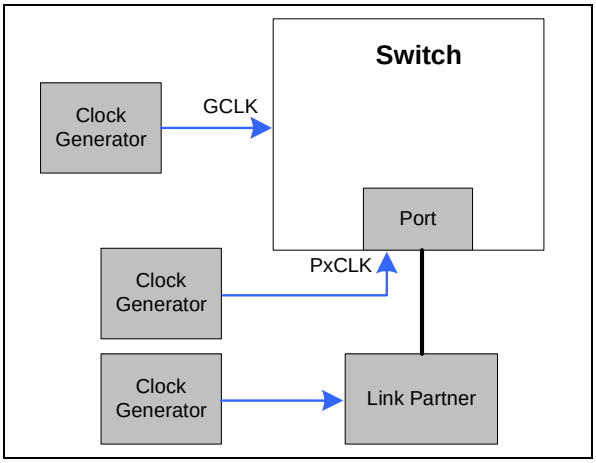

Figure 2.6 Clocking Connection for a Port in Local Port Clocked Mode, in a Non-Common Clocked Configuration

Depending on the stack configuration, some ports may be inactive (see section Stack Configuration on page 3-5). The PxCLK clock associated with an inactive port is unused by the hardware and its pins should be connected to ground. For example, if port 0 is configured as x8, ports 1, 2, and 3 become inactive (since they share the same stack). When configured for local port clocking, port 0 uses P00CLK as it's reference clock. P02CLK becomes unused by the hardware and this clock should be connected to Vss on the system board.

Support for Spread Spectrum Clocking (SSC)

The PES32NT8xG2 supports Spread Spectrum Clocking (SSC) for ports operating in the global clocked or local port clocked modes. The use of SSC is optional. To use SSC, the following requirements apply.

- If the GCLK has SSC, then all ports must operate in global clocked mode and be configured in a common clocked configuration with their link partners.
- If the GCLK does not have SSC, then a port may be configured in global clocked mode or local port clocked mode.
- If a port is operating in local port clocked mode and the port's local clock (PxCLK) has SSC, the following must be met:
  - The port must operate in a common-clocked configuration with its link partner.
  - The global reference clock input (GCLK) must not use SSC.
  - The GCLK's frequency must be equal to or faster than the PxCLK's frequency. This statement does not include the nominal +/-300 ppm deviations on either of these clocks. For example, the GCLK's frequency may be 100 Mhz - 300ppm while the PxCLK's frequency is 100 Mhz + 300ppm. In addition, frequency increases, if any, introduced by the SSC component on PxCLK must be correspondingly added to the GCLK frequency. Table 2.2 shows some allowed GCLK and PxCLK combinations.

Notes

Nominal PxCLK Frequency	PxCLK SSC Modulation	Effective PxCLK Frequency	Allowed GCLK Frequency
100 MHz + 300ppm	+0 / - 5000ppm	100 Mhz +300 / -4700ppm	100 Mhz + / -300ppm
100 MHz - 300ppm	+0 / - 5000ppm	100 Mhz -300 / -5300ppm	100 Mhz + / -300ppm
100 MHz + 300ppm	+0 / - 5000ppm	100 Mhz +300 / -4700ppm	125 Mhz + / -300ppm
100 MHz - 300ppm	+0 / - 5000ppm	100 Mhz -300 / -5300ppm	125 Mhz + / -300ppm

Table 2.2 GCLK and PxCLK frequencies when PxCLK has SSC

This results in the port clocking mode requirements summarized in Table 2.3.

Port Clocking Mode	Clock Used by Port for Transmitting and Receiving Data	Global Reference Clock Input Restrictions
Global Clocked	GCLK	none
Local Port Clocked	PxCLK	GCLK must not use SSC

Table 2.3 Port Clocking Mode Requirements

Port Clocking Mode Selection

The port clocking mode used by a port is determined by the corresponding Port Clocking Mode (PxCLKMODE) field in the Port Clocking Mode (PCLKMODE) register. The initial port clocking mode of a port is determined by the state of the CLKMODE[1:0] pins in the boot configuration vector as shown in Table 2.4. This signal also determines the initial value of the Slot Clock Configuration (SCLK) field in each port's PCI Express Link Status (PCIELSTS) register.

The SCLK field controls the advertisement of whether or not the port uses the same reference clock frequency as the link partner. The SCLK field may be modified by software (e.g., PCI Express configuration requests, EEPROM, etc.) on a per-port basis, thus allowing for common or non-common clocked configurations independently for each port. When the port operates in a multi-function mode (e.g., upstream switch port with NT function, NT with DMA function, etc.), the SCLK field reports the same value for all functions of the port.

Notes

CLKMODE[1:0] Value in Boot Configuration Vector	Port 0 Clocking Mode	Port 0 SCLK	Other Ports[1] Clocking Mode	Other Ports[1] SCLK
0	Global Clocked	0 (non-common clocked)	Global Clocked	0 (non-common clocked)
1	Global Clocked	1 (common clocked)	Global Clocked	0 (non-common clocked)
2	Global Clocked	0 (non-common clocked)	Global Clocked	1 (common clocked)
3	Global Clocked	1 (common clocked)	Global Clocked	1 (common clocked)

Table 2.4 Initial Port Clocking Mode and Slot Clock Configuration State

1. Other ports are 2, 4, 6 ,8, 12, 16, 20.

The port clocking mode associated with a port may be modified at any time, with the only requirement being that the reference clock that will be used by the port after the port's clocking mode is modified must be stable prior to the modification. Modifying a port's clocking mode causes the port's PHY to transition to the Detect state.

– This is not considered a surprise link down event.

If the clock mode change requires a modification of the reference clock associated with the port's SerDes, the SerDes is re-initialized. If the port is not in disabled mode, the PHY retrains the link. Modifying a port's clock mode is subject to the following restrictions outlined in Table 2.5 regarding the clock frequencies.

Initial Port Clock Mode	Subsequent Port Clock Mode Change by Programming the PCLKMODE Register	Limitations / Caveats
Global Clocked Mode	Local Port Clocked Mode	If the GCLK operates at a nominal frequency of 100 Mhz, the port's PxCLK must also operate at a nominal frequency of 100 Mhz. If the GCLK operates at a nominal frequency of 125 Mhz, the port's PxCLK must also operate at a nominal frequency of 125 Mhz. Per the rules outlined in section Support for Spread Spectrum Clocking (SSC) on page 2-5, this port mode change is only allowed when the GCLK does not have SSC. Still, the PxCLK is allowed to have an SSC component on top of the frequencies described above.
Local Port Clocked Mode	Global Clocked Mode	This port clock mode change is only allowed if the global clock (GCLK) frequency is 100 Mhz (as indicated by the GCLKFSEL pin).

Table 2.5 Clock Frequency Limitations when Modifying a Port's Clock Mode

System Clocking Configurations

Based on the requirements outlined in the sections above, Table 2.6 summarizes valid system clocking configurations (highlighted in green). Invalid system configurations are highlighted in red.

PES32NT8xG2 Port Configuration			Link Partner Refclk	Valid Config.	Notes
Port Clocking Mode	Local Port Clock	Global Clock			
Global Clocked	don't care	GCLK	Same GCLK as the switch	Yes	Global clocked with common Refclk architecture (Figure 2.3)
Global Clocked	don't care	GCLK with SSC	Same GCLK as the switch	Yes	Global clocked with common Refclk architecture and SSC (same as Figure 2.3, with SSC on GCLK)
Global Clocked	don't care	GCLK	Different Refclk than the switch	Yes	Global clocked with separate Refclk architecture (Figure 2.4)
Global Clocked	don't care	GCLK with SSC	Different Refclk than the switch	No	Violates PCI Express +/- 300 ppm clock difference requirement
Local Port Clocked	PxCLK	GCLK	Same PxCLK as the switch	Yes	Local port clocked with common Refclk architecture (Figure 2.5)
Local Port Clocked	PxCLK	GCLK with SSC	Same PxCLK or different Refclk as the switch	No	Not supported
Local Port Clocked	PxCLK with SSC	GCLK	Same PxCLK as the switch	Yes	Local port clocked with common Refclk architecture and SSC (same as Figure 2.5, with SSC on PxCLK and no SSC on GCLK)
Local Port Clocked	PxCLK with SSC	GCLK with SSC	Same PxCLK or different Refclk as the switch	No	Not supported
Local Port Clocked	PxCLK	GCLK	Different Refclk than the switch	Yes	Local port clocked with separate Refclk architecture (Figure 2.6)
Local Port Clocked	PxCLK with SSC	GCLK	Different Refclk than the switch	No	Violates PCI Express +/- 300 ppm clock difference requirement

Table 2.6 Valid PES32NT8xG2 System Clocking Configurations

Reset and Initialization

Notes

Overview

This chapter describes the PES32NT8xG2 resets and initialization. There are two classes of switch resets. The first is a *switch fundamental reset* which is the reset used to initialize the entire device. The second class is referred to as *partition resets*. This second class has multiple sub-categories. Partition resets are associated with a specific PES32NT8xG2 switch partition and correspond to those resets defined in the PCI Express base specification (e.g., fundamental reset, hot reset, etc). Switch resets are described in section Switch Fundamental Reset on page 3-2 while partition resets are described in section Partition Resets on page 3-10.

When multiple resets are initiated concurrently, the precedence shown in Table 3.1 is used to determine which one is acted upon.

- Reset types and causes are described in detail in the following sections.
  - A switch fundamental reset affects the entire device
  - A partition reset affects the partition and ports associated with that partition
  - A port reset affects only that one port
- When a high priority and low priority reset are initiated concurrently and the condition causing the high priority reset ends prior to that causing the low priority reset, then the device/partition/port immediately transitions to the reset associated with low priority reset condition.
  - If the high priority and low priority resets share the same reset type, then the device/partition/port remains in the corresponding reset when the high priority reset condition ends.
  - If the high priority and low priority reset have different reset types, then the device/partition/port transitions to the low priority reset type when the high priority reset condition ends.

Priority	Reset Type	Reset Cause
1 (Highest)	Switch fundamental reset	Global reset pin (PERSTN) assertion
2	Port mode change reset	Port operating mode change and OMA field set to port reset in the corresponding SWPORTxCTL register
	Partition fundamental reset	Directed by STATE field value in SWPARTxCTL register
3	Partition fundamental reset	Assertion of partition fundamental reset pin (PARTxPERSTN)
4	Partition hot reset	Reception of TS1 ordered sets on upstream port indicating a hot reset
5	Partition hot reset	Data link layer of the upstream port transitioning to DL_Down state
6	Partition upstream secondary bus reset	Setting of the SRESET bit in the partition's upstream port PCI-to-PCI bridge BCTL register
7 (Lowest)	Partition downstream secondary bus reset	Setting of the SRESET bit in the in the corresponding port's PCI-to-PCI bridge BCTL register

Table 3.1 PES32NT8xG2 Reset Precedence

Notes

Switch Fundamental Reset

A switch fundamental reset may be cold or warm. A cold switch fundamental reset occurs following a device being powered-on and assertion of the global reset (PERSTN) signal. A warm switch fundamental reset occurs when a switch fundamental reset is initiated while power remains applied. The PES32NT8xG2 behaves in the same manner regardless of whether the switch fundamental reset is cold or warm.

A switch fundamental reset may be initiated by any of the following conditions.

- A cold switch fundamental reset initiated by application of power (i.e., a power-on) followed by assertion of the global reset (PERSTN) signal.

Note: Refer to the device data-sheet for power sequencing requirements.

- A warm switch fundamental reset initiated by assertion of PERSTN while power remains applied.

When a switch fundamental reset is initiated, the following sequence is executed.

1. Wait for the switch fundamental reset condition to clear (e.g., negation of PERSTN).
2. On negation of PERSTN, sample the boot configuration vector signals shown in Table 3.2.
3. All registers are initialized to their default value.
   - Partition and port configuration registers are initialized as dictated by the SWMODE value in the boot configuration vector (see section Switch Modes on page 3-8).
4. The Register Unlock (REGUNLOCK) bit is set in the Switch Control (SWCTL) register. This allows all register fields with type Read-Write-Locked (RWL) to be modified.
5. The on-chip PLL and SerDes are initialized (e.g., PLL lock).
6. The master SMBus interface is initialized.
7. The slave SMBus is taken out of reset and initialized. The slave SMBus address is specified by the SSMBADDR[2:1] signals in the boot configuration vector.
8. Within 20 ms after the switch fundamental reset condition clears, the reset signal to the stacks is negated and link training begins on all ports. While link training takes place, execution of the reset sequence continues.
9. Within 100 ms following clearing of the switch fundamental reset condition, the following occurs.
   - All ports that have PCI Express base specification compliant link partners have completed link training.
   - All ports are able to receive and process TLPs.
10. If the sampled Switch Mode (SWMODE) state corresponds to a mode that supports serial EEPROM initialization, then the contents of the serial EEPROM are read and appropriate switch registers are updated. Otherwise, this step is not executed.
    - Refer to section Serial EEPROM on page 12-2 for details on serial EEPROM initialization.
    - While the contents of the EEPROM are read, all ports enter a quasi-reset state. In quasi-reset state, each port responds to all type 0 configuration request TLPs with a configuration-request-retry-status completion[1]. All other TLPs are ignored (i.e., flow control credits are returned but the TLP is discarded).
    - If a one is written by the serial EEPROM to the Full Link Retrain (FLRET) bit in any Phy Link State 0 (PHYLSTATE0) register, then link retraining is initiated on the corresponding port using the current link parameters.
    - If an error is detected during loading of the serial EEPROM, then loading of the serial EEPROM is aborted and the RSTHALT bit is set in the SWCTL register. Error information is recorded in the SMBUSSTS register (refer to section Initialization from Serial EEPROM on page 12-3).

[1] This includes configuration requests to the port's Global Address Space Access and Data registers (GASAADDR and GASADATA). Type 1 configuration request TLPs are handled as unsupported requests.

Notes

– When serial EEPROM initialization completes, the EEPROM Done (EEPROMDONE) bit in the SMBUSSTS register is set and the switch's ports start processing configuration requests normally, unless the RSTHALT bit in the SWCTL register is set. If serial EEPROM initialization completes with an error, the RSTHALT bit in the SWCTL register is set as described in section Initialization from Serial EEPROM on page 12-3. In this case, the ports remain a quasi-reset state as described in step 11.

11. If the RSTHALT bit in the SWCTL register is set (e.g., due to the assertion of the RSTHALT signal in the sampled boot vector), all ports enter (or remain) in a quasi-reset state. Otherwise, this step is not executed.

– All ports remain in quasi-reset state until the Reset Halt (RSTHALT) bit is cleared by software in the SWCTL register. This provides a synchronization point for a device on the slave SMBus to initialize the device. When device initialization is completed, the slave SMBus device clears the RSTHALT bit allowing the device to begin normal operation.

12. The Register Unlock (REGUNLOCK) bit is cleared in the Switch Control (SWCTL) register.

13. Normal device operation begins.

The PCI Express Base Specification 2.1 indicates that a device must respond to configuration request transactions within 100ms from the end of Conventional Reset (cold, warm, or hot). Additionally, the PCI Express Base Specification indicates that a device must respond to configuration requests with a successful completion within 1.0 second after Conventional Reset of a device. The reset sequence above guarantees that the switch will be ready to respond successfully to configuration requests within the 1.0 second period as long as the serial EEPROM initialization process completes within 200 ms.

– Under normal circumstances, 200 ms is more than adequate to initialize registers in the device with a master SMBus operating frequency of 400 KHz.

Serial EEPROM initialization may cause writes to register fields that initiate side effects such as link retraining. These side effects are initiated at the point at which the write occurs. Therefore, serial EEPROM initialization should be structured in a manner so as to ensure proper configuration prior to initiation of these side effects.

The operation of a switch fundamental reset with serial EEPROM initialization is illustrated in Figure 3.1.

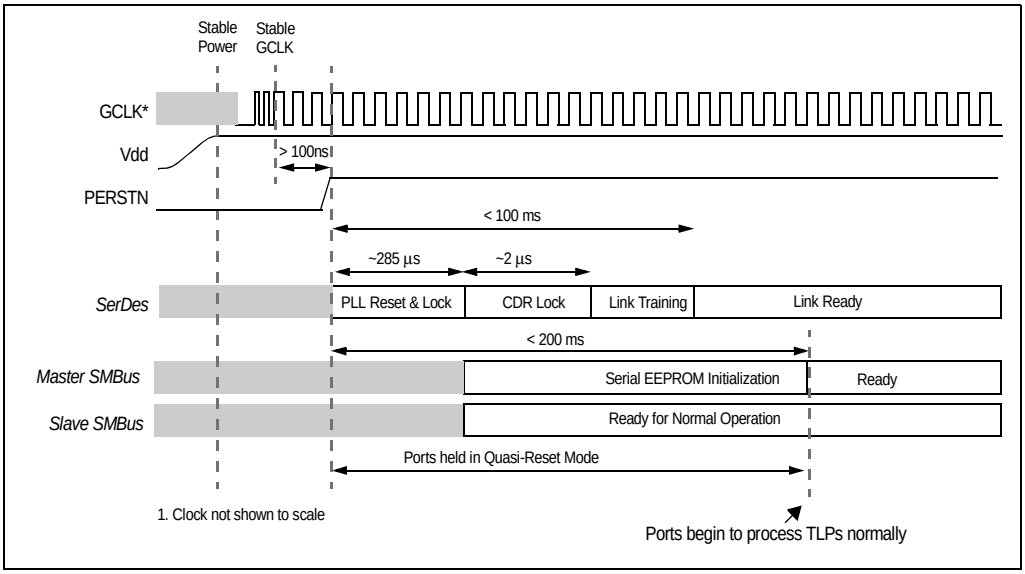

Figure 3.1 Switch Fundamental Reset with Serial EEPROM Initialization

Notes

The operation of a switch fundamental reset using RSTHALT is illustrated in Figure 3.2.

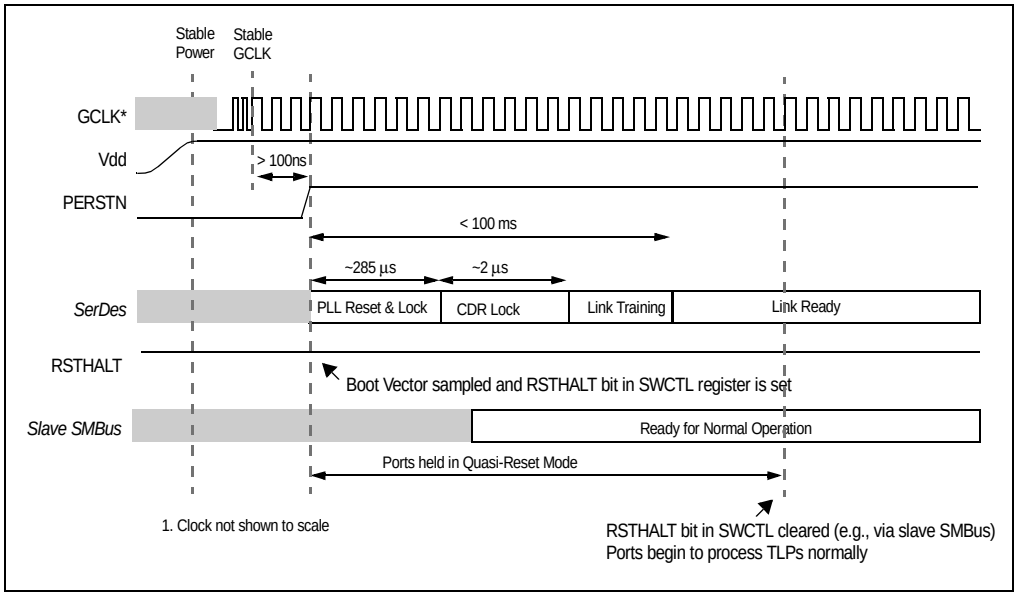

Figure 3.2 Fundamental Reset Using RSTHALT to Keep Device in Quasi-Reset State

Boot Configuration Vector

A boot configuration vector consisting of the signals listed in Table 3.2 is sampled during a switch fundamental reset. Since the boot configuration vector is only sampled during a switch fundamental reset, the state of the signals that make up the boot configuration vector is ignored outside of a switch fundamental reset sequence.

While basic switch operation may be configured using signals in the boot configuration vector, advanced switch features require more complex initialization. As noted in table Table 3.2, some of the initial values specified by the boot configuration vector may be overridden by software, serial EEPROM, or an external SMBus device.

- See section Slave SMBus Interface on page 12-19 for a description of the slave SMBus interface.
- See section Initialization from Serial EEPROM on page 12-3 for a description of the serial EEPROM operation.

The state of all of the boot configuration signals in Table 3.2 sampled during a switch fundamental reset may be determined from the Boot Configuration Status (BCVSTS) register.

Notes

Signal	May Be Overridden	Name/Description
GCLKFSEL	N	**Global Clock Frequency Select.** These pins specify the frequency of the GCLKP and GCLKN signals.
CLKMODE[1:0]	Y	**Clock Mode.** These pins specify the clocking mode used by switch ports. See Table 2.4 for a definition of the encoding of these signals. The value of these signals may be overridden by modifying the Port Clocking Mode (PCLKMODE) register.
RSTHALT	Y	**Reset Halt.** When this pin is asserted during a switch fundamental reset sequence, the switch remains in a quasi-reset state with the Master and Slave SMBuses active. This allows software to read and write registers internal to the device before normal device operation begins. The device exits the quasi-reset state when the RSTHALT bit is cleared in the SWCTL register by an SMBus master. Refer to section Switch Fundamental Reset on page 3-2 for further details.
SSMBADDR[2:1]	N	**Slave SMBus Address.** SMBus address of the switch on the slave SMBus.[1]
SWMODE[3:0]	N	**Switch Mode.** These pins specify the switch operating mode. See section Switch Modes on page 3-8.
STK0CFG[0]	Y	**Stack 0 Configuration.** This pin selects the configuration of stack 0 during a switch fundamental reset. Refer to section Stack Configuration on page 3-5 for further details.
STK1CFG[0]	Y	**Stack 1 Configuration.** This pin selects the configuration of stack 1 during a switch fundamental reset. Refer to section Stack Configuration on page 3-5 for further details.
STK2CFG[0]	Y	**Stack 2 Configuration.** This pin selects the configuration of stack 2 during a switch fundamental reset. Refer to section Stack Configuration on page 3-5 for further details.
STK3CFG[0]	Y	**Stack 3 Configuration.** This pin selects the configuration of stack 3 during a switch fundamental reset. Refer to section Stack Configuration on page 3-5 for further details.

Table 3.2 Boot Configuration Vector Signals

1. Note that the PES32NT8BG2 supports only SSMBADDR2.

Refer to the PES32NT8xG2 Data Sheet for additional information on the signals that make up the boot configuration vector.

Stack Configuration

As shown in Figure 1.1, the switch contains four stack blocks labeled Stack 0, Stack 1, Stack 2, and Stack 3. Each stack has two ports for a total of 8 ports in the device. Table 3.3 lists the ports associated with each stack.

Notes

Stack	Ports Associated with the Stack
Stack 0	0, 2
Stack 1	4, 6
Stack 2	8, 12
Stack 3	16, 20

Table 3.3 Ports in Each Stack

Each stack may be configured as one x8 port or two x4 ports[1]. The configuration of each stack is controlled by the Stack Configuration (STK[3:0]CFG) registers. These registers are located in the Switch Configuration and Status space (see Chapter 19). Tables 3.4 through 3.7 show the possible configurations for each stack.

- Each STKxCFG register controls the configuration of the corresponding stack (e.g., STK0CFG controls the configuration of Stack 0, STK1CFG for Stack 1, etc.)
- Stack configurations not shown in the table are not allowed. Programming the STKxCFG register to values not shown in the table produces undefined results.

STKCFG Field in the STK0CFG Register		Stack Configuration	
Hex	Binary	Port 2	Port 0
0x0	0b00000		x8
0x1	0b00001	x4	x4
Others		Reserved	

Table 3.4 Possible Configurations for Stack 0

STKCFG Field in the STK1CFG Register		Stack Configuration	
Hex	Binary	Port 6	Port 4
0x0	0b00000		x8
0x1	0b00001	x4	x4
Others		Reserved	

Table 3.5 Possible Configurations for Stack 1

1. In the PES32NT8xG2 device, x4 ports cannot be bifurcated.

Notes

STKCFG Field in the STK2CFG Register		Stack Configuration	
Hex	Binary	P12	P8
0x0	0b00000		x8
0x1	0b00001	x4	x4
Others		Reserved	

Table 3.6 Possible Configurations for Stack 2

STKCFG Field in the STK3CFG Register		Stack Configuration	
Hex	Binary	P20	P16
0x0	0b00000		x8
0x1	0b00001	x4	x4
Others		Reserved	

Table 3.7 Possible Configurations for Stack 3

Depending on the stack configuration, one port in the stack may be 'activated' and other 'de-activated'. For example, when the STKCFG field STK0CFG register is configured to 0x0, port 0 is activated and port 2 is de-activated. A de-activated port has the following behavior:

- All output signals associated with the port are placed in a negated state (e.g., link status and hot-plug signals).
  - The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.
  - PxACTIVEN and PxLINKUPN are negated.
- All input signals associated with the port are ignored and have no effect on the operation of the device.
  - The state of the following hot-plug input signals is ignored: PxAPN, PxMRLN, PxPDN, PxPFN, and PxPWRGDN.
- The port is not associated with a PCI Express link. PCI Express configuration requests targeting the port are not possible and the port is not part of the PCI Express hierarchy.
- The port is not associated with any switch partition. The port is unaffected by the state of any switch partition, and vice-versa.
- Unused logic is placed in a low power state.
- All registers associated with the port remain accessible from the global address space.[1]
- The port remains in this state regardless of the setting of the port's operating mode (i.e., via the port's SWPORTxCTL register).

An activated port behaves as described throughout the rest of this manual and may be configured in one of several operating modes, as described in Chapter 5.

[1] Refer to Chapter 19, Register Organization, for details on the switch's global address space.

Notes

Static Configuration of a Stack

A stack may be configured statically using the corresponding Stack Configuration (STKxCFG) pins. These pins are sampled by the switch as part of the boot-configuration vector during switch fundamental reset. The STKxCFG pins determine the initial value of the STKCFG field in the corresponding STKxCFG register. The encoding of the STKxCFG pins is identical to that of the STKCFG field shown in Tables 3.4 through 3.7.

– For all stacks, the STKxCFG pins have 1 bit each (i.e., STK0CFG[0] and STK1CFG[0]). This bit corresponds to the least significant bit of the STKCFG field in the corresponding STKxCFG register. Therefore, for these stacks, configurations 0x0 and 0x1 may be selected statically.

Dynamic Reconfiguration of a Stack via EEPROM / SMBus

In addition to static configuration as described above, each stack may be reconfigured via the EEPROM or SMBus slave interface during the switch fundamental reset sequence (i.e., at the EEPROM loading step or via the slave SMBus interface when the ports are in quasi-reset state[1]).

Dynamic reconfiguration of a stack requires the following procedure.

1. The operating mode of all ports associated with the stack must be set to disabled (see Chapter 5, Switch Partition and Port Configuration).
2. The stack must be reconfigured by programming the STKCFG field in the corresponding STKxCFG register.
3. The operating mode of the ports associated with the stack must be set as desired. For example, some ports in the stack may be set to operate in downstream switch mode and others in upstream switch mode.

Dynamic reconfiguration of a stack through other methods (i.e., through PCI Express configuration requests or via SMBus after the fundamental reset sequence completes) is not supported.

Switch Modes

The Switch Mode pins (SWMODE[3:0]) sampled during switch fundamental reset determine the mode of operation and initial configuration of the PES32NT8xG2 switch at boot time. Switch modes may be subdivided into normal switch modes and test modes. Normal switch modes are listed in Table 3.8.

SWMODE[3:0] Pins	Switch Mode
0x0	Single Partition
0x1	Single Partition with Serial EEPROM
0x2	Single Partition with Serial EEPROM Jump 0 Initialization
0x3	Single Partition with Serial EEPROM Jump 1 Initialization
0x8	Single partition with reduced latency
0x9	Single partition with Serial EEPROM initialization and reduced latency
0xA	Multi-partition with Unattached ports
0xB	Multi-partition with Unattached ports and I^2C Reset
0xC	Multi-partition with Unattached ports and Serial EEPROM initialization

Table 3.8 Normal Switch Modes (Part 1 of 2)

1. Refer to section section Switch Fundamental Reset on page 3-2 for details on the quasi-reset state.

Notes

SWMODE[3:0] Pins	Switch Mode
0xD	Multi-partition with Unattached ports with I^2C Reset and Serial EEPROM initialization
0xE	Multi-partition with Disabled ports
0xF	Multi-partition with Disabled ports and Serial EEPROM initialization

Table 3.8 Normal Switch Modes (Part 2 of 2)

The PES32NT8xG2 has one functional operating mode. Normal switch modes (i.e., switch modes that do not represent test modes) utilize this same single functional operating mode with different register initial values. These different initial values lead to the different behaviors.

- The behavior of any normal switch mode may be modified through serial EEPROM or SMBus initialization during the switch fundamental reset sequence.
- Since normal switch modes simply represent different initial register values, it is possible to modify the behavior of any normal switch mode to match the behavior of another mode through serial EEPROM or SMBus initialization during the switch fundamental reset sequence.

  The only exception to this rule are the reduced latency modes (i.e., "Single partition with reduced latency" and "Single partition with Serial EEPROM initialization and reduced latency"). These modes represent a single partition switch configuration in which partition state and port modes (see Chapter 5, Switch Partition and Port Configuration) can't be modified, except for port device number in downstream ports[1]. In these modes, the best-case latency across the switch is reduced by 12 ns.

  The reduced latency modes are suitable for users who do not wish to reconfigure ports and partitions in the switch and want a single partition configuration with minimized latency across the switch.

Table 3.9 lists the initial value of register fields that are dependent on switch modes. The effect of these initial values are described in section Single Partition Mode on page 3-10 through section Multi-Partition with Disabled Ports on page 3-10.

Some of the switch modes have an option with and without serial EEPROM initialization. Except for serial EEPROM initialization, these switch modes are identical. Therefore, Table 3.9 lists only the version without serial EEPROM initialization.

Register and Field	Switch Mode (SWMODE)		
	Single Partition (0x0 to 0x3, 0x8 and 0x9)	Multi-Partition with Unattached Ports (0xA, 0xB, 0xC or 0xD)	Multi-Partition with Disabled Ports (0xE or 0xF)
SWPART[0]CTL.STATE	0x1	0x0	0x0
SWPART[7:1]CTL.STATE	0x0	0x0	0x0
SWPORT[0]CTL.MODE	0x2	0x5	0x0
SWPORT[23:1]CTL.MODE	0x1	0x5	0x0

Table 3.9 Switch Mode Dependent Register Initialization

1. Modification of partition state and port mode in these modes produces undefined results.

Notes

Single Partition Mode

In single partition mode, the initial values outlined in Table 3.9 result in the following configuration.

- All ports are members of partition zero.
- Port 0 is configured as the upstream switch port of partition zero. All other ports are configured as downstream switch ports of partition zero.
- The initial state of partition zero is active. The initial state of all other partitions is disabled.

Multi-Partition with Unattached Ports

In this mode, the initial values outlined in Table 3.9 result in the following configuration.

- All ports are configured to operate in unattached mode.
- The initial state of all partitions is disabled.

Multi-Partition with Disabled Ports

In this mode, the initial values outlined in Table 3.9 result in the following configuration.

- All ports are configured to operate in disabled mode.

The initial state of all partitions is disabled.

Partition Resets

A partition reset is a reset that is associated with a specific switch partition. The reset has an effect only on those functions and ports associated with that switch partition. It has no effect on the operation of other switch partitions, ports in other switch partitions, or logic not associated with a switch partition (e.g., master SMBus, slave SMBus).

A partition reset may be subdivided into four subcategories: partition fundamental reset, partition hot reset, partition upstream secondary bus reset, and partition downstream secondary bus reset. These subcategories correspond to resets defined by the PCI architecture.

- A partition fundamental reset logically causes all logic associated with a partition to take on its initial state, but does not cause the state of register fields denoted as SWSticky to be modified.
- A partition hot reset logically causes all logic in the partition to be returned to an initial state, but does not cause the state of register fields denoted as Sticky or SWSticky to be modified.
- A partition upstream secondary bus reset is only applicable to partitions with an upstream switch port [1]. This type of reset logically causes all devices on the virtual PCI bus of a partition to be hot reset except the upstream port.
- A partition downstream secondary bus reset is only applicable to partitions with one or more downstream switch ports. This type of reset causes a hot reset to be propagated on the external link of the corresponding downstream switch port.

The operation of the slave SMBus interface is unaffected by a partition reset. Using the slave SMBus to access a register that is in the process of being reset causes the register's default value to be returned on a read and written data to be ignored on writes.

Partition Fundamental Reset

A partition fundamental reset is initiated by any of the following events.

- A switch fundamental reset (refer to section Switch Fundamental Reset on page 3-2).
- Assertion of a partition fundamental reset signal.
- As directed by the Switch Partition State (STATE) field in the Switch Partition (SWPARTxCTL) register.

1. Refer to section Switch Partitions on page 5-1 for a description of the port operating modes that are considered upstream switch ports, etc.

Notes

Associated with each partition is a partition fundamental reset input (PARTxPERSTN).

- The partition fundamental reset input for the first four partitions (i.e., partitions zero through three) are available as GPIO alternate functions.
- The partition fundamental reset input for all partitions are available on external I/O expanders (refer to section I/O Expanders on page 12-11).

When a partition fundamental reset is initiated, the following sequence of actions take place.

1. All logic associated with the switch partition (i.e., ports, switch core buffers, etc.) is logically reset to its initial state.
2. All port links associated with the partition enter the 'Detect' state.
3. All registers and fields, except those designated as SWSticky, take on their initial value. The value of SWSticky registers and fields is preserved across a partition fundamental reset.
4. As long as the condition that initiated the partition fundamental reset persists (e.g., the fundamental reset signal is asserted or the STATE field remains set to reset), logic associated with the partition remains at this step.
5. Ports associated with the partition begin to link train and normal partition operation begins.

Partition Hot Reset

A partition hot reset is initiated by any of the following events.

- Reception of TS1 ordered-sets on the partition's upstream port indicating a hot reset.
- Data link layer of the partition's upstream port indicates a DL_Down status. The only exception case is a DL_Down caused by the upstream port's link transitioning to L2/L3 Ready state (refer to section Link States on page 7-9).

When a partition hot reset is initiated the following sequence of actions take place.

1. The upstream port associated with the partition transitions its PHY LTSSM state to the appropriate state (i.e., the Hot Reset state on reception of TS1 ordered-sets indicating hot reset or else the Detect state).
2. Each downstream switch port associated with the partition (if any) whose link is 'up' propagates a hot reset by transmitting TS1 ordered sets with the hot reset bit set.

   If the link associated with a downstream switch port is in the Disabled LTSSM state, then a hot reset will not be propagated out on that port. The port will instead transition to the Detect LTSSM state. Although this is not technically a hot reset, this has the same functional effect on downstream components.
3. All logic associated with the switch partition (i.e., ports, switch core buffers, etc.) is logically reset to its initial state.
4. All register fields and registers associated with the switch partition except those designated Sticky and SWSticky, are reset to their initial value. The value of Sticky and SWSticky registers and fields is preserved across a hot reset.

   If the upstream port is a multi-function port, all functions of the port are affected by the hot reset.
5. As long as the condition that initiated the partition hot reset persists, logic associated with the partition remains at this step.
6. The port(s) associated with the partition begin to link train and normal partition operation begins.

The initiation of a hot reset due to the data link layer of the upstream port reporting a DL_Down status may be disabled by setting the Disable Link Down Hot Reset (DLDHRST) bit in the corresponding Switch Partition Control (SWPARTxCTL) register. When the DLDHRST is set and the upstream port's data link is down, the PHY LTSSM transitions to the appropriate states but the hot reset steps described above are not executed. As a result, the behavior of the partition is the following:

- The upstream port's function(s) are not reset and continue operation.
- TLPs destined to the partition's upstream port link are handled as follows.

Notes

- TLPs received by the secondary side of the PCI-to-PCI bridge function, which are destined to the upstream port's link, are treated as unsupported requests by the function.
- TLPs received by an NT function in another partition, which are destined to the upstream link associated with the NT function in this partition, are treated as unsupported requests by the NT function that first received the TLP.
- The DMA function continues normal operation, but silently discards TLPs destined to the upstream link.

– TLPs generated by the functions in the partition, and that are normally routed to the root (e.g., MSIs, INTx messages, PM_PME messages, etc.) are silently discarded.

– All transfers not destined to the partition's upstream port link (e.g., peer-to-peer TLPs between downstream switch ports, peer-to-peer TLPs between upstream port functions) continue to operate normally.

Note that other hot reset trigger conditions (i.e., hot reset triggered by reception of training sets with the hot reset bit set on the upstream port) are unaffected by the DLDHRST bit.

Partition Upstream Secondary Bus Reset

A partition upstream secondary bus reset is initiated by any of the following events.

– A one is written to the Secondary Bus Reset (SRESET) bit in the Bridge Control (BCTL) register of the PCI-to-PCI bridge function in the partition's upstream switch port [1].

When an upstream secondary bus reset occurs, the following sequence of actions take place on logic associated with the affected partition.

1. Each downstream switch port whose link is up propagates the reset by transmitting TS1 ordered sets with the hot reset bit set.

   If the link associated with a downstream switch port is in the Disabled LTSSM state, then a hot reset will not be propagated out on that port. The port will instead transition to the Detect LTSSM state. Although not a hot reset, this has the same functional effect on downstream components.

2. All registers fields in all registers associated with downstream switch ports, except those designated Sticky and SWSticky, are reset to their initial value. The value of fields designated Sticky or SWSticky is unaffected by an Upstream Secondary Bus Reset.
3. All TLPs received from downstream switch ports and queued in the switch are discarded.
4. Logic in the stack and switch core associated with the downstream switch ports are gracefully reset.
5. Wait for software to clear the SRESET bit in the BCTL register of the upstream switch port's PCI-to-PCI bridge function.
6. Normal downstream switch port operation begins.

The operation of the upstream switch port is unaffected by a secondary bus reset. The link remains up and Type 0 configuration read and write transactions that target the upstream port function(s) complete normally.

– The DMA and NT functions (if present in the upstream port of the partition) are unaffected and continue to operate normally.

During an Upstream Secondary Bus Reset, all TLPs destined to the secondary side of the upstream switch port's PCI-to-PCI bridge are treated as unsupported requests.

Partition Downstream Secondary Bus Reset

A partition downstream secondary bus reset may be initiated by the following condition:

– A one is written to the Secondary Bus Reset (SRESET) bit in the Bridge Control (BCTL) register of the PCI-to-PCI bridge function in a downstream switch port.

[1] Refer to section Switch Partitions on page 5-1 for a description of the port operating modes that are considered upstream ports, downstream switch ports, upstream switch ports, etc.

Notes

When a downstream secondary bus reset occurs, the following sequence of actions take place on logic associated with the affected partition.

- If the corresponding downstream switch port's link is up, TS1 ordered sets with the hot reset bit set are transmitted.
  - If the link associated with a downstream switch port is in the Disabled LTSSM state, then a hot reset will not be propagated out on that port. The port will instead transition to the Detect LTSSM state. Although not a hot reset, this has the same functional effect on downstream components.
- All TLPs received from corresponding downstream switch port and queued are discarded.
- Wait for software to clear the Secondary Bus Reset (SRESET) bit in the downstream switch port's Bridge Control Register (BCTL).
- Normal downstream switch port operation begins.

The operation of the upstream switch port is unaffected by a partition downstream secondary bus reset.

The operation of other downstream switch ports in this and other partitions is unaffected by a partition downstream secondary bus reset. During a partition downstream secondary bus reset, type 0 configuration read and write transactions that target the downstream switch port complete normally. During a partition downstream secondary bus reset, all TLPs destined to the secondary side of the downstream switch port's PCI-to-PCI bridge are treated as unsupported requests.

Port Mode Change Reset

A port mode change reset occurs when a port operating mode change is initiated and the OMA field in the corresponding SWPORTxCTL register specifies a reset. Port mode change reset behavior is described in section Reset Mode Change Behavior on page 5-21.

Notes

Switch Core

Notes

Overview

This chapter provides a detailed description of the PES32NT8xG2's switch core. As shown in section Architectural Overview on page 1-2, the switch core interconnects four stacks and two DMA modules. The four stacks are numbered 0 to 3. Each stack may be configured with a maximum of two x4 ports. Thus, the switch-core interconnects up to 8 ports in the device plus the two DMA modules.

The switch core's main function is to transfer TLPs among these ports efficiently and reliably. In order to do so, the switch core provides buffering, ordering, arbitration, and error detection services.

Switch Core Architecture

Figure 4.1 shows a high level diagram of the switch core block. The switch core is based on a non-blocking crossbar design with combined input-output buffering, optimized for system interconnect (i.e., peer-to-peer) as well as fanout (i.e., root-to-endpoint) applications.

At a high level, the switch core is composed of ingress buffers, a crossbar fabric interconnect, and egress buffers. These blocks are complemented with ordering, arbitration, and error handling logic (not shown in the figure). As packets are received from the link they are stored in the corresponding ingress buffer. After undergoing ordering and arbitration, they are transferred to the corresponding egress buffer via the crossbar interconnect.

The presence of egress buffers provides head-of-line-blocking (HOLB) relief when an egress port is congested. For example, a packet received on port 0 that is destined to port 1 may be transferred from port 0's ingress buffer to port 1's egress buffer even if port 1 does not have sufficient egress link credits. This transfer allows subsequent packets received on port 0 to be transmitted to their destination (e.g., to other ports).

In the PES32NT8xG2, all ports support a single virtual channel (i.e., VC0). Each port has dedicated ingress and egress buffers associated with VC0. These are referred to as the port's Ingress Frame Buffer (IFB) and Egress Frame Buffer (EFB) respectively. The IFB stores data received by the port from the link. The EFB stores data that will be transmitted or processed by the port[1].

The port IFBs and EFBs are implemented using shared memory modules. A memory module is capable of sustaining full bandwidth throughput on a x4 Gen2 link and may be shared by four x1 ports, two x2 ports, or one x4 port. Two memory modules are used for a x8 Gen2 port. Figure 4.1 shows the IFBs and EFBs for each port. The boundary between memory modules is shown using dashed lines. Note that each memory module has a dedicated connection to the switch core's crossbar.

1. In the switch, configuration request TLPs that target the port function(s) are processed on the egress side of the port.

Notes

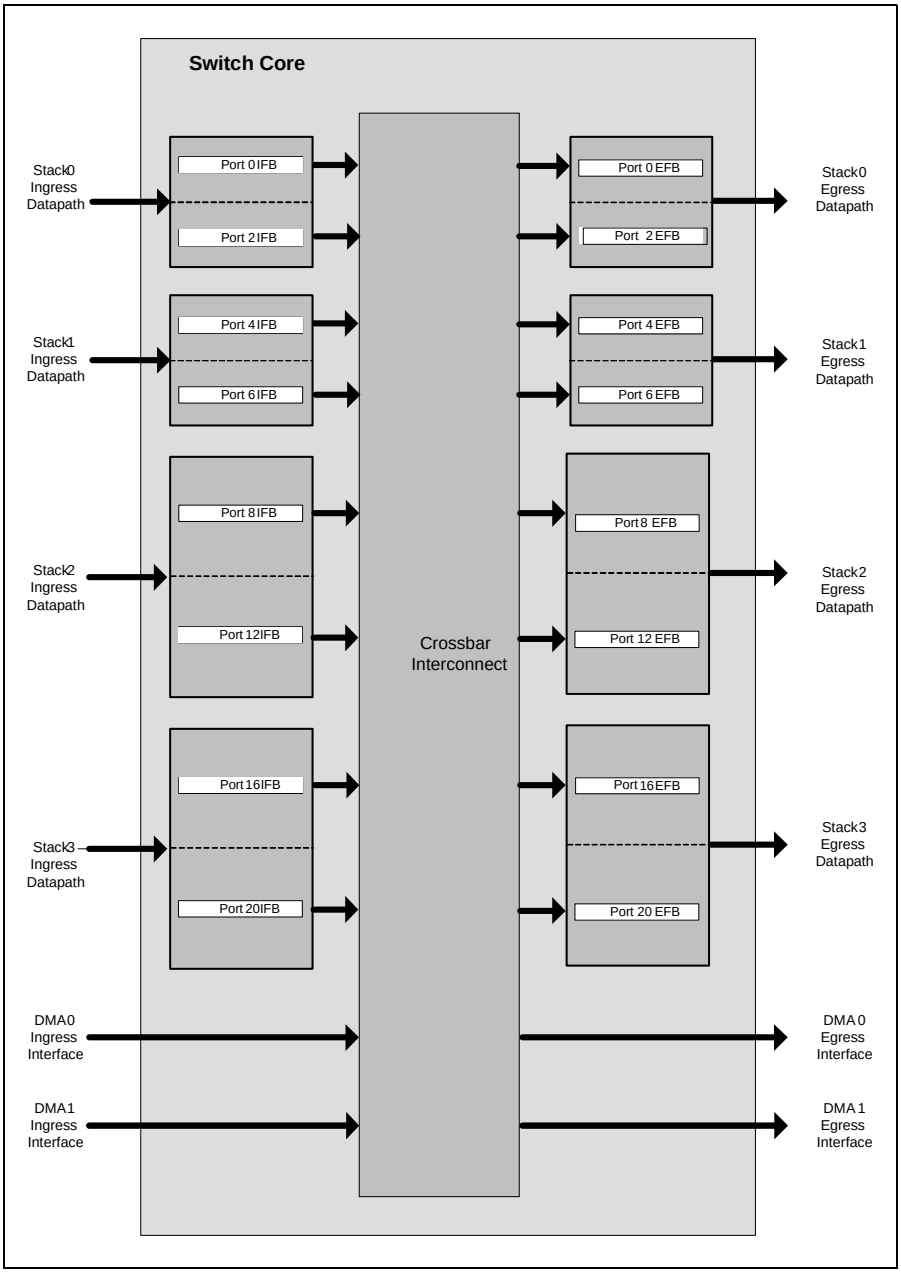

Figure 4.1 High Level Diagram of Switch Core

The crossbar interconnect is a matrix of pathways, capable of concurrently transferring data among all memory modules. The crossbar pathways are sized to sustain the throughput associated with a x8 Gen2 port.

In addition to the port IFBs and EFBs, the switch core crossbar provides interconnection interfaces for two DMA modules. DMA module 0 is logically associated with function 2 of port 0. DMA module 1 is logically associated with function 2 of port 8. The DMA ingress interface carries TLPs emitted by a DMA module, while the DMA egress interface carries TLPs destined to a DMA module.

Notes

Ingress Buffer

The switch core implements a per-port ingress buffer called the Ingress Frame Buffer (IFB). When a packet is received from the link, the ingress port determines the packet's route and subjects it to TC/VC mapping. If a valid mapping to VC0 is found, the packet is then stored in the port's IFB, together with its routing and handling information (i.e., the packet's descriptor).

The IFB consists of three queues. These queues are the posted transaction queue (PT queue), the non-posted transaction queue (NP queue), and the completion transaction queue (CP queue). The queues for the IFB are implemented using a descriptor memory and a data memory.

The size of a port's IFB depends on the port's maximum link width as determined by the configuration of the stack associated with the port. The IFB sizes are shown in Table 4.1. When two or more ports are merged as determined by the stack's configuration, the IFB descriptor and data memories for these ports are merged.

Note that ports with maximum link width of x4 or x8 support an MPS of up to 2 KB.

Port Width	IFB Queue	Total Size and Limitations (per-port)	Advertised Data Credits	Advertised Header Credits
x8	Posted	8192 Bytes and up to 127 TLPs	512	127
	Non Posted	2048 Bytes and up to 127 TLPs	128	127
	Completion	8192 Bytes and up to 127 TLPs	512	127
x4	Posted	4096 Bytes and up to 64 TLPs	256	64
	Non Posted	1024 Bytes and up to 64 TLPs	64	64
	Completion	4096 Bytes and up to 64 TLPs	256	64

Table 4.1 IFB Buffer Sizes

Egress Buffer

The switch core implements a per-port egress buffer called the Egress Frame Buffer (EFB). The EFBs provide head-of-line-blocking (HOLB) relief to the IFBs by allowing packets to be stored in an egress port's EFB even if the egress port's link does not have sufficient credits to transmit the packet. HOLB relief prevents subsequent packets in the IFB from being blocked by a head-of-line packet destined to a congested egress port, potentially allowing traffic to non-congested ports to proceed. Note that the HOLB relief is temporary and only lasts until the congested egress port's EFB is full. Under normal circumstances, it is not expected that this scenario will occur in a system.

Each EFB consists of three queues. These are the posted queue, non-posted queue, and completion queue. The use of these queues allows for packet re-ordering to improve transmission efficiency on the egress link. Refer to section Packet Ordering on page 4-6 for details. The queues for both EFBs are implemented using a descriptor memory and a data memory.

The size of a port's EFB depends on the port's link width as determined by the configuration of the stack associated with the port. The EFB sizes are shown in Table 4.2. When two or more ports are merged as determined by the stack's configuration, the EFB descriptor and data memories for these ports are merged.

Notes

Stack Mode	EFB Queue	Total Size and Limitations (per-port)
x8 Merged	Posted	8192 Bytes and up to 128 TLPs
	Non Posted	2048 Bytes and up to 128 TLPs
	Completion	8192 Bytes and up to 128 TLPs
x4 Bifurcated	Posted	4096 Bytes and up to 64 TLPs
	Non Posted	1024 Bytes and up to 64 TLPs
	Completion	4096 Bytes and up to 64 TLPs
x2 Bifurcated	Posted	2048 Bytes and up to 32 TLPs
	Non Posted	512 Bytes and up to 32 TLPs
	Completion	2048 Bytes and up to 32 TLPs
x1 Bifurcated	Posted	1024 Bytes and up to 16 TLPs
	Non Posted	256 Bytes and up to 16 TLPs
	Completion	1024 Bytes and up to 16 TLPs

Table 4.2 EFB Buffer Sizes

In addition to providing HOLB relief, the EFB is used as a dynamically-sized replay buffer. This allows for efficient use of the egress buffer space: when transmitted packets are not being acknowledged by the link partner the replay buffer grows to allow further transmission; when transmitted packets are successfully acknowledged by the link partner the replay buffer shrinks and this space is used as egress buffer space to provide more HOLB relief to the IFBs. Assuming a link partner issues ACK DLLPs at the rates recommended in the PCI Express Specification 2.1, the replay buffer naturally grows to the optimal size for the port's link width and speed. Table 4.3 shows the maximum number of TLPs that may be stored in the EFB's replay buffer.

Stack Mode	Replay Buffer Storage Limit
x8 Merged	128 TLPs
x4 Bifurcated	64 TLPs
x2 Bifurcated	32 TLPs
x1 Bifurcated	16 TLPs

Table 4.3 Replay Buffer Storage Limit

Crossbar Interconnect

The crossbar is a matrix of pathways, capable of concurrently transferring data between all the memory modules associated with the port IFBs and EFBs, as well as the two DMA modules. As mentioned before, the port IFBs and EFBs are implemented using shared memory modules. A memory module is capable of

Notes

sustaining full bandwidth throughput on a x4 Gen2 link and may be shared by four x1 ports, two x2 ports, or one x4 port. Two memory modules are used for an x8 Gen2 port. The PES32NT8xG2 switch core contains eight ingress memory modules and eight egress memory modules as shown in Figure 4.1.

The crossbar has ten ingress data interfaces (i.e., eight for ingress memory modules plus two for DMA modules) and ten egress data interfaces (i.e., eight for egress memory modules plus two for DMA modules).

The crossbar ingress and egress data pathways are sized at 160 bits. Given that a x8 Gen2 port has a throughput of 128 bits per cycle, the crossbar has 20% "overspeed". This overspeed compensates for the contention experienced by ports whose IFBs or EFBs share a memory module.

Virtual Channel Support

All PES32NT8xG2 ports support one virtual channel (i.e., VC0). In all port operating modes, function 0 of the port contains a VC Capability Structure that provides architected port arbitration and TC/VC mapping for VC0. Depending on the port operating mode, function 0 of the port may be a PCI-to-PCI bridge or NT function.

TLPs received by a port from the link are subjected to TC/VC mapping prior to being stored in the port's IFB. TLPs whose traffic class does not map to VC0 are treated as malformed TLPs by the port and logged as such in all functions of the port. Such TLPs are nullified prior to entering the IFB.

TLPs stored in the port's EFB are also subjected to TC/VC mapping prior to being transmitted on the link. TLPs whose traffic class does not map to VC0 are treated as malformed TLPs by the port and logged as such in all functions of the port. Such TLPs are nullified prior to being transmitted on the link.

Packet Routing Classes

As mentioned above, the switch core is responsible for transferring packets among ports. As packets are received from the PCI Express link, the ingress stack's application layer determines the packet route and sends this information to the switch core in the form of a packet descriptor.

From a switch core perspective, packet transfers among ports may be categorized as:

- Route-to-Self transfers (transfers from a port to itself)
- Port-to-port transfers (transfers among different ports)

Route-to-self transfers are implemented to process configuration requests that target the port, as well as for proprietary internal control messaging between the ingress and egress logic of the port. Port-to-port transfers are used for traffic routing of PCI Express TLPs, as well as for proprietary internal control messaging among ports. The DMA modules are treated as any other port from a switch core's perspective.

The PES32NT8xG2 is a partitionable PCI Express switch, meaning that ports may be partitioned into groups that logically operate as completely independent switches. In addition, the switch supports non-transparent bridging which allows the transfer of packets among partitions. Thus, port-to-port transfers may be further categorized as:

- Transfers among ports in the same partition (intra-partition transfers)
- Transfers among ports in different partitions (inter-partition transfers)

Intra-partition transfers occur among switch ports that are logically in the same partition. These include packet transfers from an upstream switch port to a downstream switch port, from a downstream switch port to an upstream switch port, and among downstream switch ports.

Inter-partition transfers are logically done across the non-transparent-bridge formed by two NT endpoints, one in each partition. Thus, an inter-partition transfer is logically received by the NT endpoint in the packet's source partition and transmitted by the NT endpoint in the packet's destination partition.

Notes

Packet Ordering

The PCI Express Base Specification 2.1 contains packet ordering rules to ensure the producer/consumer model is honored across a PCI Express hierarchy and to prevent deadlocks.

- The switch honors the strict and relaxed ordering rules defined in the PCI Express Base Specification.
- The switch does not support the ID-Based Ordering (IDO) rules defined in the PCI Express Base Specification.

The switch core performs packet ordering on a per-port basis, at the output of the ingress and egress buffers of each port. Table 4.4 shows the ordering rules honored by the switch core.

Row Pass Column?		Posted Request	Non-Posted Request		Completion	
		Memory Write or Message Request	Read Request	IO or Configuration Write Request	Read Completion	IO or Configuration Write Completion
Posted Request	Memory Write or Message Request	No	Yes	Yes	Yes	Yes
Non Posted Request	Read Request	No	No	No	Yes	Yes
	IO or Configuration Write Request	No	No	No	Yes	Yes
Completion Request	Read Completion	'Yes' if packet has RO bit set; Else 'No'	Yes	Yes	No	No
	IO or Configuration Write Completion		Yes	Yes	No	No

Table 4.4 Packet Ordering Rules in the PES32NT8xG2

Arbitration

Packets stored in the ingress buffer of each port are subject to arbitration as they are moved towards the target egress port. The switch core performs all packet arbitration functions in the PES32NT8xG2. The following sub-sections describe these in detail.

Port Arbitration

Figure 4.2 shows the architectural model of port arbitration.

Notes

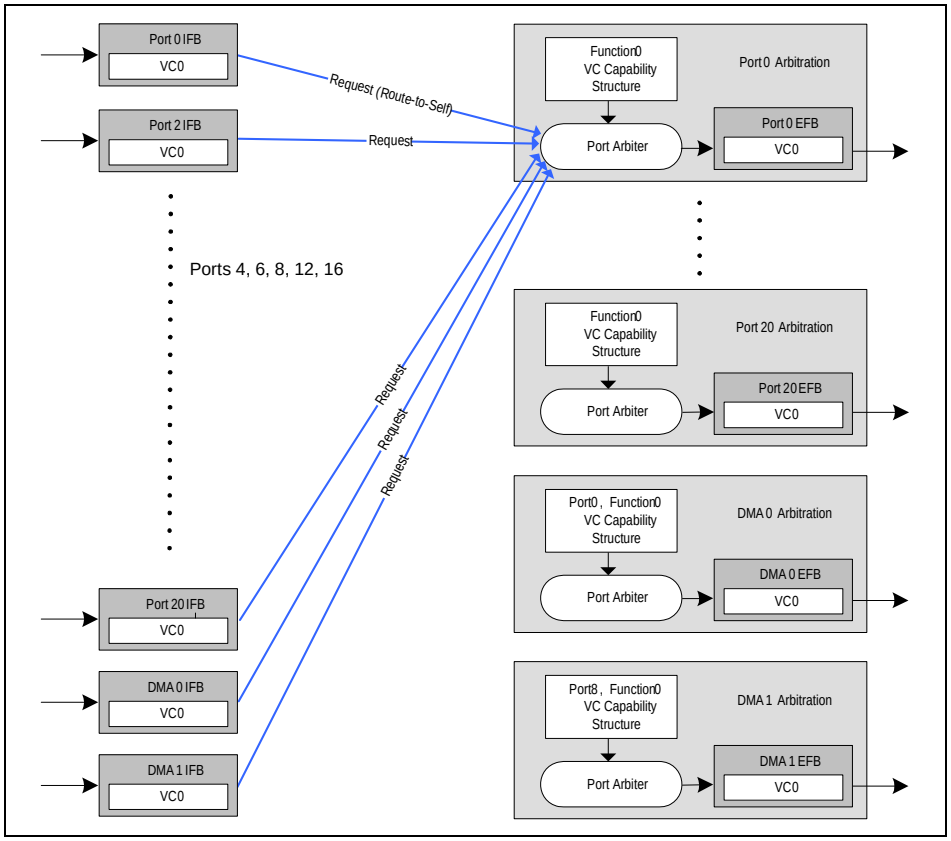

Figure 4.2 Architectural Model of Arbitration

Port arbitration resolves contention when multiple ingress ports target the same egress port. As shown in Figure 4.2, port arbitration is done independently by each port on the egress side (i.e., port arbitration regulates TLP entry into the corresponding EFB). Ports are numbered 0, 2, 4, 6, 8, 12, 16, and 20, plus two DMA modules named DMA module 0 and DMA module 1.

- DMA module 0 is logically associated with function 2 of port 0.
- DMA module 1 is logically associated with function 2 of port 8.

Each port has a dedicated port arbiter. The DMA modules also have a dedicated port arbiter that controls access into the DMA's EFB[1]. Ingress ports, or the DMA modules, that wish to transfer one or more packets to the egress port participate in port arbitration. Prior to participating in port arbitration, each ingress port does packet ordering. Based on this, each ingress port selects zero, one, or multiple packets as candidates for transfer towards the egress buffers (EFBs).

Each port arbiter performs arbitration according to the configuration of the VC Capability Structure in function 0 of the corresponding port. Depending on the operating mode of the port (e.g., upstream switch port, NT function port, upstream switch port with DMA function, etc.), function 0 of the port may be a PCI-to-PCI bridge function or an NT function.

- If function 0 of the port is a PCI-to-PCI bridge function, then the VC Capability Structure associated with the NT function is ignored and must not be configured by software (i.e., it must not be linked into the NT function's capabilities list).
- If function 0 of the port is an NT function, then the VC Capability structure associated with the PCI-to-PCI bridge function is ignored and must not be configured by software (i.e., it must not be linked into the PCI-to-PCI bridge function's capabilities list via the global address space).

1. The DMA EFB contains packets to be processed by the DMA engine. For more information, contact IDT at ssdhelp@idt.com.

Notes

Switch ports in this device support port arbitration using hardware fixed round-robin. As such, the port's VC Capability Structure indicates support for a hardware-fixed algorithm only (i.e., round-robin).

Hardware Fixed Round-Robin Arbitration

By default, all ports are programmed for hardware fixed round-robin port arbitration. A port operates in this mode unless it is configured for WRR arbitration as discussed in section Proprietary Weighted Round Robin (WRR) Arbitration below. When a port is programmed for hardware fixed round-robin, the port implements a round-robin scheme among all requesting ingress ports, including the DMA module(s) requesting transfers to the port.

- Other ports in the same partition can transfer packets to the port (i.e., intra-partition transfers).
- Other ports in other partitions can also transfer packets to the port (i.e., inter-partition transfers).

Proprietary Weighted Round Robin (WRR) Arbitration

All ports in the switch support a proprietary Weighted Round Robin (WRR) port arbitration scheme. This scheme is enabled on each port independently, by setting the Enable WRR Port Arbitration (EWRRPA) register in the port's PORTCTL register. WRR may be enabled on a port to enforce a differentiated priority policy among ingress ports that send traffic to the port.

When WRR is enabled on a port, the port's arbiter follows the weights programmed in the VC0 Port Arbiter Counter Initialization (VC0PARBCIx) registers located in the port's configuration space. These registers contain 10 port-arbitration count fields. Each field is associated with a port, plus two fields associated with the two DMA modules (e.g., 8 ports + 2 DMA modules = 10).

For example, the port arbiter for Port 0 follows the weights programmed in the VC0PARBCIx registers located in Port 0's configuration space (in function 0 of the port). The P2IC field in these VC0PARBCIx registers contains the count value for packet transfers requested by port 2 towards port 0. Similarly, the P4IC field contains the count value for packet transfers requested by port 4 towards port 0, and so on up to the number of ports in the device.

In addition, the P24IC field contains the count value for packet transfers requested by the DMA engine 0 (logically associated with function 2 of port 0) towards port 0. The P25IC field contains the count value for packet transfers requested by the DMA engine 1 (logically associated with function 2 of port 8) towards port 0.

The value programmed in a count field associated with a port, divided by the sum of the values programmed in all fields, represents the percentage of arbitration cycles allocated to that port. The fields are 8-bits wide each, so WRR may be programmed with a granularity of 0.015% increments.

Port arbitration can be said to occur in arbitration "epochs". At the start of each epoch, the port arbiter initializes the counters per the value in the VC0PARBCIx registers. Each time the port arbiter issues a grant to a requesting port, the counter associated with that port is decremented by one unit (unless its value is zero). Ports whose associated count value is zero are not granted by the arbiter until the current arbitration epoch ends and a new one begins. An arbitration epoch ends due to all counters being zero or due to no port with a non-zero count requesting service[1].

When a value in a count field is programmed to 0x0, the port associated with that field is never granted access by the port arbiter (i.e., the port is starved). A user must never program a value of 0x0 in a count field, unless it is known that the port associated with that count field will never issue requests to the port arbiter. This consideration includes the DMA engines (aliased to ports numbered 24 and 25) as well as transfers among ports in different partitions.

For example, if ports 0 and 8 are located in different switch partitions and transfers among these ports are possible (e.g., a TLP received by port 0 can cross partitions and be emitted by port 8, or vice-versa), then the P8IC field in port 0's VC0PARBCIx register must not be programmed to zero. Similarly, the P0IC field in port 8's VC0PARBCIx register must not be programmed to zero.

1. There is no overhead introduced by the end of an arbitration epoch (i.e., no clock cycles are added to the arbitration).

Notes

As another example, if the DMA engine located in function 2 of port 0 is active, the P24IC field of all ports out of which a DMA may issue traffic must not be set to 0x0. This includes ports in the same logical partition as the DMA, or ports in other partitions (i.e., when the DMA transmits packets across the NT bridge or NT multicast).

Finally, note that the percentage of arbitration cycles allocated for route-to-self transfers (i.e., see section Packet Routing Classes on page 4-5) may be controlled by modifying the appropriate field in the VC0PARBCIx registers. For example, arbitration cycle allocation for route-to-self transfers in port 0 is controlled via the Port 0 Initial Count (P0IC) field in this port's VC0PARBCI[0] register. Similarly, arbitration cycle allocation for route-to-self transfers in port 20 is controlled via the P20IC field in this port's VC0PARBCI[5] register.

By default, arbitration cycle allocation for route-to-self transfers is set to the maximum value to prevent starvation of this type of transfer. It is recommended that the value not be modified, and it must never be set to 0x0.

Cut-Through Routing

The PES32NT8xG2 utilizes a combined input and output buffered cut-through switching architecture to forward PCI Express TLPs between switch ports. Cut-through means that while a TLP is being received on an ingress link, it can be simultaneously routed across the switch and transferred on the egress link. The entire TLP need not be received and buffered prior to starting the routing process (i.e., store-and-forward). This reduces the latency experienced by packets as they are transferred across the switch.

Typically, cut-through occurs when a TLP is received on an ingress link whose bandwidth is greater than or equal to the bandwidth of the egress link. For example, a TLP received on a x4 Gen 2 port and destined to a x1 Gen 2 port is cut-through the switch. This rule ensures that the ingress link has enough bandwidth to prevent 'underflow' of the egress link.

In addition to this, the PES32NT8xG2 does "adaptive cut-through", meaning that packets are cut-through even if the egress link bandwidth is greater than the ingress link bandwidth. In this case, the cut-through transfer starts when the ingress port has received enough quantity of the packet such that the packet can be sent to the egress link without underflowing this link.

The ingress and egress link bandwidth is determined by the negotiated speed and width of the links.

Table 4.5 shows the conditions under which cut-through and adaptive-cut-through occur. When the conditions are met, cut-through is performed across the IFB, crossbar[1], and EFB. Note that a packet undergoing a cut-through transfer across the switch core may be temporarily delayed by the presence of prior packets in the IFB and/or EFB (i.e., head-of-line blocking). In this case, the packet starts cutting-through as soon as it becomes unblocked.

When cut-through routing of a packet is not possible, the packet is fully buffered in the appropriate IFB prior to being transferred to the EFB and towards the egress link (i.e., store-and-forward operation). Once the packet is stored in the IFB, there is no necessity to fully store it in the EFB as it is transferred towards the egress link (i.e., the packet can cut-through the EFB).

1. During cut-through transfers, the crossbar maintains the connection between the appropriate IFB and EFB through-out the duration of the transfer.

Notes

Ingress Link Speed (GT/s)	Ingress Link Width	Egress Link Speed (GT/s)	Egress Link Width	Conditions for Cut-Through
2.5	x8	2.5	x8, x4, x2, x1	Always
		5.0	x4, x2, x1	Always
			x8	At least 50% of packet is in IFB
	x4	2.5	x4, x2, x1	Always
			x8	At least 50% of packet is in IFB
		5.0	x2, x1	Always
			x4	At least 50% of packet is in IFB
			x8	At least 75% of packet is in IFB
	x2	2.5	x2, x1	Always
			x4	At least 50% of packet is in IFB
			x8	At least 75% of packet is in IFB
		5.0	x1	Always
			x2	At least 50% of packet is in IFB
			x4	At least 75% of packet is in IFB
			x8	At least 100% of packet is in IFB
	x1	2.5	x1	Always
			x2	At least 50% of packet is in IFB
			x4	At least 75% of packet is in IFB
			x8	Never (100% of packet is in IFB)
		5.0	x1	At least 50% of packet is in IFB
			x2	At least 75% of packet is in IFB
			x4	Never (100% of packet is in IFB)
			x8	

Table 4.5 Conditions for Cut-Through Transfers (Part 1 of 2)

Notes

Ingress Link Speed (GT/s)	Ingress Link Width	Egress Link Speed (GT/s)	Egress Link Width	Conditions for Cut-Through
5.0	x8	2.5	x8, x4, x2, x1	Always
		5.0	x8, x4, x2, x1	Always
	x4	2.5	x8, x4, x2, x1	Always
		5.0	x8	At least 50% of packet is in IFB
			x4, x2, x1	Always
	x2	2.5	x8	At least 50% of packet is in IFB
			x4, x2, x1	Always
		5.0	x8	At least 75% of packet is in IFB
			x4	At least 50% of packet is in IFB
			x2, x1	Always
	x1	2.5	x8	At least 75% of packet is in IFB
			x4	At least 50% of packet is in IFB
			x2, x1	Always
		5.0	x8	Never (100% of packet is in IFB)
			x4	At least 75% of packet is in IFB
			x2	At least 50% of packet is in IFB
			x1	Always

Table 4.5 Conditions for Cut-Through Transfers (Part 2 of 2)

Request Metering

Request metering may be used to reduce congestion in PCI Express switches caused by a static rate mismatch. Request metering is available on all PES32NT8xG2 switch ports but is disabled by default. The DMA function also has a mechanism to meter requests that it generates. This mechanism operates independently from the mechanism described in this section. Refer to section DMA Request Rate Control on page 15-22 for details.

A static rate mismatch is a mismatch in the capacity of the path from a component injecting traffic into the fabric (e.g., a Root Complex) and the ultimate destination (e.g., an Endpoint). An example of a static rate mismatch in a PCI Express fabric is a x8 root injecting traffic destined to a x1 endpoint. PCI Express fabrics are typically no more than one switch deep. Therefore, static rate mismatches typically occur within a switch due to asymmetric link rates.

Figure 4.3 illustrates the effect of congestion on PCI Express fabric caused by a static rate mismatch. In this example, there are two endpoints issuing memory read requests to a root. Endpoint A has a x1 link to the switch, while endpoint B and the root complex have a x8 link.

Memory read request TLPs are three or four DWords in size. A single memory read request may result in up to 4 KB of completion data being returned to the requester. Depending on system architecture and configured maximum payload size, this completion data may be returned as a single completion TLP or may be returned as a series of small (e.g., 64B data) TLPs.

Notes

Consider an example where endpoints A and B are injecting read request to the root at a high rate and the root is able to inject completion data into the fabric at a rate higher than which may be supported by endpoint A's egress link. The result is that the endpoint A's EFB and the root's IFB may become filled with queued completion data blocking completion data to endpoint B.

If read requests are injected sporadically or at a low rate, then buffering within the switch may be used to accommodate short lived contention and allow completions to endpoints to proceed without interfering. If read requests are injected at a high rate, then no amount of buffering in the switch will prevent completions from interfering.

PCI Express has no end-to-end QoS mechanisms. Therefore, it is common for endpoints to be designed to inject requests into a fabric at high rates. Request metering is a congestion avoidance mechanism that limits the request injection rate into a fabric. Although this example illustrates the effect of a static rate mismatch in an I/O connectivity application, similar situations may occur in system interconnect applications.

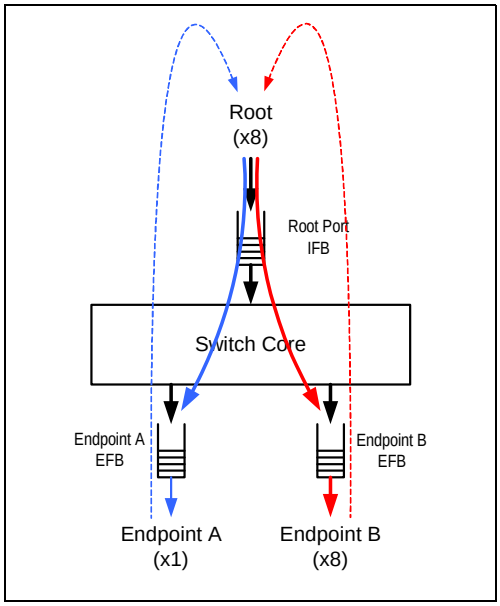

Figure 4.3 PCI Express Switch Static Rate Mismatch

Request metering operation is illustrated in Figure 4.4. Figure 4.4(a) shows requests injection without request metering. Figure 4.4(b) shows requests injection with request metering. Request metering is implemented by logic at the interface between the IFB and the switch core arbiter. When a request reaches the head of the non-posted IFB queue, request metering logic examines the request and estimates the amount of time that the associated completion TLPs will consume on the endpoint link (i.e., completion transfer time). The request is then allowed to proceed and a timer is initiated with the estimated completion transfer time. The next request from that IFB is not allowed to proceed until the timer has expired.

Notes

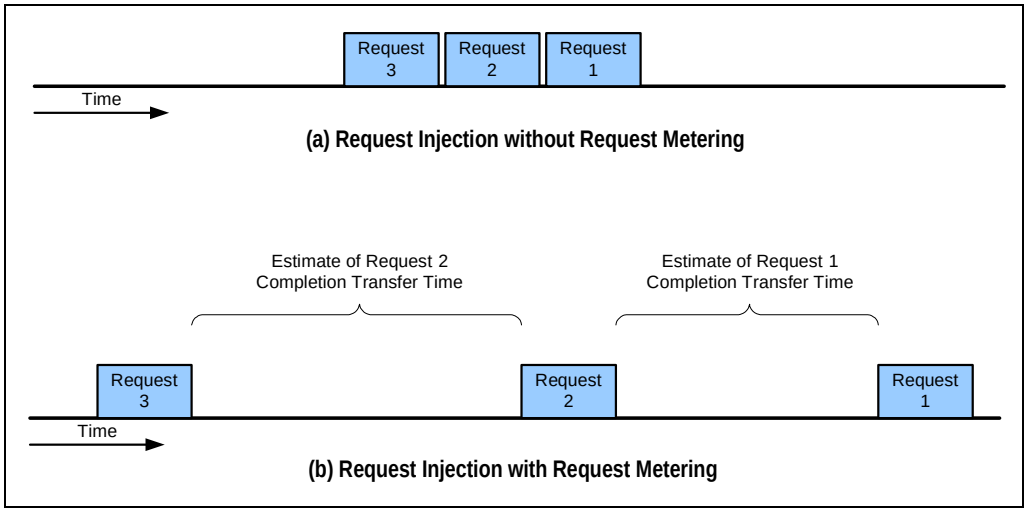

Figure 4.4 PCI Express Switch Static Rate Mismatch

The request metering implementation in the switch makes a number of simplifying assumptions that may or may not be true in all systems. Therefore, it should be expected that some amount of parameter tuning may be required to achieve optimum performance.

Note: Tuning of the request metering mechanism should take into account the completion timeout value of the associated requesters (i.e., request metering should be tuned such that a requester's completion timeout value is not violated).

Operation

The completion transfer timer is implemented using a counter. The counter is loaded with an estimate of the number of DWords that will be transferred on the link in servicing the completion and is decremented at a rate that corresponds to the number of DWords that will be transferred on the link in a 4ns period.

Request metering is enabled on an input port when the Enable (EN) bit is set in the port's Requester Metering Control (RMCTL) register.

An non-posted request TLP is allowed to be transferred into the switch core when the request metering counter is zero.

When a request is transferred into the switch core, the request metering counter is loaded with a value that estimates the number of DWords associated with the corresponding completion(s). The method for determining this value is described in section Completion Size Estimation on page 4-14.

The request metering counter is a 24-bit counter. The count represents a fixed-point 0:13:11 number (i.e., an unsigned number with 13 integer bits and 11 fractional bits) but is treated by the logic as a 24-bit unsigned integer. The value loaded into the request metering counter for the last non-posted request is available in the Count (COUNT) field of the Request Metering Counter (RMCOUNT) register.

- The requester metering initial counter value, computed as described in section Completion Size Estimation on page 4-14, is a fixed point 0:13:3 number.
- The request metering counter is a 24-bit counter that represents a fixed point 0:13:11 number (i.e., an unsigned number with 13 integer bits and 11 fractional bits).
- The least significant eight fractional bits of the initial counter value are always implicitly zero.

The request metering counter is decremented by a value that corresponds to the number of DWords transferred on the link per 4ns period. The value is equal to the sum of the decrement value plus the value of the Decrement Value Adjustment (DVADJ) field in the RMCTL register.

The decrement value is a fixed-point 0:4:3 number (i.e., an unsigned number with 4 integer bits and 3 fractional bits), determined by the port's negotiated link width and speed as shown in Table 4.6.The least significant eight fractional bits of the decrement value are always implicitly zero.

Notes

The Decrement Value Adjustment (DVADJ) field represents a 1:4:11 number (i.e., a sign-magnitude fixed-point number with 4 integer bits and 11 fractional bits). The signed nature of the DVADJ field provides fine grain programmable adjustment of the value by which the counter is decremented.

When the sum of the decrement value plus DVADJ results in a value less than or equal to zero, the hardware ignores DVADJ and uses the decrement value. The counter stops decrementing when it reaches zero or when a rollover occurs (i.e., the decrement causes it to become negative).

Link Width	Link Speed	Decrement Value	Notes
x1	Gen 1	0x02	Corresponds to 1 Byte per clock tick
x2	Gen 1	0x04	Corresponds to 2 Bytes per clock tick
x4	Gen 1	0x08	Corresponds to 4 Bytes per clock tick
x8	Gen 1	0x10	Corresponds to 8 Bytes per clock tick
x1	Gen 2	0x04	Corresponds to 2 Bytes per clock tick
x2	Gen 2	0x08	Corresponds to 4 Bytes per clock tick
x4	Gen 2	0x10	Corresponds to 8 Bytes per clock tick
x8	Gen 2	0x20	Corresponds to 16 Bytes per clock tick

Table 4.6 Request Metering Decrement Value

The computation that occurs on each clock tick by the request metering counter is shown in Figure 4.5.

```
tmp = RequestMeteringCounter
if ((DecrementValue[LinkSpeed,LinkWidth] + RMCTL.DVADJ) <= 0) {
        RequestMeteringCounter -= DecrementValue[LinkSpeed,LinkWidth]
} else {
        RequestMeteringCounter -= (DecrementValue[LinkSpeed,LinkWidth] + RMCTL.DVADJ)
}
// prevent negative count
if (tmp < RequestMeteringCounter) {
        RequestMeteringCounter = 0
}
```

Figure 4.5 Request Metering Counter Decrement Operation

Completion Size Estimation

This section describes the value that is loaded into the request metering counter when a request is transferred into the switch core. This value is referred to as the completion size estimate. The completion size estimate is based on the type of non-posted request as described below.

The request metering counter is a 24-bit counter that represents a fixed point 0:13:11 number (i.e., an unsigned number with 13 integer bits and 11 fractional bits). The completion size estimate is a 0:13:3 number. The least significant eight fractional bits of the completion size estimate are always implicitly zero.

Non-Posted Writes

The completion size estimate is 0x0018 which corresponds to 3 DWords (3 DWord header).

Notes

Non-Posted Reads

The completion size estimate is based on the Length field in the read request header and is computed as shown in Figure 4.6. All arithmetic in this section is performed using an implicit 0:13:3 representation and all values are implicitly converted to this value.

```
DataDWords = éLength / 4ù *4
If (DataDWords == 0) {
        CompletionSizeEstimate = 3
} else if (DataDWords <= CnstLimit) {
        CompletionSizeEstimate = DataDWords + 1
} else {
        OverheadDWords = (DataDWords >> OverheadFactor)
        CompletionSizeEstimate = DataDWords + OverheadDWords
}
```

Figure 4.6 Non-Posted Read Request Completion Size Estimate Computation

The number of data DWords in a non-posted request TLP is estimated by the number of PCI Express data credits required by the corresponding completion(s). Each PCI Express data credit is 4 DWords or 16 bytes. The first line in Figure 4.6 computes the number of DWords required by the completion(s) using the number of required PCI Express data credits. This corresponds to PCI Express completion data credits multiplied by 4.

If the number of data DWords is zero, then the completion size is estimated to be three DWords (i.e., a 0:13:3 representation value of 0x0018).

- Otherwise, if the number of required data DWords is less than the Constant Limit (CNSTLIMIT) field in the RMCTL register, then the completion size is estimated as the number of required data DWords plus one.
- Otherwise, if the number of required data DWords is greater than CNSTLIMIT, then the completion size is estimated using OverheadDWords as described below.

OverheadDWords represents the number of DWords of link overhead. This includes the header, data link layer overhead, and physical layer overhead of the completion TLP(s) associated with this request. Ideally, OverheadDWords would be set to the number of completion TLPs associated with the request multiplied by the TLP overhead. Unfortunately, this requires multiplication. Therefore, the following estimate may be used.

- A completion header is 3 DWords. There are 2 DWords of additional overhead associated with a TLP. Therefore, a reasonable estimate of the overhead is 5 DWords.

  In many systems, completions are 64-bytes in size (i.e., 16 DWords in size).

  OverheadDWords = (Length / 16) * 5.

  This is approximately equal to OverheadDWords = (Length / 16) * 4.

  This may be simplified to (Length / 4) and may be computed as (Length >> 2).

  Thus, an acceptable value for OverheadFactor in many systems is 2.

The OverheadFactor value used in computing the completion size estimate is contained in the Overhead Factor (OVRFACTOR) field in the RMCTL register.

Notes

Internal Errors

Internal errors are errors which are associated with a PCI Express interface, which occur within a component, and which may not be attributable to a packet or event on the PCI Express interface itself or on behalf of transactions initiated on PCI Express.

The PES32NT8xG2 classifies the following IDT proprietary switch errors as internal errors:

- Switch core time-outs
- Single and double bit internal memory ECC errors.
- End-to-end data path parity protection errors

In addition, the switch offers a mechanism by which AER errors detected on a port may be reported as internal errors in other ports. This mechanism is described in section Reporting of Port AER Errors as Internal Errors on page 4-19. Internal errors are reported by the port in which they are detected through AER as outlined in the PCI Express Base Specification. The reporting of internal errors in AER may be disabled by clearing the Internal Error Reporting Enable (IERROREN) bit in the port's Internal Error Reporting Control (IERRORCTL) register.

The setting of the IERROREN bit in the IERRORCTL register affects all functions present in the port (e.g., PCI-to-PCI bridge, NT function, and DMA function). When internal error reporting is disabled, the following AER fields become read-only in all functions of the port:

- Uncorrectable Internal Error Status (UIE) field in the AERUES register
- Uncorrectable Internal Error Mask (UIE) field in the AERUEM register
- Uncorrectable Internal Error Severity (UIE) field in the AERUESV register
- Correctable Internal Error Status (CIE) field in the AERCES register
- Correctable Internal Error Mask (CIE) field in the AERCEM register
- Header Log Overflow Mask (HLO) field in the AERCEM register

The switch does not support recording of headers for uncorrectable internal errors. When an uncorrectable internal error is reported by AER, a header of all ones is recorded. It is possible to control the reporting of internal errors detected by a port on a per-function basis. Each port function contains an Internal Error Mask register that allows selection of which internal errors are reported on the function's AER Capability Structure.

- In the PCI-to-PCI bridge function, the P2PIERRORMSK0/1 registers provide this control.
- In the NT function, the NTIERRORMSK0/1 registers provide this control.
- In the DMA function, the DMAIERRORMSK0/1 registers provide this control.

By default, the following internal errors are reported only by the DMA function's AER Capability Structure.

- DMA IFB timeout (for posted, non-posted, and completion TLPs)
- DMA IFB single and double bit errors (for control and data memories)
- DMA EFB single and double bit errors (for control and data memories)
- DMA end-to-end data-path parity error

In addition, internal errors caused by the mechanism described in section Reporting of Port AER Errors as Internal Errors on page 4-19 are only reported by the PCI-to-PCI bridge function. All other internal errors are reported by the AER Capability Structure in all functions present in the port (e.g., PCI-to-PCI bridge, NT, and DMA). The functions present in the port depend on the port's operating mode. Refer to Chapter 5, Switch Partition and Port Configuration.

Corresponding to each possible internal error source is a status bit in the Internal Error Reporting Status (IERRORSTS0/1) registers. A bit is set in the status register when the corresponding internal error is detected. The purpose of the IERRORSTS0/1 registers is to log the specific internal error(s) detected by the port. Software that is aware of the IERRORSTS0/1 registers can use this information to gain further insight regarding the internal error(s) detected by a port. Software that is not aware of the IERRORSTS0/1 registers can ignore this register.

Notes

Each internal error status bit has an associated severity bit in the Internal Error Severity (IERRORSEV0/1) registers. When an unmasked internal error is detected, the error is reported as dictated by the corresponding severity bit (i.e., either an uncorrectable internal error or a correctable internal error). When an uncorrectable or correctable internal error is reported, the corresponding AER status bit is set and processed as dictated by the PCI Express Base Specification.

If the internal error severity in the IERRORSEV0/1 register is set to uncorrectable, then the UIE bit is set in the AERUES register. Once this bit is set, the error is reported to the root-complex as specified in the PCI Express Base Specification. Note that while the UIE bit is set, the detection of a subsequent uncorrectable internal error is ignored by the AER mechanism and not reported to the root-complex. Still, the appropriate bit is logged in the IERRORSTS0/1 registers regardless of the setting of the UIE bit.

If the internal error severity in the IERRORSEV0/1 register is set to correctable, then the CIE bit is set in the AERCES register. Once this bit is set, the error is reported to the root-complex as specified in the PCI Express Base Specification. Note that while the CIE bit is set, the detection of a subsequent correctable internal error is ignored by the AER mechanism and not reported to the root-complex. Still, the appropriate bit is logged in the IERRORSTS0/1 registers regardless of the setting of the CIE bit.

Figure 4.7 shows a logical representation of the internal error circuitry within each PES32NT8xG2 port.

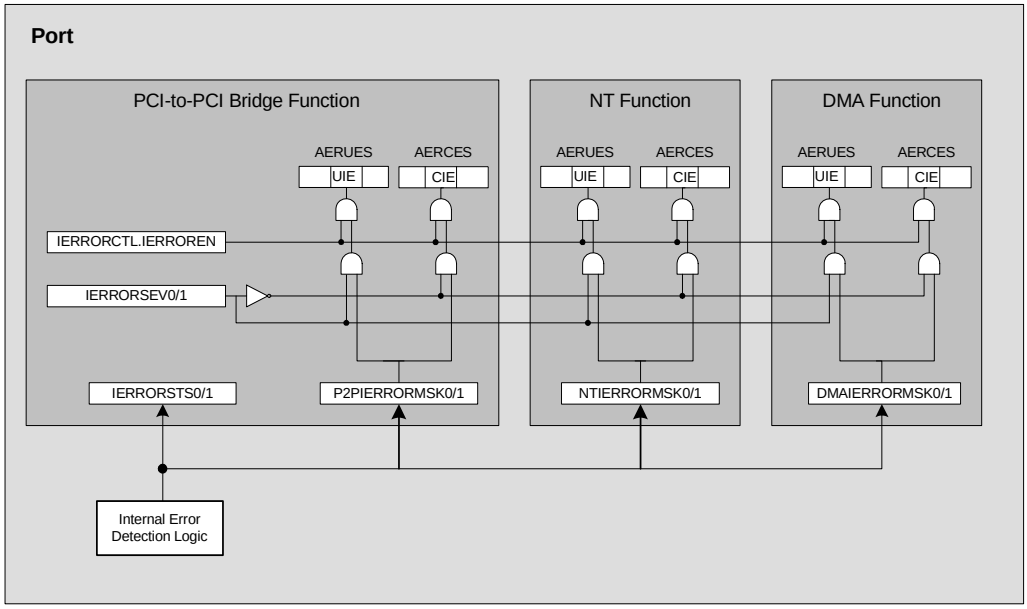

Figure 4.7 Internal Error Logic in Each PES32NT8xG2 Port

To facilitate testing of software error handlers, the occurrence of an internal error may be emulated by writing a value of one to the corresponding bit position in the Internal Error Test (IERRORTST0/1) registers. Once a bit is set in IERRORSTS0/1 registers, it is processed as though the actual error occurred (e.g., logged in the IERRORSTS0/1 register, reported by AER, etc.)

Error emulation via the IERRORTST0/1 registers is only applicable to the PCI-to-PCI bridge function. The logging of internal errors in the AER capability structure of the NT and DMA functions is not possible via the IERRORSTS0/1 registers.[1]

Switch Core Time-Outs

The switch core discards any TLP that reaches the head of an IFB or EFB queue and is more than 64 seconds old. This includes posted, non-posted, completion and inserted TLPs. Whenever a TLP is discarded by a port due to a switch time-out, a bit corresponding to the type of TLP that was discarded is

1. It is possible to test logging of internal errors in the NT functions by using the AER Emultion Registers in this function. Refer to section Error Emulation Control in the NT Function on page 14-39 for details.

Notes

set in the Internal Error Reporting Status 0 (IERRORSTS0) register. If during processing of a TLP with broadcast or multicast routing a switch core time-out occurs, then the switch core will abort processing of the TLP. This may result in the broadcast TLP being transmitted on some but not all destination ports. For ports that contain a DMA function, the DMA has separate switch time out controls.

Memory SECDED ECC Protection

PCI Express provides reliable hop-by-hop communication between interconnected devices, such as roots, switches, and endpoints, by utilizing a 32-bit Link CRC (LCRC), sequence numbers, and a link level retransmission protocol. While this mechanism provides reliable communication between interconnected devices, it does not protect against corruption that may occur inside of a device. PCI Express defines an optional end-to-end data integrity mechanism that consists of appending a 32-bit end-to-end CRC (ECRC) computed at the source over the invariant fields of a Transaction Layer Packet (TLP) that is checked at the ultimate destination of the TLP. While this mechanism provides end-to-end error detection, unfortunately it is an optional PCI Express feature and has not been implemented in many north-bridges and endpoints. In addition, the ECRC mechanism does not cover variant fields within a TLP.

Since deep sub-micron devices are known to be susceptible to single-event-upsets, a mechanism is desired that detects errors that occur within a PCI Express switch.

The PES32NT8xG2 protects all memories (i.e., both data and control structures) with a Single Error Correction with Double Error Detection (SECDED) Error Correcting Code (ECC). The objective of this memory protection is to prevent silent data corruption. Single bit errors are automatically corrected and optionally reported while double bit errors are optionally reported.

Double bit errors are uncorrectable memory errors that may compromise the integrity of control and data structures. Detection of a double bit error may result in further modification of one or more memory bits in the data quantity in which the error was detected (i.e., single bit error correction is not disabled when a double bit error is detected and a double bit error may result in one or more single bit corrections).

Associated with each port are five memories: IFB control, IFB data, EFB control, EFB data, and Replay Buffer Control. Each port contains memory error control and status registers that are used to manage memory errors associated with that port. In addition, ports that contain a DMA function have four other memories: DMA IFB control, DMA IFB data, DMA EFB control, and DMA EFB data. Such ports contain error control and status registers that are used to manage memory errors in its associated DMA memories.

When a single or double bit error is detected in a memory, the status bit corresponding to the memory in which the error was detected is set in the Internal Error Reporting Status 0 (IERRORSTS0) register.

A double bit error detected by a memory associated with TLP data (i.e., IFB or EFB data) results in the TLP being nullified when it reaches the DL layer of an egress port. The TLP is nullified by inverting the computed LCRC and ending the packet with an EDB symbol. Nullified TLPs received by a link partner are discarded. Although the TLP is nullified, flow control credits associated with the egress port may not be correctly updated. Thus, double bit errors could result in a flow control credit leak.

Note: The DL layer never replays a TLP with a sequence number different from that initially used. If a double bit error is detected during a DL layer replay, then all TLPs in the replay buffer are flushed.

If a double bit error is detected by an internal memory in a TLP that targets a function in the switch (e.g., a configuration read or write request to the PCI-to-PCI bridge function, or a TLP that targets the DMA function), then the TLP is discarded. This may inhibit the logging of other errors (e.g., unsupported request) caused by that same TLP.

End-to-End Data Path Parity Protection

In addition to memory ECC protection, the PES32NT8xG2 supports end-to-end data path parity protection. Data flowing into the switch is protected by the LCRC. Within the Data Link (DL) layer of the switch ingress port, the LCRC is checked and a 32-bit DWord even parity is computed on the received TLP data. If an LCRC error is detected at this point, the link level retransmission protocol is used to recover from the error by forcing a retransmission by the link partner.

Notes

As the TLP flows through the switch, its alignment or contents may be modified. In all such cases, parity is updated and not recomputed. Hence, any error that occurs is propagated and not masked by a parity regeneration. When the TLP reaches the DL layer of the switch egress port, parity is checked and in parallel an LCRC is computed. If the TLP is parity error free, then the LCRC and TLP contents are known to be correct and the LCRC is used to protect the packet through the lower portion of the DL layer, PHY layer, and link transmission.

If a parity error is detected by the DL layer of an egress port, then the TLP is nullified by inverting the computed LCRC and ending the packet with an EDB symbol. Nullified TLPs received by the link-partner are discarded. In addition to nullifying the TLP, the End-to-End Parity Error (E2EPE) bit is set in the Internal Error Status 0 (IERRORSTS0) register.

Note: The DL layer never replays a TLP with a sequence number different from that initially used. If a parity error is detected during a DL layer replay, then all TLPs in the replay buffer are flushed.

In addition to TLPs that flow through the switch, cases exist in which TLPs are produced and consumed by the switch (e.g., configuration requests that target a function in the switch, TLPs that target a DMA function, requests and completions generated by a switch function, etc.) Whenever a TLP is produced by the switch, parity is computed as the TLP is generated. Thus, error protection is provided on produced TLPs as they flow through the switch. In addition, parity is checked on all consumed TLPs. If an error is detected, the TLP is discarded and an error is reported by setting the E2EPE bit in the IERRORSTS0 register.

A parity error reported at a switch port cannot be definitively used to identify the location within the device at which the fault occurred as the fault may have occurred at another port, in the switch core, or may have occurred locally at the port.

Reporting of Port AER Errors as Internal Errors

In scenarios in which the PES32NT8xG2 switch is multi-partitioned, a need may exist to inform the root associated with each partition of anomalous conditions occurring in ports associated with other partitions. For example, a root acting as a switch manager may have a need to be notified of a surprise link down condition in a port associated with another switch partition. The switch manager could use this information to reconfigure the switch.

The event signaling mechanism described in Chapter 16, Switch Events, provides this capability by allowing events in a partition to be notified to root devices in other partitions via interrupts generated by each partition's upstream port. Still, the event signaling mechanism is limited to notifying partitions of a number of pre-defined events (e.g., port link down, port link up, failover, etc.), which do not include port AER errors.

In order to notify a partition of the occurrence of port AER errors in other partitions, the switch offers a mechanism by which AER errors that occur in a port (e.g., ACS violation, receiver overflow, etc.) may be reported as internal errors in the AER Capability Structure of any other port. In this case, the port(s) in which the error is logged as an AER internal error report the error to the system as defined by AER rules (i.e., an uncorrectable fatal, non-fatal, or correctable error message may be generated by the port).

As mentioned above, each port contains internal error detection logic that feeds into the port's Internal Error Status (IERRORSTS0/1) registers as well as the AER internal error status bits (see Figure 4.7). Apart from detecting internal errors in the port itself, the internal error detection logic of a port is capable of noticing when other ports have detected an AER error.

When the internal error detection logic in a port notices the occurrence of an AER error in another port, a bit is set in the IERRORSTS1 register of the former port. The IERRORSTS1 register has several bits (e.g., P0AER, P1AER, P2AER, etc.) Bit PxAER is set when port 'x' has notified the detection of an AER error as described next.

Notes

Each port is capable of notifying the detection of an AER error to other ports. Each port has an internal non-software visible register named Port AER Status (PAERSTS) which provides a gathering point for combined AER correctable and uncorrectable errors of all functions (e.g., PCI-to-PCI bridge, NT, and DMA) in the port.

- Bits in this internal register have a transient nature (i.e., they are set by hardware when the error is detected, and cleared by hardware as the error condition passes). As a result, this register is not visible by software.
- Each bit in the PAERSTS register corresponds to an AER error (e.g., Data Link Protocol error, Surprise Down error, etc.)
- The Internal Error (IE) bit in the PAERSTS register of a port is set when the port logs an internal error that occurred in the port itself (i.e., errors logged in the IERRORSTS0 register).
  - Note that the IE bit in the PAERSTS register is not set when a port logs an internal error originally detected by another port (i.e., errors logged in the IERRORSTS1 register). This prevents a feedback when two ports monitor each other's AER errors.
- Other bits in the PAERSTS register are set when the corresponding AER error is detected in any of the port functions (i.e., the error is logged in the AERUES or AERCES register of any of the port functions).
  - Note that depending on the port operating mode, some functions are not present in the port. These functions do not have effect on the port's PAERSTS register.

Associated with the PAERSTS register is the software-visible Port AER Mask (PAERMSK) register. The PAERMSK register determines which bits in the PAERSTS register result in a notification being sent to other ports. When any unmasked bit is set in the PAERSTS register of a port, the port notifies all other ports of the occurrence of an AER error. As a result, the bit corresponding to the port that detected the error (i.e., PxAER) is set in the IERRORSTS1 register of all other ports in the switch.

A port that detects an AER error does not notify itself (i.e., the IERRORSTS1 register of a port is not affected by the PAERSTS register associated with that same port).

Figure 4.8 shows a simplified schematic of the connection described above. As shown, the internal error detection logic in a port (e.g., Port 1) is capable of noticing the detection of AER errors by any other port in the switch (e.g., Port 0). That is, when Port 0 detects an AER error in any of its functions, and that error is unmasked in the PAERMSK register in Port 0, the error is notified to Port 1. In Port 1, the P0AER bit is set in the IERRORSTS1 register. Port 1 can be configured to report that error as an AER correctable or uncorrectable internal error (refer to section Internal Errors on page 4-16). AER software can then service Port 1 appropriately. Such software can use the status in Port 1's IERRORSTS0/1 register to determine the exact cause of the internal error. In this example, software can determine that Port 0 had an AER error by noticing that Port 1's P0AER bit is set in the IERRORSTS1 register. This information can then be used to manage the switch appropriately (e.g., reconfigure partitions, etc.)

Notes

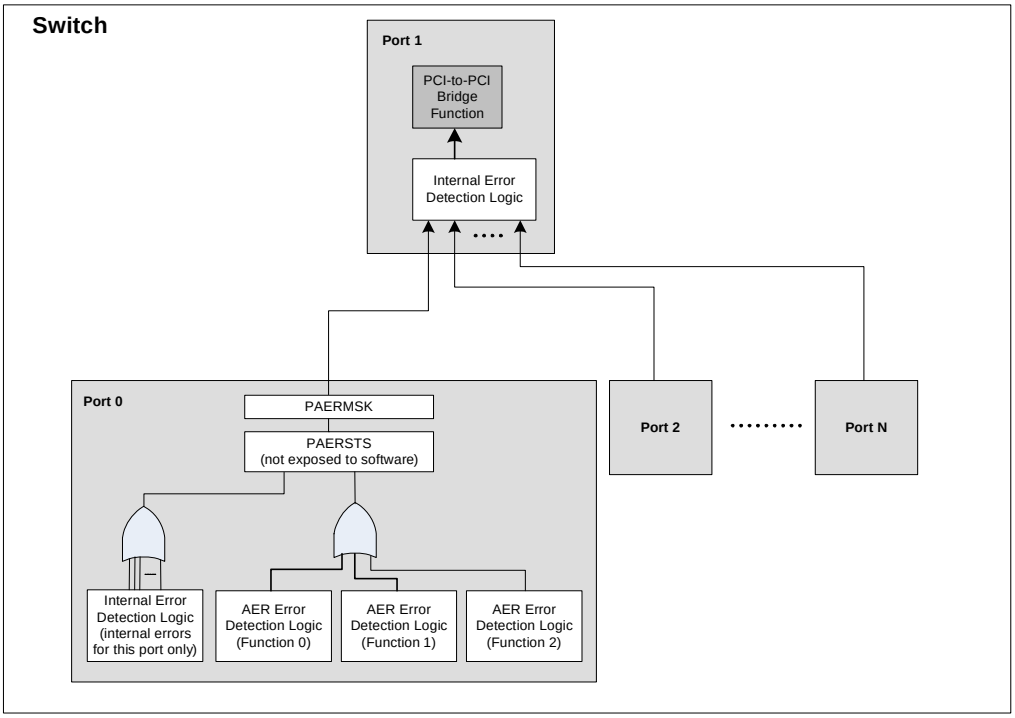

Figure 4.8 Reporting of Port AER Errors as Internal Errors

Notes

Switch Partition and Port Configuration

Notes

Overview

The PES32NT8xG2 supports up to 8 active switch partitions. Each switch partition represents an independent PCI Express hierarchy whose operation is independent of other switch partitions. A port may be configured to operate in one of the following modes.

- Disabled
- Unattached
- Upstream switch port (i.e., upstream PCI-to-PCI bridge)
- Downstream switch port (i.e., downstream PCI-to-PCI bridge)
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function
- NT with DMA function

Ports may be dynamically assigned to partitions, and the operating mode of a switch port may be dynamically reconfigured without affecting in any way unrelated switch partitions.

Switch Partitions

A switch partition represents a logical container to which ports are attached. Each switch partition has an associated ID. The PES32NT8xG2 supports 8 switch partitions with IDs 0 through 7.

- A port is attached to a switch partition by setting the Switch Partition (SWPART) field in the corresponding Switch Port x Control (SWPORTxCTL) register to the ID of the partition to which the port should be attached and setting the Mode (MODE) field in the SWPORTxCTL register to one of the following modes.
- Upstream switch port
- Downstream switch port
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function
- NT with DMA function

A port whose MODE field in the SWPORTxCTL register is set to disabled or unattached mode is not associated with a switch partition. In these modes, the behavior of a port is unaffected by the state of any partition.

The following terms are used throughout this document.

- An *upstream port* is a port attached to a partition and configured to operate in one of the following modes:
  - Upstream switch port
  - Upstream switch port with DMA function
  - Upstream switch port with NT function
  - Upstream switch port with NT and DMA functions
  - NT function

Notes

- NT with DMA function

– A *downstream switch port* is a port configured to operate in downstream switch port mode and attached to a partition.

– An *upstream switch port* is an upstream port with a PCI-to-PCI bridge function (i.e., a port in upstream switch port mode, upstream switch port with DMA function mode, upstream switch port with NT function mode, or upstream switch port with NT and DMA functions mode).

– An *NT endpoint port* is a port configured to operate in NT function mode or NT with DMA function mode, and attached to a partition.

A port in one of the following modes is considered a *multi-function* port:

– Upstream switch port with DMA function

– Upstream switch port with NT function

– Upstream switch port with NT and DMA functions

– NT with DMA function

Multi-function ports follow the rules for multi-function devices outlined in the PCI Express Base Specification Revision 2.1.

Partition Configuration

The following list represents valid switch partition configurations. The behavior of all other configurations is undefined.

– A switch partition with no associated ports

– A switch partition with one upstream switch port and one or more downstream switch ports

– A switch partition with no upstream port and one or more downstream switch ports

– A switch partition with an NT endpoint port

Note the following:

– The Upstream Port (US) bit is set in the Switch Partition Status (SWPARTxSTS) register when there is an upstream port associated with the partition.

– When the US bit is set in the SWPARTxSTS register, the Upstream Port ID (USID) field contains the port ID of the upstream port of the partition.

A switch partition with no associated ports represents a logical entity. The state and configuration of such a partition has no functional effect on the operation of the device. A switch partition with one upstream switch port and one or more downstream switch ports represents a PCI Express switch configuration. Such a configuration operates as a standard PCI Express switch using the associated ports.

– The NT and DMA functions in the upstream switch port, if present, operate independently from the PCI Express switch, but participate in the rules outlined in the PCI Express Base Specification for multi-function devices (i.e., when the NT or DMA functions are present in the upstream switch port, the port is a multi-function device).

A switch partition with one upstream switch port and no downstream switch ports has the following behavior.

– All requests that target the port's PCI-to-PCI bridge function, except configuration requests, are treated as unsupported requests.

– All requests that target the port's NT or DMA functions (if present per the port's operating mode), are handled per the function's determination.

- The request may be accepted or rejected (i.e., unsupported request) depending on the configuration of the function at the time the request is received.

– All received completions that target the port's PCI-to-PCI bridge function are treated as unexpected completions.

– All received completions that target the port's NT or DMA functions (if present per the port's operating mode), are handled per the function's determination.

Notes

- The completion may be expected or unexpected depending on the configuration of the function at the time the completion is received.

– The upstream switch port is allowed to enter and exit L0s and L1 ASPM state without regard to the ASPM state of a downstream switch port (i.e., since there are no downstream switch ports, they play no role in determining when an upstream port enters or exists a low power ASPM state).

- Refer to section Link Active State Power Management (ASPM) on page 7-12 for a description of the rules governing entry into the L0s and L1 ASPM states in an upstream switch port with one or more functions.

A switch partition with no upstream port and one or more downstream switch ports operates in the following manner.

– Peer-to-peer TLP transfers between downstream switch ports are allowed to progress.

– TLPs not destined to a downstream switch port are treated as unsupported requests.

– TLPs generated by the switch and that are normally routed to the root (e.g., INTx messages) are silently discarded.

– Downstream switch ports are allowed to enter and exit L0s and L1 ASPM state without regard to the ASPM state of the upstream port (i.e., since there is no upstream port, then upstream port plays no role in determining when a downstream switch port enters or exists a low power ASPM state).

A switch partition with an NT endpoint port represents a PCI Express endpoint in the PCI Express hierarchy associated with the partition.

– Such a partition must not contain any downstream switch ports. Violating this rule produces undefined results.

A switch partition may have at most one attached upstream port.

– Associating more than one such port with a partition is considered a programming error and produces undefined results.

Partition State

A partition may be in one of three states: Disabled, Fundamental Reset, or Active. The state of a partition is determined by the State (STATE) field in the Switch Partition Control (SWPARTxCTL) register. The initial state of a partition depends on the selected switch mode in the boot configuration vector. See section Partition Resets on page 3-10 for details.

Disabled

A partition in the disabled state represents unused and idle resources. Ports associated with a disabled partition are in a disabled mode (see section Disabled on page 5-7) regardless of the value of the Mode (MODE) field in the Switch Port Control (SWPORTxCTL) register. Transitioning a port into the disabled mode as a result of a partition being transitioned into the disabled state is considered a port operating mode change. See section Port Operating Mode Change on page 5-13 for details.

No hardware-initiated fundamental reset or hot reset is possible in the partition (e.g., the partition ignores the PARTxPERSTN signal and a link-down in the partition's upstream port (if any) does not cause a hot reset).

Fundamental Reset

A partition in the fundamental reset state operates in the same manner as a traditional PCI Express component would with the fundamental reset (PERSTN) signal asserted. This corresponds to a partition fundamental reset. See section Partition Fundamental Reset on page 3-10 for a details. The fundamental reset state provides a software mechanism for resetting all logic and ports associated with a partition.

Register values are initialized on entry to the fundamental reset state. Following initialization, register values may be modified by masters in other partitions or via an external SMBus master.

Notes

The partition fundamental reset condition is considered to persist as long as the STATE field in the SWPARTxCTL register remains in the fundamental reset state. No hardware-initiated hot reset is possible in the partition (e.g., a link-down in the partition's upstream port (if any) does not cause a hot reset). See Table 3.1 for details on reset precedence.

Transitioning a partition from the fundamental reset state to the active state requires that the system meet the requirements associated with a conventional reset outlined in Section 6.6.1 of the PCI Express Base Specification.

Active

A partition in the active state is in the normal operating mode.

Partition State Change

The state of a partition may be modified at any time. Valid partition state transitions are shown in Figure 5.1. State transitions other than those shown in Figure 5.1 produce undefined results.

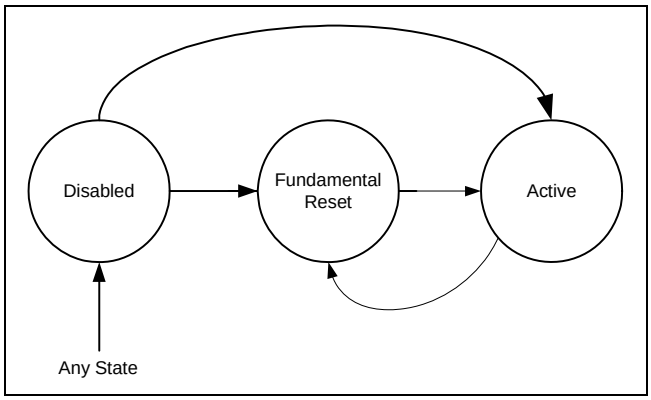

Figure 5.1 Allowable Partition State Transitions

Since a partition state change may take a significant amount of time to complete, status bits are provided to indicate when the change has started and when it has completed.

- The Switch Partition State Change Initiated (SCI) bit in the Switch Partition Status (SWPARTxSTS) register is set when a state change begins.
- The Switch Partition State Change Completed (SCC) bit in the Switch Port Status (SWPORTxSTS) register is set when a state change completes.

Once a partition state change has been initiated, it must be allowed to complete before a new partition state change is initiated on the same partition. Violating this rule produces undefined results. The setting of the SCC bit indicates that all changes associated with the state change in the entire device have completed (e.g., port operating mode changes).

Partition State Change Latency

Partition state changes typically complete within a few clock cycles, unless the following condition is true:

- When the partition state change causes a fundamental reset in the partition (i.e., partition state set to fundamental reset), the latency to complete the state change is 250 microseconds.

This delay ensures that the ingress and egress buffers of ports associated with the partition are fully drained before the partition state change completes.

In addition, when the partition state change has the side-effect of modifying the operating mode of one or more ports (e.g., partition is placed in the disabled state), the partition state change completes after the operating mode of the ports in the partition is changed. Refer to section Port Operating Mode Change on page 5-13 for details on the latency port operating mode changes.

Notes

Partition State Change via EEPROM

When modifying the state of a partition via the serial EEPROM, the following recommendations and requirements apply. Prior to modifying the state of a partition, it is required that the following proprietary timer registers be set to 0x0. This will ensure fast execution of the partition state change action. The last instructions in the EEPROM must set these timers back to their default values.

- Side Effect Delay Timer (SEDELAY register)
- Port Operating Mode Change Drain Delay Timer (POMCDELAY register)
- Reset Drain Delay Timer (RDRAINDELAY register)
- Upstream Secondary Bus Reset Delay (USSBRDELAY register)

After initiating a partition change, it is highly recommended that the EEPROM Wait configuration block (see section Initialization from Serial EEPROM on page 12-3) be embedded in the EEPROM sequence in order to wait for the SCC bit in the SWPARTxSTS register to be set. By doing so, it is possible to obtain an indication of partition state change completion prior to proceeding with other state changes to the same partition.

Switch Ports

A switch port is a logical entity that represents a PCI Express link, stack logic (e.g., physical layer, data link layer, transaction layer, configuration space for each function in the port, etc.) and TLP queues (i.e., port's ingress and egress buffers) required to communicate on that link. All switch ports are completely independent. The configuration or state of one switch port in no way affects the operation or restricts the possible configuration of another port.

Switch Port Mode

A switch port may be configured to operate in one of the following modes.

- Disabled
- Unattached
- Upstream switch port
- Downstream switch port
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function
- NT with DMA function

In general, the mode of a port is determined by the Mode (MODE) field in the Switch Port Control (SWPORTxCTL) register. The only exception is a port that is attached to a partition that is in the disabled state. A disabled partition causes all attached ports to be placed in a disabled mode regardless of the MODE field setting.

A port that is disabled or unattached due to the MODE field of the port being set to that mode is not associated with any switch partition. A port whose MODE field is not set to disabled and that is associated with a partition in the disabled state is disabled but remains associated to that partition.

Notes

A port in an *operational mode* is associated to the partition specified by the Switch Partition (SWPART) field in the corresponding Switch Port Control (SWPORTxCTL) register. The following switch port modes are considered *operational modes*.

- Upstream switch port
- Downstream switch port
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- NT function
- NT with DMA function

A logical representation of a switch port is presented in Figure 5.2.[1] The figure shows a port with three functions (i.e., PCI-to-PCI bridge, NT, and DMA).

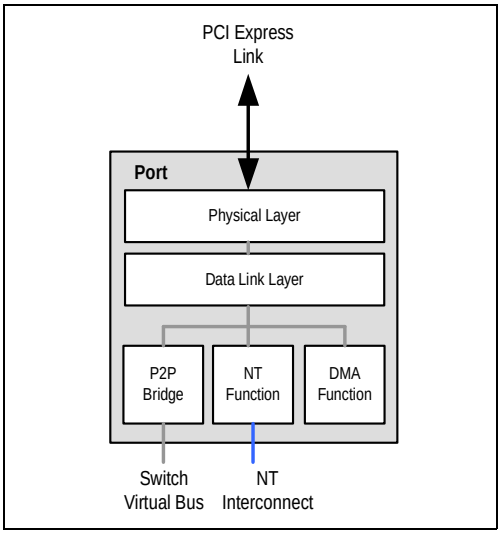

Figure 5.2 Logical Representation of a Port with PCI-to-PCI bridge, NT, and DMA Functions

Depending on the mode of a port, zero, one, two, or three functions may be present in the port. Table 5.1 shows the port operating modes and the functions present in each mode.

- The operation of the PCI-to-PCI bridge function is described in detail in Chapter 10.
- The operation of the NT function is described in detail in Chapter 14.
- The operation of the DMA function is described in detail in Chapter 15.
- The configuration registers associated with each port function are listed in Chapter 19.
- A function that is not present in a port is quiesced and has no effect on the operation of the port. Specifically, a function that is not present in the port adheres to the following.
  - Ignores all received PCI Express requests (e.g., configuration requests, memory read or write requests, etc.)
  - Does not generate any TLPs (e.g., MSI, error messages, INTx messages, etc.)
  - Does not participate on the rules that govern multi-function ports (e.g., link power state, error message generation, etc.)
  - Does not log any non-function specific errors detected by the port.

1. For simplicity, the figure does not show the TLP ingress and egress buffers associated with the port.

Notes

Regardless of a port's operating mode, all registers in all functions of the port remain accessible via the switch's global address space, via the SMBus slave interface, or via serial EEPROM (see Chapter 19, Register Organization). In addition, proprietary port registers (refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11) continue to have effect on the operation of the port.

Proprietary port registers are mapped to the PCI-to-PCI bridge function's configuration space. For port operating modes in which the PCI-to-PCI bridge function is not present in the port, the proprietary port registers in this function are only accessible via the switch's global address space and continue to affect the operation of the port.

Port Operating Mode	Port Functions
Disabled	None.
Unattached	Function 0: PCI-to-PCI bridge, but only responds to accesses to the GASAADDR and GAS-ADATA registers (see section Unattached on page 5-8).
Upstream switch port	Function 0: PCI-to-PCI bridge
Downstream switch port	Function 0: PCI-to-PCI bridge
Upstream switch port with DMA	Function 0: PCI-to-PCI bridge Function 2: DMA
Upstream switch port with NT	Function 0: PCI-to-PCI bridge Function 1: NT
Upstream switch port with NT and DMA	Function 0: PCI-to-PCI bridge Function 1: NT Function 2: DMA
NT function	Function 0: NT
NT with DMA function	Function 0: NT Function 2: DMA

Table 5.1 Port Functions for Each Port Operating Mode

Disabled

A port that is disabled is considered unused and is placed in a low power state. For example, the port does not generate power management messages, INTx or MSI interrupts, error messages, etc. A port in the disabled mode has the following behavior.

– All output signals associated with the port are placed in a negated state (e.g., link status and hot-plug signals).

The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2. PxACTIVEN and PxLINKUPN are negated. All input signals associated with the port are ignored and have no effect on the operation of the device. The state of the following hot-plug input signals is ignored: PxAPN, PxMRLN, PxPDN, PxPFN, and PxPWRGDN.

Boot configuration vector signals are sampled during a switch fundamental reset and thus their dynamic state has no effect on the operating mode of the port in any port mode. For example, if the PxyMERGEN signal associated with the port was asserted during a switch fundamental reset, then the port remains merged while disabled.

Notes

A port is not associated with any switch partition if the disabled port mode is due to the Port Mode (MODE) field in the Switch Port Control (SWPORTxCTL) register being set to Disabled. Since the port is not associated with a switch partition in this mode, the port is unaffected by the state of any switch partition, and vice-versa. Additionally, the state of no switch partition input signal (e.g., PARTxPERSTN) has effect on the port.

A port that is disabled due to the corresponding partition state being set to disabled remains associated to that partition. A port's MODE field being set to disabled takes precedence over the partition state. This means that if the MODE field of a port is set to disabled, then the port is not associated with any partition.

The PHY layer's Link Training & Status State Machine (LTSSM) immediately transitions to the Detect state, the SerDes lanes associated with the port are powered down[1], and the termination associated with SerDes receive lanes is turned off.

- The SerDes lanes remain powered down with the termination off as long as the port is disabled.
- Unused logic is placed in a low power state (e.g., clock is gated).
- All registers associated with the port remain accessible from the global address space.
- PCI Express configuration requests targeting the port are not possible.

Register state is preserved in the disabled state. Therefore, transitioning a port from an operational state to a disabled state and then back to an operational state results in no change the register values except for fields that report status. All configuration registers in all functions associated with the port are accessible through the global address space by the serial EEPROM, other ports, and the SMBus. In this mode, all registers in the port's PCI-to-PCI bridge function (if present) are organized as that shown in Table 19.2 for a downstream switch port. PCI-to-PCI bridge register fields whose value is dependent on port mode (i.e., upstream or downstream) are configured to operate as a downstream switch port.

Unattached

An unattached port is a port not associated with a switch partition and except for the link, is not operational. A port in the unattached mode has the following behavior.

- The physical and data-link layers remain operational and the link behaves as an upstream port (i.e., downstream component of the link). The LTSSM transitions to and remains in the L0 state. If necessary, the link trains from the Detect state.
- The transaction layer functionality necessary to respond to configuration requests and manage flow control remains active. All other logic associated with the port is disabled. For example, the port does not generate power management messages, it does not generate INTx or MSI interrupts, and it does not generate error messages.
- All output signals associated with the port, except signals associated with the link, are placed in a negated state, except the link status and activity signals. The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2. The link status and activity signals (PxACTIVEN and PxLINKUPN) remain active.
- All input signals associated with the port, except inputs from the SerDes, are ignored and have no effect on the operation of the device.
  - The state of the following hot-plug input signals is ignored: PxAPN, PxMRLN, PxPDN, PxPFN, and PxPWRGDN.
  - Boot configuration vector signals are sampled during a switch fundamental reset and thus their dynamic state has no effect on the operating mode of the port in any port mode.
  - Since a port in this state is not associated with a switch partition, the state of no switch partition input signal (e.g., PARTxPERSTN) has effect on the port.

Since the port is not associated with a switch partition in this state, the port is unaffected by the state of any switch partition, and vice-versa. An unattached port must only receive configuration request TLPs. The operation of the port is undefined when TLPs other than configuration request TLPs are received by the port.

1. Refer to section SerDes Power Management on page 8-14 for further details.

Notes

The port responds to received TLPs as follows:

- All received PCI Express configuration requests that do not target function 0 are completed with a configuration-request-retry-status completion. The intent of this requirement is to prevent standard enumeration software from detecting the existence of port functions which may not be present in the port after the partition is configured.
- All received requests that target function 0 but do not target function 0's Global Address Space Access Address or Data (GASAADDR or GASADATA) registers are completed with a configuration-request-retry-status completion. The intent of this requirement is to prevent standard enumeration of the unattached port by its root-complex.
  - Function 0 responds to received PCI Express configuration requests that target the GASAADDR and GASADATA normally (i.e., the requested action is performed and a completion is generated). A root-complex with customized switch configuration software may access the switch global address space (and therefore all configuration registers in the device) by accessing the GASAADDR and GASADATA registers in the unattached port. By accessing the global address space, the root may reconfigure all ports and partitions in the switch.
- All other received requests are treated as unsupported requests (i.e., logging error status and if appropriate, generating the appropriate completion). The generated completion (if any) has a completer ID of zero.
- All registers in the port remain accessible from the global address space.

Configuration registers are accessible through the global address space by the serial EEPROM, other ports, and the SMBus. Although the link in this mode behaves as an upstream port, all registers in the port's PCI-to-PCI bridge function take on the organization and initial values shown in Table 19.2 for a downstream switch port.

Registers and fields in the PCI-to-PCI bridge function that affect the behavior of the link (listed below) operate normally (i.e., control fields control behavior and status fields report status) as though the port were an upstream port.

- PCIELCTL (all fields related to link/phy)
- PCIELSTS (all fields related to link/phy)
- PCIELCTL2 (all fields related to link/phy)
- PCIELSTS2 (all fields related to link/phy)
- SERDESCFG
- LANESTS[1:0]
- PHYPRBS

PCI-to-PCI bridge function registers other than those listed above operate as though the port were a downstream switch port and take on the initial value of a downstream switch port. A port in this mode is unaffected by the following:

- The reception of TS1 ordered-sets indicating a hot reset.
- The data link layer of the port transitioning to the DL_Down state.
- The reception of a Set_Slot_Power_Limit message.

Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0. Automatic speed change may be enabled by modifying the value of the Initial Link Speed Change Control (ILSCC) bit in the port's Phy Link Configuration 0 (PHYLCFG0) register.

Upstream Switch Port

A port in upstream switch port mode is composed of a PCI-to-PCI bridge function and has the following behavior.

- Behaves as an upstream switch port as defined in the PCI Express Base Specification.
  - All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals). The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.
- The LTSSM is operational and behaves as an upstream port.

Notes

- Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0. Automatic speed change may be enabled by modifying the value of the Initial Link Speed Change Control (ILSCC) bit in the PCI-to-PCI bridge function's Phy Link Configuration 0 (PHYLCFG0) register.

The PCI-to-PCI bridge function associated with the upstream port has an associated type 1 header. The function number is zero.

PCI Express requests that do not target function 0 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

Downstream Switch Port

A port in downstream switch port mode is composed of a PCI-to-PCI bridge function and has the following behavior.

- Behaves as a downstream switch port as defined in the PCI Express Base Specification.
- The LTSSM is operational and behaves as a downstream switch port.
  - Since the link operates as a downstream switch port (i.e., upstream component), an automatic speed change is initiated when the link enters L0.

The PCI-to-PCI bridge function associated with a downstream switch port has an associated type 1 header. The Device Number (DEVNUM) field in the SWPORTxCTL register determines the device number of the downstream switch port within the switch partition. The function number of the PCI-to-PCI bridge function in a downstream switch port is always zero.

The behavior when two or more downstream switch ports in the same partition are configured with the same device number is undefined. Since partitions are independent, two downstream switch ports in different partitions may share the same device number.

If the DEVNUM field is modified using a PCI Express configuration write request, then the modification takes place prior to the generation of a completion for the request. Still, the completion uses the old (i.e., unmodified) device number (since this is the number expected by the requester).

- Modifying the DEVNUM field using a PCI configuration write request does not modify the bus and device numbers captured by the PCI-to-PCI bridge. Therefore, a subsequent Type 0 configuration write must be performed to a PCI-to-PCI bridge following modification of its device number.
  - Failure to follow this procedure will result in an incorrect value in the requester and completer ID fields of TLPs generated by the PCI-to-PCI bridge.
  - As expected, modifying the DEVNUM field via SMBus or Serial EEPROM does not modify the bus and device numbers captured by the PCI-to-PCI bridge.

PCI Express requests that do not target function 0 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

Upstream Switch Port with DMA Function

A port in upstream switch port with DMA function mode is a multi-function upstream port composed of a PCI-to-PCI bridge function and a Direct Memory Access Controller (DMA) function.

- The PCI-to-PCI bridge function is function 0 of the port (i.e., type 1 header).
- The DMA function is function 2 of the port (i.e., type 0 header).

The port has the following behavior:

- The PCI-to-PCI bridge function has the behavior of an upstream switch port defined in the PCI Express Base Specification.

  All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals). The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.
- The DMA function behaves as described in Chapter 15.
- The port follows the behavior for multi-function ports described in the PCI Express Base Specification and in this document.
- The LTSSM is operational and behaves as an upstream port.

Notes

Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0.

Automatic speed change may be enabled by modifying the value of the Initial Link Speed Change Control (ILSCC) bit in the PCI-to-PCI bridge function's Phy Link Configuration 0 (PHYLCFG0) register.

– PCI Express requests that do not target function 0 or function 2 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

Upstream Switch Port with NT Function

A port in upstream switch port with NT function mode is a multi-function upstream port composed of a PCI-to-PCI bridge function and a non-transparent-bridge (NT) function.

– The PCI-to-PCI bridge function is function 0 of the port (i.e., type 1 header).
– The NT function is function 1 of the port (i.e., type 0 header).

The port has the following behavior.

– The PCI-to-PCI bridge function has the behavior of an upstream switch port defined in the PCI Express Base Specification. All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals). The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.
– The NT function behaves as described in Chapter 14.
– The port follows the behavior for multi-function ports described in the PCI Express Base Specification and in this document.
– The LTSSM is operational and behaves as an upstream port.
  • Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0.
  • Automatic speed change may be enabled by modifying the value of the Initial Link Speed Change Control (ILSCC) bit in the PCI-to-PCI bridge function's Phy Link Configuration 0 (PHYLCFG0) register.
– PCI Express requests that do not target function 0 or function 1 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

Upstream Switch Port with NT and DMA Functions

A port in upstream switch port with NT and DMA functions is a multi-function upstream port composed of a PCI-to-PCI bridge function, a non-transparent-bridge (NT) function, and a DMA function.

– The PCI-to-PCI bridge function is function 0 of the port (i.e., type 1 header).
– The NT function is function 1 of the port (i.e., type 0 header).
– The DMA function is function 2 of the port (i.e., type 0 header).

The port has the following behavior.

– The PCI-to-PCI bridge function has the behavior of an upstream switch port defined in the PCI Express Base Specification.
  • All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals). The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.
– The NT function behaves as described in Chapter 14.
– The DMA function behaves as described in Chapter 15.
– The port follows the behavior for multi-function ports described in the PCI Express Base Specification and in this document.
– The LTSSM is operational and behaves as an upstream port.
  • Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0.

Notes

- Automatic speed change may be enabled by modifying the value of the Initial Link Speed Change Control (ILSCC) bit in the PCI-to-PCI bridge function's Phy Link Configuration 0 (PHYLCFG0) register.

– PCI Express requests that do not target functions 0, 1, 2 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

NT Function Port

A port in NT function mode is a port composed of a non-transparent-bridge (NT) function.

– The NT function is function 0 of the port (i.e., type 0 header).

The port has the following behavior.

– The NT function behaves as described in Chapter 14.

– The LTSSM is operational and behaves as an upstream port.

- Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0.
- The NT function does not implement the ILSCC bit. Still, although the PCI-to-PCI bridge function is not logically present in the port, proprietary registers within the function remain active. Therefore, it is possible to program the ILSCC bit in a port that operates in NT function mode by accessing the port's PCI-to-PCI bridge function registers via the global address space access registers (GASAADDR and GASADATA) located in the NT function's configuration space.

– Port output signals not associated with the NT function are kept in a negated state.

- All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals). The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2.

– PCI Express requests that do not target function 0 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

NT with DMA Function Port

A port in NT with DMA function mode is a multi-function upstream port composed of a non-transparent-bridge (NT) function and a DMA function.

– The NT function is function 0 of the port (i.e., type 0 header).

– The DMA function is function 2 of the port (i.e., type 0 header).

The port has the following behavior.

– The NT function behaves as described in Chapter 14.

– The DMA function behaves as described in Chapter 15.

– The port follows the behavior for multi-function ports described in the PCI Express Base Specification and in this document.

– The LTSSM is operational and behaves as an upstream port.

- Since the link operates as an upstream port (i.e., downstream component), an automatic speed change is not initiated when the link enters L0.
- The NT and DMA functions do not implement the ILSCC bit. Still, although the PCI-to-PCI bridge function is not logically present in the port, proprietary registers within this function remain active. Therefore, it is possible to program the ILSCC bit in a port that operates in NT function mode or NT with DMA function mode by accessing the port's PCI-to-PCI bridge function registers via the global address space access registers (GASAADDR and GASADATA) located in the NT or DMA function's configuration space.

– Port output signals not associated with the NT function are kept in a negated state.

- All port output signals associated with downstream switch port operation are placed in a negated state (i.e., hot-plug signals).

Notes

- The negated value of PxAIN, PxILOCKP, PxPEP, PxPIN, and PxRSTN is determined as shown in Table 11.2. PCI Express requests that do not target function 0 or function 2 are completed with unsupported request status by the port. The completion has a value of all zeroes in the function number field of the completer ID.

Port Operating Mode Change

The operating mode of a port is determined by the Port Mode (MODE), Switch Partition (SWPART), and Device Number (DEVNUM) fields in the SWPORTxCTL register as well as, in some cases, the state (STATE) field in the corresponding partition control (SWPARTxCTL) register. The initial operating mode of a port is determined by the switch mode setting in the boot configuration vector sampled during a switch fundamental reset. Refer to section Partition Resets on page 3-10.

The following events constitute a port operating mode change and initiate the action specified by the Operating Mode Change Action (OMA) field in the corresponding SWPORTxCTL register.

- Any modification of the value in the MODE, SWPART[1], or DEVNUM[2] fields in the SWPORTxCTL register.
  - A write of the same value already contained in these fields does not result in a port operating mode change.
- Setting the STATE field in the SWPARTxCTL register of the partition with which the port is associated to disabled.
  - When a port is disabled due to the STATE field in the corresponding partition SWPARTxCTL field being set to disabled, modification of values in the MODE, SWPART, or DEVNUM fields that do not modify the partition with which the port is associated result in the actions associated with an operating mode change (e.g., OMCI and OMCC are set), but the port remains in the disabled mode during and after the operating mode change process.
  - The modification of the MODE field to disabled causes the port to unattach from the disabled partition; however, the port remains in the disabled mode. All of the actions associated with an operating mode change (e.g., OMCI and OMCC are set) continue to operate as described above.
  - The modification of the SWPART field causes the port to unattach from the disabled partition. The new operating mode is dependent on the MODE field and the state of the new partition with which the port is associated (if any).

Table 5.2shows the port operating mode transitions that are supported as well as those that are not supported. Entries marked in yellow correspond to port operating mode changes that are supported only when any of the following conditions are satisfied:

- The OMA field in the port's SWPORTxCTL register is set to reset.
- If the operating mode of the port before the change has a DMA function present, then the DMA function must be quiesced prior to the port operating mode change. In addition, the OMA field must be set to reset when modifying the operating mode of a port configured to transmit NT Multicast TLPs. Refer to section Usage Restrictions on page 17-11 for additional information.

Violating these rules produces undefined results.

1. Note: Modifying the switch partition number of a port in unattached mode does not make logical sense (i.e., since unattached ports are not associated with any partition). Doing this produces undefined results.

2. Note that the DEVNUM field is only applicable to ports configured in downstream switch port mode.

Notes

		TO								
		UN	DIS	US	US + NT	NT	US + DMA	NT + DMA	US + NT + DMA	DS
FROM	UNATTACHED (UN)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	DISABLED (DIS)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	UPSTREAM (US)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	US + NT	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
	NT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	US + DMA	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No
	NT + DMA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	US + NT + DMA	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No
	DS	Yes	Yes	No	No	No	No	No	No	Yes

Table 5.2 Port Operating Mode Changes Supported by the Switch

Note that the port operating mode changes shown as not supported in Table 5.2 only apply for direct transitions between the operating modes. Indirect transitions between such operating modes are possible. For example, the following indirect transition between upstream and downstream port operating modes is supported:

– Upstream switch port ⇒ Unattached ⇒ Downstream switch port

Since an operating mode change may take a significant amount of time to complete (see next section), status bits are provided to indicate when the change has started and when it has completed.

– The Operating Mode Change Initiated (OMCI) bit in the Switch Port Status (SWPORTxSTS) register is set when a mode change begins.
– The Operating Mode Change Completed (OMCC) bit in the Switch Port Status (SWPORTxSTS) register is set when a mode change completes.

Once an operating mode change to a port has been initiated, this operating mode change must be allowed to complete before a new operating mode change is initiated on the same port. Violating this rule produces undefined results.

Port Operating Mode Change Latency

The latency to complete a port operating mode change depends on the following factors:

– The reset action on the port, as dictated by the OMA field in the port's SWPORTxCTL register.
– The TLP traffic conditions on the port at the time the port operating mode change takes place.

By default, the latency to complete a port operating mode change is set to 2 ms if the port's OMA field is set to 'no action'. If the port's OMA field is set to 'reset', the latency is 2.5 ms.

This latency is large to ensure that TLPs held in the port's ingress buffers are sent towards their destination port before the port is migrated across partitions. This assures that after the port is migrated, the port will not emit TLPs from the old partition into its new partition. Depending on traffic congestion in the system, TLPs held in the port's ingress buffer may take from a few nanoseconds to several microseconds before they are transferred to the egress port's EFB.

Notes

Port Operating Mode Change via EEPROM

When modifying the operating mode of a port via the serial EEPROM, the following recommendations and requirements apply.

- Prior to modifying the port operating mode, it is required that the following proprietary timer registers be set to 0x0. This will ensure fast execution of the port mode change action. The last instructions in the EEPROM must set these timers back to their default values.
  - Side Effect Delay Timer (SEDELAY register)
  - Port Operating Mode Change Drain Delay Timer (POMCDELAY register)
  - Reset Drain Delay Timer (RDRAINDELAY register)
  - Upstream Secondary Bus Reset Delay (USSBRDELAY register)
- After initiating a port operating mode change, it is highly recommended that the EEPROM Wait configuration block (see section Initialization from Serial EEPROM on page 12-3) be embedded in the EEPROM sequence in order to wait for the OMCC bit in the SWPORTxSTS register to be set. By doing so, it is possible to obtain an indication of port operating mode change completion prior to proceeding with other port operating mode changes to the same port.

Common Operating Mode Change Behavior

This section specifies common port operating mode change behavior that occurs regardless of the OMA field setting. The following sections describe behaviors associated with specific OMA field settings. Modifying the operating mode of a port does not require that traffic on that port be stopped during modification of the operating mode (i.e., it is not necessary to quiesce the traffic). TLPs flowing through the port during the modification may be discarded; however, normal operation resumes following the port modification.

Any modification to the operating mode of a port has the following common behavior.

- TLPs received by the port from the link before the operating mode modification is initiated are treated in a manner consistent with the old operating mode of the port, and if appropriate, corresponding partition.
- TLPs received by the port from the link after the operating mode modification is completed are treated in a manner consistent with the new operating mode of the port, and if appropriate, corresponding partition.
- TLPs received by the port from the link during the operating mode modification (i.e., in the time period between the initiation and completion of the change) may be silently dropped.

A port operating mode change may result in a port being removed from a partition, a port being added to a partition, both a removal and an addition, or neither.

- The partition from which a port is removed is referred to as the *source partition*.
- The partition to which a port is added is referred to as the *destination partition*.
- When a port operating mode changes within a partition, it may have a substantial effect on the partition and is thus logically viewed as a removal followed by an addition of the same port to the same partition.

A port operating mode change that is caused only by a device number change is logically viewed as a source partition removal, followed by a device number change, followed by a destination partition addition to the same partition. The intent of this requirement is to make a device number change operate in the same manner as all other port operating mode changes. When a port operating mode change is initiated, the operation logically executes in the following order.

1. The OMCI bit in the SWPORTxSTS register is set.
2. The effect on the source partition, if appropriate, takes place (i.e., cleanly remove the port from the partition).
3. The effect on the destination partition, if appropriate, takes place (i.e., cleanly add the port to the partition).
4. The effect on the port, if appropriate, takes place (i.e., as dictated by the OMA field).
5. The OMCC bit in the SWPORTxSTS register is set.

Notes

Mode Change Effect on Source Partition

A port operating mode change that results in a port being removed from a partition has the following effect on that partition (i.e., the source partition). If the port being removed is an upstream port, the removal of the port results in the partition behaving as described in section Partition Configuration on page 5-2.

- If the upstream port being removed was the only port in the partition (e.g., removal of an NT endpoint port), the partition becomes an empty partition.
- If the port being removed was an upstream switch port, the partition behaves as a partition with no upstream port and one or more downstream switch ports.

Partition Hot Reset

See section Partition Hot Reset on page 3-11 for an overview of partition hot reset. The removal of an upstream port that is initiating a partition hot reset (i.e., as the result of reception of TS1 ordered sets with the hot reset bit set or DL_Down) has the effect of removing the hot reset condition on the partition.

- If the upstream port has a PCI-to-PCI bridge function, removing the hot reset condition on the partition implies removing the hot-reset from affecting the partition's virtual PCI bus.

The removal of a downstream switch port that is affected by a source partition hot reset has no effect on the source partition (i.e., the hot reset operation continues normally on the other ports in the partition).

- The removed downstream switch port stops participating in partition hot reset(s) after the removal of the port has been completed.

Partition Upstream Secondary Bus Reset

See section Partition Upstream Secondary Bus Reset on page 3-12 for an overview of partition upstream secondary bus reset. The removal of an upstream switch port whose SRESET bit in the BCTL register of the PCI-to-PCI bridge function is set, has the same effect as clearing of the SRESET bit (i.e., the partition upstream secondary bus reset condition is removed).

The removal of a downstream switch port that is affected by a source partition upstream secondary bus reset has no effect on the source partition (i.e., the hot reset operation continues normally on the other ports in the partition).

- The removed downstream switch port stops participating in the upstream secondary bus reset after the removal of the port has been completed.

Partition Downstream Secondary Bus Reset

See section Partition Downstream Secondary Bus Reset on page 3-12 for an overview of partition downstream secondary bus reset. The removal of a downstream switch port whose SRESET bit in the BCTL register is set has no effect on the source partition since other ports in the partition are unaffected by this type of reset (i.e., the hot reset is propagated to the endpoint located below the port that is being removed).

Routing

Removing a port from a partition results in the corresponding invalidation of routes to all functions of that port. For example, removing a downstream switch port from a partition causes all other downstream switch ports in the partition, as well as the upstream port's PCI-to-PCI bridge function, to invalidate routes to the moved downstream switch port. As a result, TLPs destined to the moved port will be treated as unsupported requests by the PCI-to-PCI bridge function that first receives the request.

Also, removing an upstream switch port from a partition causes all downstream switch ports in the partition to invalidate routes to the function(s) in the moved upstream port. As a result, TLPs received by a downstream switch port and destined to a function in the moved port will be treated as unsupported requests by the PCI-to-PCI bridge function in the receiving port.

Finally, removing an upstream port that contains an NT function from a partition causes all other NT functions to invalidate routes to the affected partition (i.e., the affected partition no longer has an NT function that can emit the TLPs). This is considered a destination partition error by the NT function that receives the TLP (refer to section Transaction Layer Errors on page 14-26).

Notes

L0s ASPM

A switch partition exhibits a correlation between the L0s ASPM state of its upstream switch port and its downstream switch port(s). Refer to section Link Active State Power Management (ASPM) on page 7-12 and to the PCI Express Base Specification 2.1 for details.

– Downstream switch port removal

  A switch partition must initiate an exit from L0s on the transmitter of the upstream switch port if it detects an exit from L0s on the receiver of any downstream switch port.

  Removing a downstream switch port from a partition removes it from affecting the L0s ASPM state of upstream switch port in the source partition.

– Upstream switch port removal

  A switch partition must initiate an exit from L0s on the transmitters of all downstream switch ports associated with the partition if it detects an exit from L0s on the receiver of its upstream switch port.

  Removing an upstream switch port from a partition removes it from affecting the L0s ASPM state of downstream switch ports of the source partition.

L1 ASPM

An upstream switch port is not allowed to initiate entry into L1 unless all of the downstream switch ports associated within the partition are in an L1 (or deeper) state. Refer to section Link Active State Power Management (ASPM) on page 7-12 and to the PCI Express Base Specification 2.1 for details.

– Downstream switch port removal

  Removing a downstream switch port from a partition removes it from affecting the L1 ASPM state of the upstream switch port in the source partition.

– Upstream switch port removal

  Removing an upstream switch port from a partition has no effect on the L1 ASPM state of downstream switch ports associated with the partition.

PME Synchronization

Removing a port from a partition has the effect of removing it from participation in PME synchronization associated with the source partition. If PME synchronization is in progress, then PME synchronization completes before the port (upstream or downstream) is removed from the partition.

– For a port in upstream switch port mode, PME synchronization completes when the port aggregates PME_TO_Ack messages from all downstream switch ports in the partition or when the upstream port abandons the aggregation.
– For a port in NT function or NT function with DMA mode, PME synchronization completes when the port generates a PME_TO_Ack message.
– For a port in upstream switch port with NT function mode or upstream switch port with NT and DMA function mode, PME synchronization completes when the port aggregates PME_TO_Ack messages from all downstream switch ports in the partition or when the upstream port abandons the aggregation.
– For a downstream switch port, PME synchronization completes when the port notifies the upstream port of the reception of a PME_TO_Ack message or a PME_TO_Ack timer timeout.

In this scenario, the OMCI field in the port's SWPORTxSTS register is set after the operating mode change is requested, and the OMCC field in this same register is set after the PME synchronization has finished and the operating mode change completes.

Bus Locking

Removing the upstream switch port from a partition immediately unlocks the partition. Removing a downstream switch port that is locked has the effect of unlocking the partition. Removing a downstream switch port that is not locked from a locked partition has no effect on the locked nature of the partition.

Notes

INTx Interrupt Signaling

Removing an upstream port from a partition has no effect on the partition since the interrupt state is associated with the root located above the upstream port that is being removed. An INTx state change signaled by a downstream switch port in the source partition has no effect on the upstream port as the latter is no longer associated with the partition.

Removing a downstream switch port from a partition has the affect of removing all INTx virtual wire assertions associated with the port (i.e., INTA, INTB, INTC and INTD from the port are negated).

Mode Change Effect on Destination Partition

A port operating mode change that results in a port being added to a partition has the following effect on that partition (i.e., the destination partition). Some of the behaviors described below rely on the state of the port. The state of the port and its corresponding configuration registers are determined by the OMA field as described in section No Action Mode Change Behavior on page 5-21.

Partition Hot Reset

See section Partition Hot Reset on page 3-11 for an overview of partition hot reset. The addition of an upstream port that is initiating a hot reset (i.e., as the result of reception of TS1 ordered sets with the hot reset bit set or DL_Down) has the effect of initiating a partition hot reset to the destination partition as described in section Partition Hot Reset on page 3-11.

– An upstream port whose link transitioned to the Detect state (i.e., DL_Down) as a result of the operating mode change may trigger a hot reset in the destination partition as described in section Partition Hot Reset on page 3-11.

The addition of a downstream switch port to a destination partition that has a partition hot reset in progress does not have an effect on the ongoing hot reset in the destination partition (i.e., the hot reset operation continues normally on the other ports in the partition). The added downstream switch port participates in the ongoing hot reset after the addition of the port has completed.

Partition Upstream Secondary Bus Reset

See section Partition Upstream Secondary Bus Reset on page 3-12 for an overview of partition upstream secondary bus reset. The addition of an upstream switch port whose SRESET bit in the BCTL register of the port's PCI-to-PCI bridge function is set, has the effect of initiating a partition upstream secondary bus reset. The addition of a downstream switch port to a destination partition that has an upstream secondary bus reset in progress does not have an effect on the ongoing hot reset in the destination partition (i.e., the hot reset operation continues normally on the other ports in the partition). The added downstream switch port participates in the ongoing hot reset after the addition of the port has completed.

Partition Downstream Secondary Bus Reset

See section Partition Downstream Secondary Bus Reset on page 3-12 for an overview of partition downstream secondary bus reset. The addition of a downstream switch port whose SRESET bit in the BCTL register is set has no effect on the destination partition since other ports in the partition are unaffected by this type of reset (i.e., the hot reset is propagated to the endpoint located below the port that is being removed).

Routing

Adding a port to a partition results in the routing specified by the configuration registers associated with that port being enabled in the destination partition. For example, adding a downstream switch port to a partition causes all other ports in the partition to validate routes to the moved downstream switch port. Also, adding an upstream port to a partition causes all downstream switch ports in the partition to validate routes to the function(s) in the moved upstream port.

Finally, adding an upstream port that contains an NT function to a partition causes all other NT functions to validate routes to the affected partition.

Notes

L0s ASPM

A switch partition exhibits a correlation between the L0s ASPM state of its upstream and downstream switch port(s). Refer to section Link Active State Power Management (ASPM) on page 7-12 and to the PCI Express Base Specification 2.1 for details.

Downstream switch port addition

A switch partition must initiate an exit from L0s on the transmitter of the upstream switch port if it detects an exit from L0s on the receiver of any downstream switch port. Adding a downstream switch port to a partition causes it to affect the L0s ASPM state of upstream switch port in the destination partition.

- If the upstream switch port's transmitter is in L0 and a downstream switch port whose receiver is in L0s is added to the partition, then an entry to L0s may be initiated on the transmitter of the upstream port per the rules described in section Link Active State Power Management (ASPM) on page 7-12.
- If the upstream port's transmitter is in L0s and a downstream switch port whose receiver is in L0 is added to the partition, then an exit from L0s is initiated on the transmitter of the upstream port.

Adding a downstream switch port to a partition causes the added port to be affected by the L0s ASPM state of the upstream switch port in the destination partition. For example, if a downstream switch port whose transmitter is in L0 is added to a switch partition whose upstream switch port receiver is in L0s, then an entry to L0s is initiated on the added port's transmitter per the rules described in section Link Active State Power Management (ASPM) on page 7-12.

Also, if a downstream switch port whose transmitter is in L0s is added to a switch partition whose upstream switch port receiver is in L0, then an exit from L0s is initiated on the added port's transmitter.

Upstream switch port addition

A switch partition must initiate an exit from L0s on the transmitters of all downstream switch ports associated with the partition if it detects an exit from L0s on the receiver of its upstream switch port. See the PCI Express Base Specification for details. Adding an upstream switch port to a partition causes it to affect the L0s ASPM state of downstream switch ports in the destination partition.

- If an upstream switch port whose receiver is in L0s is added to a switch partition, then an entry to L0s is initiated on the transmitter of all downstream switch ports in L0 per the rules described in section Link Active State Power Management (ASPM) on page 7-12.
- If an upstream switch port whose receiver is in L0 is added to a switch partition, then an exit from L0s is initiated on the transmitter of all downstream switch ports that are in L0s.

Adding an upstream switch port to a partition causes the added port to be affected by the L0s ASPM state of downstream switch ports in the destination partition. For example, if an upstream switch port whose transmitter is in L0 is added to a switch partition whose downstream switch ports are in L0s, then an entry to L0s may initiated on the port's transmitter per the rules described in section Link Active State Power Management (ASPM) on page 7-12.

Also, if an upstream switch port whose transmitter is in L0s is added to a switch partition whose downstream switch ports are in L0, then an exit from L0s is initiated on the port's transmitter.

L1 ASPM

Downstream switch port addition

An upstream switch port is not allowed to initiate entry into L1 unless all of the downstream switch ports associated with the partition are in an L1 (or deeper) state. See section Link Active State Power Management (ASPM) on page 7-12 and to the PCI Express Base Specification for details. Adding a downstream switch port to a partition causes it to affect the L1 ASPM initiation of the upstream switch port. For example, if a downstream switch port in L0 is added to a switch partition whose upstream switch port is in L1 ASPM, then an exit from L1 is initiated on the upstream switch port.

Adding a downstream switch port to a partition causes the added port to be affected by the ASPM state of the upstream switch port in the destination partition. For example, if a downstream switch port in L1 ASPM is added to a switch partition whose upstream switch port is in L0, then an exit from L1 is initiated on the added port.

Notes

Upstream switch port addition

Adding an upstream switch port to a partition causes it to affect the L1 ASPM state of downstream switch ports in the destination partition. For example, if an upstream switch port in L0 is added to a switch partition, then an exit from L1 is initiated on all downstream switch ports in L1 ASPM.

Adding an upstream switch port to a partition where all downstream switch ports are in L1 ASPM causes the upstream switch port to enter the L1 ASPM state per the rules described in section Link Active State Power Management (ASPM) on page 7-12.

PME Synchronization

If PME synchronization is in progress then the port being added must be a downstream switch port[1]. This downstream switch port being added does not participate in PME synchronization in progress within the destination partition. The added downstream switch port participates in PME synchronization initiated within the destination partition after the addition of the port has been completed.

Bus Locking

If the destination partition is bus locked, then the port being added must be a downstream switch port[2]. By the nature that the port is added to a locked partition, it cannot be one of the two locked ports. Adding a downstream switch port to a partition that is locked causes it to adopt the locked behavior associated with the destination partition. The port blocks all requests from being propagated to either of the locked ports.

INTx Interrupt Signaling

Adding an upstream switch port to a partition causes the PCI-to-PCI bridge function in the upstream switch port to adopt the aggregated interrupt state of the downstream switch ports associated with the destination partition. This may result in the generation of Assert_INTx and Deassert_INTx messages if the new aggregated state is different from that previously reported to the root.

Adding a downstream switch port to a partition causes the partition's upstream switch port PCI-to-PCI bridge function to aggregate the interrupt state of the added downstream switch port. This may result in the generation of Assert_INTx and Deassert_INTx messages if the new aggregated state is different from that previously reported to the root.

Mode Change Effect on Port

A port mode change triggers the execution of the operation dictated by the OMA field value described in section No Action Mode Change Behavior on page 5-21. The following operations are common to all port mode changes.

TLP Buffering

All TLPs associated with the source partition (i.e., the partition from which the port was removed) are silently discarded from the port's ingress buffer. The port's egress buffer stops accepting stops accepting TLPs associated with the source partition from the switch core.

TLPs associated with the source partition that are already queued in the port's egress buffer are silently discarded if the value of the OMA field causes the link to go down. Otherwise, if the value of the OMA field does not cause the link to go down, then TLPs from the source partition queued in the port's egress buffer are transmitted on the link normally (i.e., there is no clear delineation on the link between TLPs associated with the source partition and those associated with the destination partition).

Regardless of the OMA field value, TLPs queued in the egress buffer that target function(s) associated with the port (i.e., target configuration or memory space of a function associated with the port) are silently discarded.

1. PME synchronization requires that an upstream port be present in the partition.

2. Bus locking requires that an upstream switch port be present in the partition.

Notes

PME Synchronization

Any PME synchronization state associated with the port is reset. If the port has completed PME synchronization, then the LTSSM transitions to the Detect state and then to the LTSSM state, if any, specified by the OMA field value.

Bus Locking

Any bus locking state associated with the port is reset.

Port State or Context

Any port state or context associated with the previous mode of operation (e.g., downstream switch port) that is not relevant to the new mode of operation (e.g., upstream switch port) is cleared. All other state is preserved.

No Action Mode Change Behavior

Modifying the operating mode of a port when the OMA field is set to no action has the following behavior in addition to that specified by the common operating mode change behavior. The state of all registers associated with the port are preserved (i.e., they are not reset). Status fields may change as appropriate. For example, if the port is in $D3_{Hot}$ at the time the operating mode change occurs, it continues operating in this state after the operating mode change completes.

If the link is up, it remains up in operating in the same LTSSM mode (i.e., upstream or downstream).If the link is down and the new operating mode is an operational mode, then the link begins link training from the Detect state in the specified LTSSM mode (i.e., upstream or downstream).

Reset Mode Change Behavior

Modifying the operating mode of a port when the OMA field is set to port reset has the following behavior in addition to that specified by the common operating mode change behavior:

- All states associated with the port are reset.
- Registers associated with the port are reset to their initial value except those designated SWSticky. SWSticky registers and field contents are preserved. For example, if the port is in $D3_{Hot}$ at the time the operating mode change occurs, it changes to the D0 state after the operating mode change completes.
- The port reset output is asserted (i.e., PxRSTN).
- The port remains in a reset state for at least 250 µs.
- The port reset output remains asserted while in the reset state. During this time, the LTSSM remains in the Detect state (i.e., the link enters the DL_Down state).
- The port reset output is negated (i.e., PxRSTN).
- Following an exit from the reset state, if the new operating mode is an operational mode, then the link begins link training from the Detect state in the specified LTSSM mode (i.e., upstream or downstream).

Partition Reconfiguration and Failover

Partition reconfiguration may be initiated statically through a fundamental reset or dynamically while the system is running. Static reconfiguration requires a switch fundamental reset and is nothing more than configuration performed either through modification of the sampled boot configuration vector or initialization performed via serial EEPROM or SMBus (see Chapter 12). Thus, this section focuses on dynamic reconfiguration.

Partition reconfiguration is the modification of the operating mode of one or more switch ports associated with a partition. Possible partition reconfigurations are list below.

- A downstream switch port is added or removed from a partition
- An upstream port is added or removed from a partition
- The operating mode of the upstream port is modified.

Notes

Partition reconfiguration may be initiated by software through modification of the operating mode of a port, or initiated automatically as the result of a failover. When the Failover Enable (FEN) bit is set in the Switch Port Control (SWPORTxCTL) register, automatic failover reconfiguration is enabled. A failover is initiated when the failover capability selected by the Failover Capability Select (FCAPSEL) field in the SWPORTxCTL register triggers a failover. See Chapter 6 for more information on the failover capability structure.

A failover capability structure may initiate a primary failover or a secondary failover. A primary failover event signaled by the capability structure selected by the FCAPSEL field has the following effect on a port.

- The new port operating mode specified by the PFMODE, PFSWPART and PFDEVNUM fields in the SWPORTxFCTL register are copied into the corresponding fields in the SWPORTxCTL register.
- If these new values result in a port operating mode change, then the OMCI bit in the SWPORxTSTS register is set and the OMCC bit is set when the mode change completes.
- The Primary Failover Status (PFAILOVER) bit is set in the SWPORTxSTS register.

A secondary failover event signaled by the capability structure selected by the FCAPSEL field has the following effect on a port.

- The new port operating mode specified by the SFMODE, SFSWPART and SFDEVNUM fields in the SWPORTxFCTL register are copied into the corresponding fields in the SWPORTxCTL register.
- If these new values result in a port operating mode change, then the OMCI bit in the SWPORTxSTS register is set and the OMCC bit is set when the mode change completes.
  - The Secondary Failover Status (SFAILOVER) bit is set in the SWPORTxSTS register.

A failover event may also trigger a modification in the state of a partition. When the Failover Enable (FEN) bit is set in the Switch Partition Failover Control (SWPARTxCTL) register, automatic failover reconfiguration is enabled. A failover is initiated when the failover capability selected by the Failover Capability Select (FCAPSEL) field in the SWPARTxCTL triggers a failover.

A primary failover event signaled by the capability structure selected by the FCAPSEL field has the following effect on the partition.

- The Primary Failover State (PFSTATE) field in the Switch Partition Failover Control (SWPARTxFCTL) register is copied into the STATE field of the SWPARTxCTL register. If this results in a value change in the contents of the STATE field, then the SCI bit in the SWPARTxSTS register is set and the SCC bit is set when the state change completes.
- The Primary Failover Status (PFAILOVER) bit is set in the SWPARTxSTS register.

A secondary failover event signaled by the capability structure selected by the FCAPSEL field has the following effect on the partition.

- The Secondary Failover State (SFSTATE) field in the Switch Partition Failover Control (SWPARTxFCTL) register is copied into the STATE field of the SWPARTxCTL register. If this results in a value change in the contents of the STATE field, then the SCI bit in the SWPARTxSTS register is set and the SCC bit is set when the state change completes.
- The Secondary Failover Status (SFAILOVER) bit is set in the SWPARTxSTS register.

A failover event that causes both partition reconfiguration as well as partition state change has the following execution order.

1. Port reconfiguration is executed first.
2. Once all port reconfiguration has completed, the partition state change is performed.

Notes

Partition Reconfiguration Latency

The amount of time that switch takes to do a partition reconfiguration depends on the reconfiguration actions. A partition reconfiguration action may involve partition state changes and/or port operating mode changes.

- The latency to perform port operating mode changes is described in section Port Operating Mode Change Latency on page 5-14.This latency defaults to 2 ms or 2.5 ms, depending on the port's operating mode change action (OMA).
- The latency to perform partition state changes is described in section Partition State Change Latency on page 5-4. This latency is typically a few cycles, unless the partition state change has the side-effect of causing port operating mode changes. For the latter case, the delay is as described in the previous bullet.

System Notification of Partition Reconfiguration

A system may require software notification when a partition reconfiguration occurs. The PES32NT8xG2 contains multiple mechanisms to accomplish this.

- If the reconfiguration is the result of a failover event, the failover may be signaled by the device via the following mechanisms:
  - Generation of an interrupt by the upstream port of the reconfigured partition. The interrupt may be generated by the upstream port's PCI-to-PCI bridge function or the NT function. Refer to section Interrupts on page 10-4 and section Interrupts on page 14-20 respectively for details.
  - Through the Event Signaling mechanism described in section Switch Events on page 16-1.

Alternatively, if the reconfiguration results in the addition, removal, or change in operating mode of the upstream port associated with the partition, then the system may be notified of the reconfiguration by a link down or link up event detected by the component upstream of the partition (i.e., the root or switch downstream port). This form of notification requires that the OMA field in the SWPORTxCTL register be set to reset.

Finally, if the reconfiguration is done by a switch management agent, the agent can coordinate and notify the reconfiguration using the global signals event mechanism described in section Global Signals on page 16-4.

Notes

Failover

Notes

Overview

The PES32NT8xG2 supports a flexible failover mechanism that allows the construction of highly-available systems. The failover mechanism can be used to automatically reconfigure switch partitions (as described in section Partition Reconfiguration and Failover on page 5-21) upon detection of a pre-defined trigger.As shown in Figure 6.1, there is a clear distinction in the switch between the policy used to trigger a failover and the reconfiguration. A failover capability implements the policy for determining when to trigger a failover. The triggering of a failover is broadcast to all switch partitions and ports. Switch partitions and ports sensitive to the failover, perform failover reconfiguration.

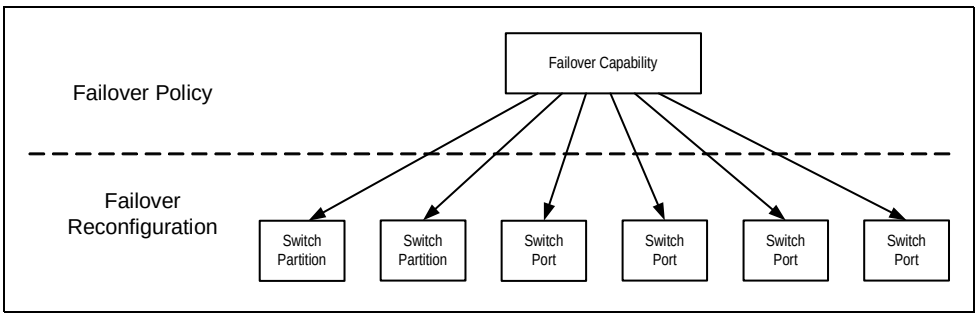

Figure 6.1 Failover Policy vs. Failover Reconfiguration

The PES32NT8xG2 implements four failover capabilities. A failover capability may signal a primary or secondary failover. A port is sensitive to a failover capability when the Failover Enable (FEN) bit is set and the Failover Capability Select (FCAPSEL) field selects the failover capability. Both fields are in the Switch Port x Control (SWPORTxCTL) register. A partition is sensitive to a failover capability when the Failover Enable (FEN) bit is set and the Failover Capability Select (FCAPSEL) field selects the failover capability. Both fields are in the Switch Partition x Control (SWPARTxCTL) register.

Examples of failover reconfiguration are provided below. See section Partition Reconfiguration and Failover on page 5-21 for details on port and partition failover reconfiguration.

- A downstream switch port is added or removed from a partition
- An upstream port is added or removed from a partition
- The operating mode of the upstream port is modified.

Failover Initiation

Each of the four PES32NT8xG2 failover capabilities has associated with it three registers.

- Failover Capability x Control (FCAPxCTL)
- Failover Capability x Status (FCAPxSTS)
- Failover Capability x Watchdog Timer (FCAPxTIMER)

A failover capability sets the policy for determining when a failover is triggered. There are three possible policies.

- Software initiated failover
- Signal initiated failover
- Watchdog timer initiated failover.

Notes

The following sections describe each of these policies. In most systems it is expected that a failover capability will only use one policy at a time. While enabling multiple policies in a single failover capability is not prohibited, care must be exercised to ensure that only one failover occurs at a time. If a second failover is triggered while an earlier failover is in progress, then the behavior is undefined.

Software Initiated Failover

A software initiated failover is initiated by writing a one to the Failover Software Trigger (FSWTRIG) register in the Failover Capability Control (FCAPxCTL) register. A software initiated failover may be instituted by software running on a root, endpoint, or on an SMBus master. When a failover is triggered, the type of failover is determined by the state of the Failover Mode (FMODE) field in the corresponding Failover Capability Status (FCAPxSTS) register.

- If the current failover mode is primary, then a secondary failover is triggered.
- If the current failover mode is secondary, then a primary failover is triggered.

Signal Initiated Failover

A failover may be triggered by a change in the state of a package signal pin.

- Failover trigger input 0 (FAILOVER0) is associated with failover capability 0.
- Failover trigger input 1 (FAILOVER1) is associated with failover capability 1.
- Failover trigger input 2 (FAILOVER2) is associated with failover capability 2.
- Failover trigger input 3 (FAILOVER3) is associated with failover capability 3..

Failover trigger inputs are GPIO alternate functions. See Chapter 13 for details.

Signal initiated failover is enabled when the Failover Signal Trigger Enable (FSIGEN) bit in the FCAPxCTL register is set. When the FSIGEN bit is cleared, the state of the corresponding FAILOVERx signal has no effect on the operation of the device.

The Failover Signal Polarity (FSIGPOL) bit in the FCAPxCTL determines how state changes in the FAILOVERx signal are interpreted.

- A secondary failover may be initiated on a low-to-high or high-to-low transition.
- A primary failover may be initiated on a high-to-low or low-to-high transition.

The state of the FAILOVERx signal should not be changed at the same time the signal's polarity is configured using the FSIGPOL bit. If the state of the FAILOVERx signal changes at the same time that the FSIGPOL bit is modified, the failover may or may not be triggered, depending on the timing of the events, the state of the FAILOVERx signal, and the polarity setting in the FSIGPOL bit.

Note: To avoid this situation, it is recommended that the FSIGPOL bit be configured prior to enabling the GPIO alternate function associated with the FAILOVERx signal.

The state of the FAILOVERx signal should not be modified more frequently than once per second. Modifying the state of the FAILOVERx signal more frequently than once per second produces undefined results.

Watchdog Timer Initiated Failover

A failover may be triggered as the result of an expiration of a watchdog timer. Such a failover is initiated when the Failover Timer Trigger Enable (FTIMEN) bit is set in the FCAPxCTL register is set and the Count (COUNT) field in the Failover Capability Watchdog Timer (FCAPxTIMER) register transitions from one to zero.

When non-zero, the COUNT field in the FCAPxTIMER register is decremented once per microsecond (1 uS). This provides a maximum watchdog timer internal of over one hour. Decrementing of the count field ceases when zero is reached. The COUNT field may be written by software at any time. Modifying the COUNT field is used to rearm the watchdog timer. If not expired, the watchdog timer continues to decrement across all resets except a switch fundamental reset.

Notes

When a failover is triggered, the type of failover is determined by the state of the Failover Mode (FMODE) field in the corresponding Failover Capability Status (FCAPxSTS) register.

- If the current failover mode is primary, then a secondary failover is triggered.
- If the current failover mode is secondary, then a primary failover is triggered.

Notes

Link Operation

Notes

Overview

Link operation in the PES32NT8xG2 switch adheres to the PCI Express Base Specification Revision 2.1, supporting speeds of 2.5 GT/s and 5.0 GT/s. This chapter does not describe the controls related to the Serializer-Deserializer (SerDes) block associated with each port. Refer to Chapter 8, SerDes, for a detailed description of this topic.

Each port's link operates independently from any other port. This chapter describes link operation from the perspective of each port.

The behavior of the port's link varies depending on whether the port is an upstream or downstream port. In the PES32NT8xG2, each port supports upstream and downstream link behavior. The behavior is determined dynamically by the port's operating mode (e.g., upstream switch port, downstream switch port, etc). Refer to Chapter 5, Switch Partition and Port Configuration, for further details on port operating modes.

Depending on the port operating mode, a port may be configured with a single function or multiple functions (i.e., PCI-to-PCI bridge, NT, and DMA). Multi-function ports follow the rules for multi-function devices outlined in the PCI Express Base Specification 2.1.

In this specification, the term full link retrain is defined as retraining a link by transitioning through the PHY's LTSSM[1] Detect state.

Port Merging

Ports in the PES32NT8xG2 may be merged to increase the port's maximum link width. Refer to section Stack Configuration on page 3-5 for a description of the possible port configurations. As described in this section, the switch allows two x4 ports to be merged into a single x8 port. The merging of ports is controlled by the stack configuration registers, as described in section Stack Configuration on page 3-5.

When ports are merged, the corresponding serial link pins (i.e., PExRP[], PExRN[], PExTP[], and PExTN[]) associated with each of the merged ports form the merged link. Specifically, the serial link pins associated with the lowest-numbered merged port correspond to the lowest numbered lanes of the merged port's link. The serial link pins associated with the highest-numbered merged port correspond to the highest numbered lanes of the merged port's link.

Note: Pin [x] of a port refers to a lane. For port 0, PE00RN[0] refers to lane 0, PE00RN[1] refers to lane 1, etc.

For example, port 8 is associated with the PE08RP[3:0] serial pins. Similarly, port 12 is associated with PE012RP[3:0] serial pins. When these ports are merged into a x8 port, port 12 is de-activated[2] and port 8 uses the pins associated with port 12's link (i.e., PE12RP[0] is associated with port 8 lane 4, and PE12RP[1] is associated with port 8 lane 5, etc.; PE08RP[0] remains associated with port 8, lane 0). The same applies to the other pins associated with the serial link (i.e., PExRN[], PExTP[], and PExTN[]) signals. Also, the MAXLNKWDTH field in port 8's PCIELCAP register is automatically set by the hardware to 0x8 (i.e., maximum link width is x8).

1. The term 'LTSSM' refers to a port's Link Training and Status State Machine in the Physical Layer.

2. Refer to section Stack Configuration on page 3-5 for a formal definition of the behavior of a deactivated port.

Notes

Port Maximum Link Width

The Maximum Link Width (MAXLNKWDTH) field in a port's PCI Express Link Capabilities (PCIELCAP) register indicates the maximum link width of the port based on the stack configuration at the time. Therefore, when a stack is configured such that ports are merged, the MAXLNKWDTH field is automatically set by the hardware to correctly indicate a merged port's maximum link width.

Polarity Inversion

Each port of this switch supports automatic polarity inversion as required by the PCI Express Base Specification. Polarity inversion is a function of the receiver and not the transmitter. The transmitter never inverts its data.

During link training, the receiver examines symbols 6 through 15 of the TS1 and TS2 ordered sets for inversion of the PExRP[n] and PExRN[n] signals. If an inversion is detected, then logic for the receiving lane automatically inverts received data. Polarity inversion is a lane function, not a link function. Therefore, it is possible for some lanes of link to be inverted and for others to not be inverted.

Lane Reversal

The PCI Express Base Specification describes an optional lane reversal feature. This switch supports the automatic lane reversal feature outlined in the specification. The operation of lane reversal is dependent on the *highest achievable link width* determined dynamically by the PHY. The highest achievable link width is the minimum of:

- The value of the MAXLNKWDTH field in the port's PCI Express Link Capabilities (PCIELCAP) register.
- The number of consecutive lanes detected by the LTSSM during the Detect state on which valid training sets are received.

Note that the highest achievable link width is not necessarily the same as the port's maximum link width advertised in the PCIELCAP register. Also, note that the actual width to which the link trains may not match the highest achievable link width (i.e, link training may fail on some lanes).

Lane reversal mapping for the various non-trivial maximum link width configurations supported by the switch are illustrated in Figures 7.1 through 7.3. In the figures, PExRP[n] refers to the PES32NT8xG2 pin associated with lane n of port 'x'.

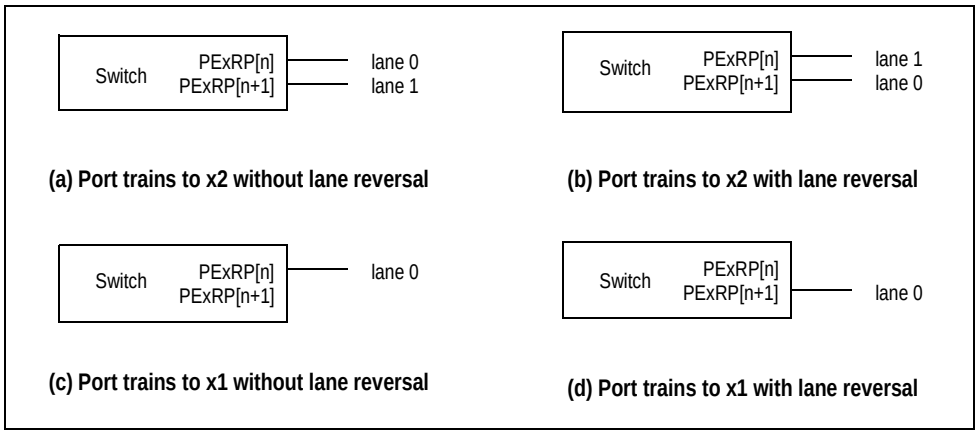

Figure 7.1 Lane Reversal for Highest Achievable Link Width of x2

Notes

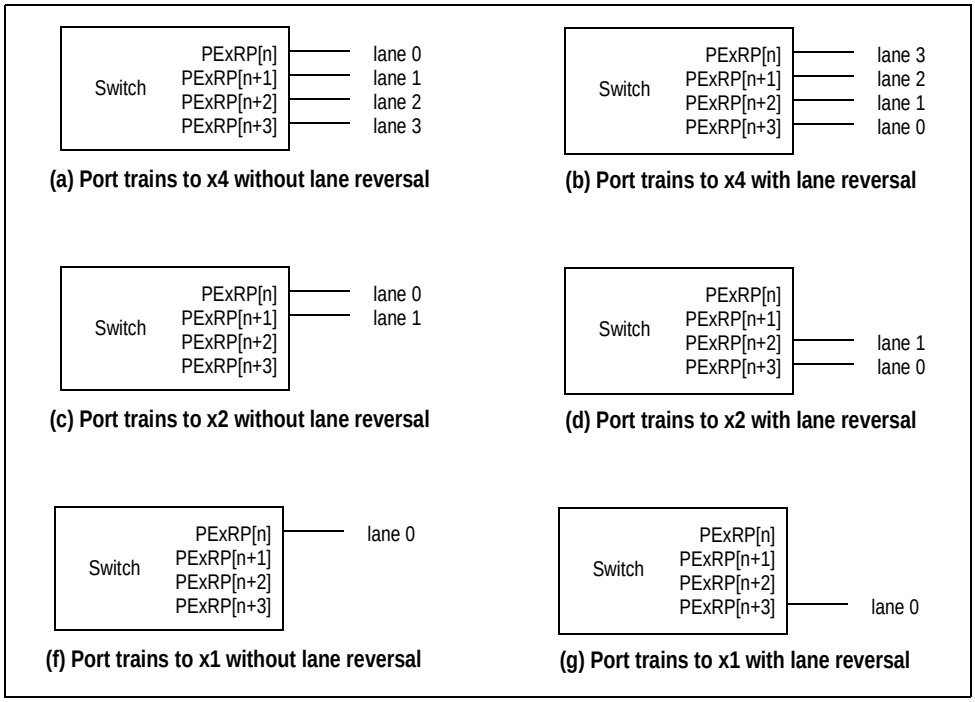

Figure 7.2 Lane Reversal for Highest Achievable Link Width of x4

Notes

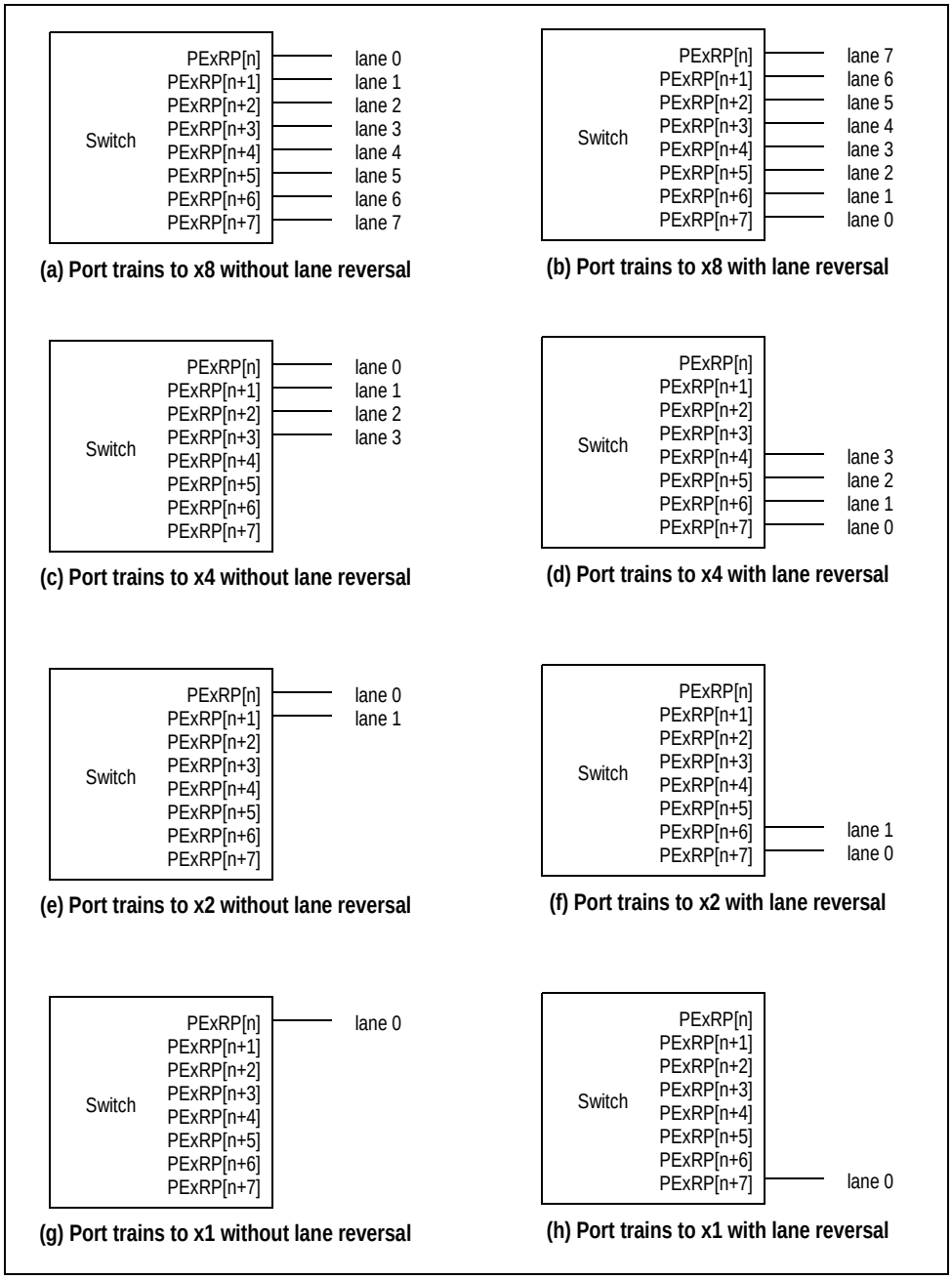

Figure 7.3 Lane Reversal for Highest Achievable Link Width of x8

Link Width Negotiation

PES32NT8xG2 ports support the optional link variable width negotiation feature outlined in the PCI Express Base Specification. The Maximum Link Width (MAXLNKWDTH) field in a port's PCI Express Link Capabilities (PCIELCAP) register contains the maximum link width that the port can achieve. This field is of type and may be modified (e.g., by firmware or software) when the REGUNLOCK bit is set in the SWCTL register. Modification of this field allows the maximum link width of the port to be configured. The new link width takes effect the next time full link training occurs.[1]

1. When re-programming the MAXLNKWDTH field in a port that operates in a multi-function mode, the user must ensure that all functions of the port have identical settings for maximum link width.

Notes

The actual link width is determined dynamically during link training. Ports limited to a maximum link width of x8 are capable of negotiating to a x8, x4, x2, or x1 link width. The actual negotiated width of a link may be determined from the Negotiated Link Width (NLW) field in the corresponding port's PCI Express Link Status (PCIELSTS) register. To force a link width to a smaller width than the default value, the MAXLNKWDTH field could be configured through Serial EEPROM initialization and full link retraining forced by setting the Full Link Retrain (FLRET) bit in the port's PHYLSTATE0 register.

Link Width Negotiation in the Presence of Bad Lanes

In an effort to maximize the link width when one or more lanes of a multi-lane link are not functioning correctly (i.e., reliable communication of training sets across the lane is not possible), the PES32NT8xG2 downstream switch ports automatically attempt a lane reversed configuration when doing so has the potential to enhance the achievable link width.

For example, if lane 1 of a x4 link is not operating correctly, the device's downstream switch port attached to the link attempts a lane reversed configuration to form a x2 link using lanes 2 and 3 (Figure 7.2 (d)). If the link partner accepts the lane reversed configuration, the optimal x2 link will be formed using lanes 2 and 3. If the link partner does not accept the lane reversed configuration, but instead requests a lane configuration supported by the device (e.g., x1 link using lane 0), the PES32NT8xG2 accepts the configuration and forms the reduced-width link. Otherwise, if the lane numbering agreement fails, the device automatically re-trains the link from the Detect state. During this re-training, the device's port does not re-attempt a lane reversed configuration, but rather tries to form the link without reversing the lanes. As a result, a x1 link is formed using lane 0 (Figure 7.2 (f)).

Dynamic Link Width Reconfiguration

The PCI Express Base Specification includes support for dynamic upconfiguration of link widths. This optional capability allows both components of a link to dynamically downconfigure and upconfigure links based on implementation specific criteria such as power savings, link bandwidth requirements, or link reliability problems.

As an example, a link that initially does a full link train to x4 may be dynamically downconfigured to x1 in order to save power when there is little traffic on the link. As traffic increases, the link may be dynamically upconfigured to its initial link width of x4. Also, the link width may be downconfigured if a particular lane is determined to be unreliable (e.g., the bit error rate in the lane is above a user-defined threshold).

With dynamic link width reconfiguration, the system designer can choose to connect components with enough lanes to handle worst case bandwidth requirements, yet not waste power when the link is not fully utilized. This capability offers an additional mechanism for link power reduction on top of the ASPM link states (L0s, L1, etc.)

Dynamic upconfiguration and downconfiguration is done on a per-link basis, and does not result in the link going into a link-down state. A link can be upconfigured up to the negotiated link width set after a full link training. For example, a link that trained to a width of x2 after a full link training cannot be upconfigured to a width above x2.

A link can be downconfigured down to x1. When a link is downconfigured to a smaller width, inactive lanes are kept in Electrical Idle with their receiver terminations enabled. These lanes continue to be associated with the downconfigured port's LTSSM.

In order for upconfiguration to occur successfully, both of the link components must support it. Furthermore, the PCI Express Base Specification recommends that a link component not initiate downconfiguration unless the link partner supports link upconfiguration, except for link reliability reasons.

The capability to upconfigure a link is transmitted among components using the in-band TS2 ordered set. When downconfiguration or upconfiguration of a link occurs, one of the components on the link initiates the process, while the other component responds to the process. The PCI Express Base Specification 2.1 indicates that both of these capabilities are optional.

Notes

Software may be notified of link width reconfiguration via the link bandwidth notification mechanism described in the PCI Express Base Specification. This mechanism is enabled by setting the Link Bandwidth Management Interrupt Enable (LBWINTEN) bit in the PCIELCTL register of switch downstream switch ports.

Dynamic Link Width Reconfiguration in the PES32NT8xG2

PES32NT8xG2 ports support dynamic link width upconfiguration and downconfiguration in response to link partner requests. This capability is honored for regular links and crosslinks.

The switch's ports do not initiate autonomous link width upconfiguration and downconfiguration of links, except for downconfiguration due to link reliability reasons. Therefore, the Hardware Autonomous Width Disable (HAWD) bit in the port's PCIELCTL register has no effect and is hardwired to 0x0. Additionally, the switch's ports never set the 'Autonomous Change' bit in the training sets exchanged with the link partner during link training.[1]

A downstream switch port's link partner may autonomously change link width. When this occurs, the PES32NT8xG2 downstream switch port sets the Link Autonomous Bandwidth Status (LABWSTS) bit in the PCIELSTS register.

Link Speed Negotiation

The PCI Express Base Specification introduces support for 5.0 GT/s data rate (i.e., Gen 2), in addition to the 2.5 GT/s data rates (i.e., Gen 1) mandated in previous versions of the specification. Per this specification, all lanes of a link must operate at the same data rate. During full link training (i.e., from the Detect state), links initially operate at 2.5 GT/s. Once the LTSSM on both components of the link reach the L0 state and the data-link layer enters the DL_Active state, the link speed may be upgraded to 5.0 GT/s if this capability is advertised by both components. The process of upgrading the link speed does not result in a link_down state.

A component advertises its supported speeds via the Data Rate Identifier bits in the TS1 and TS2 training sets transmitted to its link partner during link training. The PCI Express Base Specification permits a component to change its supported speeds dynamically. It is allowed for a component to advertise supported link speeds without necessarily changing the link speed, via the Recovery LTSSM state.

A component determines the supported speeds of its link partner by examining the Data Rate Identifier bits in the TS1/TS2 training sets received during link training, specifically in the Configuration.Complete and Recovery.RcvrCfg states. The last advertisement received overrides any previously recorded value.

Either link component may request a link speed change due to software requests or link reliability reasons (i.e., speed downgrade). Downstream components are further permitted to request link speed changes due to autonomous hardware initiated mechanisms. A component must only initiate a link speed change when it knows that its link partner supports the target speed via prior exchange of Training Sets. Gen 2 support is optional while Gen 1 support is mandatory.

If neither component in the link advertises support for Gen 2, then the link remains operating in Gen 1 speed. If one component has advertised support for Gen 1 and Gen 2, and the other has advertised support for Gen 1 only, then the link will remain operating in Gen 1 speed until the lesser speed component decides to:

- Advertise support for Gen 2 via the Recovery state without modifying the link speed. The link remains operating at Gen 1 speed.
- Transition the link speed to Gen 2 via the Recovery.Speed state. The link will operate at Gen 2 speed. In this case, the advertisement of Gen 2 speed by both components is done implicitly in the Recovery substates entered while modifying the link speed.

1. Note that the 'Autonomous Change' bit is located in bit 6 of the fourth symbol in the training sets. This bit has multiple meanings depending on the LTSSM state in which it is issued. The switch never sets this bit in LTSSM states in which this bit carries the 'autonomous change' meaning.

Notes

It is the responsibility of the upstream component of the link (i.e., switch downstream switch ports) to keep the link at the target link speed or at the highest common speed supported by both components of the link, whichever is lower. In addition, the upstream component must initiate a link speed upgrade if it has recorded support for the higher speed by its link partner and software sets the Link Retrain bit in the PCIELCTL register with a target link speed which is not equal to the current link speed.

The upstream component (i.e., switch downstream switch port) is capable of notifying software of link speed changes via the Link Bandwidth Notification mechanism described in the PCI Express Base Specification.

Link Speed Negotiation in the PES32NT8xG2

PES32NT8xG2 ports support data rates of 5.0 GT/s and 2.5 GT/s. The highest data rate of each link is determined dynamically, and depends on the following factors:

- Maximum link data rate supported by both components of the link.
- The Target Link Speed set via the Link Control 2 Register (PCIELCTL2) in function 0 of the port.
- The reliability of the link at 5.0 GT/s.

By default, the Target Link Speed (TLS) of each port is set to 5.0 GT/s. Therefore, the PES32NT8xG2 ports advertise support for 2.5 GT/s and 5.0 GT/s during the link training process via training-sets. During normal operation, the TLS field should not be modified in an upstream port.

After a fundamental reset, each port link trains to the L0 state at 2.5 GT/s (Gen 1). Once the data-link layer reaches the DL_Active state, if the Target Link Speed indicates 5.0 GT/s (default value), the PES32NT8xG2 downstream switch ports automatically initiate link speed upgrade to 5.0 GT/s (Gen 2) using the link speed change mechanism described in the PCI Express Base Specification. Upstream ports do not automatically initiate link speed upgrade to Gen 2.

- The Initial Link Speed Change Control (ILSCC) bit in a port's PHYLCFG0 register controls whether the port automatically initiates a speed upgrade to Gen 2. If the ILSCC bit is set, the port does not automatically initiate a speed change to Gen 2. Software may modify this bit to change the default behavior.
- The Link Bandwidth Management Status (LBWSTS) bit in the PCIELSTS register of downstream switch ports is not set since the initial link speed upgrade is not caused by a software directed link retrain or due to link reliability issues.

The current link speed of each port is reported via the Current Link Speed (CLS) field of the port's Link Status Register (PCIELSTS). If the port operates in a multi-function mode, all functions of the port report the same value in this field.

The above behavior also applies after full link retrain (i.e., when the LTSSM transitions through the 'Detect' state).

Assuming the target link speed is set to 5.0 GT/s, a switch port initiates a link speed upgrade in the following cases:

- Link speed upgrade after initial link train (i.e., from the Detect state) to L0 at 2.5 GT/s, when the link partner advertised support for the higher speed.
- Link speed upgrade after full link retrain (i.e., via the Detect state) to L0 at 2.5 GT/s, when the link partner advertised support for the higher speed.
- For downstream ports, when the link operates at 2.5 GT/s data rate and the link partner advertises support for the higher speed via the Recovery states. PES32NT8xG2 downstream ports automatically initiate a link speed upgrade to 5.0 GT/s to keep the link operating at the highest speed as required by the PCI Express Base Specification.
- When software sets the Link Retrain (LRET) bit in the port's PCIELCTL register and the port has recorded support for the higher speed by its link partner.[1]

[1] The speed advertisement of the link partner is noted by the switch in the latest LTSSM entry to the Configuration.Complete or Recovery.RcvrCfg sub-states.

Notes

When operating at 5.0 GT/s, a PES32NT8xG2 port initiates a link speed downgrade in the following cases:

- When the PHY layer cannot achieve reliable operation at the higher speed. In this case, the PES32NT8xG2 port continues to support the higher speed in the training-sets it transmits during link training.
- When software sets the target link speed to 2.5 GT/s and sets the LRET bit in the port's PCIELCTL register. In this case, the PES32NT8xG2 port removes support for the higher speed in the training-sets it transmits during link training.

Additionally, the PES32NT8xG2 ports always respond to link partner requests to change speed. In this case, the speed change is only successful when both components in the link advertise support the target speed. When a link speed upgrade operation fails, the PHY LTSSM reverts back to the speed before the upgrade (i.e., 2.5 GT/s) and does not autonomously initiate a subsequent link speed upgrade. In this case, the PHY continues to support Gen 1 and Gen 2 data rates and therefore responds to link partner requests for link speed upgrade, or to link speed upgrades triggered by software setting the LRET bit in the port's PCIELCTL register.

PES32NT8xG2 ports do not have a mechanism to autonomously regulate link speed. As a result, the Hardware Autonomous Speed Disable (HASD) bit in the PCIELCTL2 register has no effect and is hardwired to 0x0. Additionally, the PES32NT8xG2 ports never set the 'Autonomous Change' bit in the training sets exchanged with the link partner during link training[1]. Still, a link partner connected to a PES32NT8xG2 downstream switch port may autonomously change link speed. When this occurs, the PES32NT8xG2 downstream switch port sets the Link Autonomous Bandwidth Status (LABWSTS) bit in the PCIELSTS register.

A system designer may limit the maximum speed at which each port operates by changing the target link speed via software or EEPROM and forcing link retraining. Refer to section Link Retraining on page 7-9 for further details.

Software Management of Link Speed

Software can interact with the link control and status registers of downstream switch ports to set the link speed, as well as receive notification of link speed changes. This gives software the capability to choose the desired link speed based on system specific criteria. For example, depending on the traffic load expected on a link, software can choose to downgrade link speed to 2.5 GT/s in order to reduce power on a low-traffic link, and later upgrade the link to 5.0 GT/s when the bandwidth is required. Software may also choose to change the link speed due to link reliability reasons (i.e., a link that has reliability problems at 5.0 GT/s may be downgraded to 2.5 GT/s).

As mentioned above, the Target Link Speed (TLS) field of the Link Control 2 Register (PCIELCTL2) in function 0 of the port sets the preferred link speed. By default, the Target Link Speed of each switch port is set to 5.0 GT/s.

During normal operation, the link speed of a downstream switch port may be modified by setting the TLS field in the PCIELCTL2 register to the desired speed and initiating link retraining by writing a one to the Link Retrain (LRET) bit in the Link Control (PCIELCTL) register.

- The port will only initiate a change to a higher speed if the link partner advertised support for the higher speed in its latest entry to the Configuration.Complete or Recovery.RcvrCfg states.
- If a speed change is initiated to a speed not supported by the link partner, then the port will remain at the current speed by transitioning through the Recovery state without the "Speed_Change" bit set.

Notification of link speed changes if provided through the link bandwidth notification mechanism described in the PCI Express Base Specification. This mechanism is enabled by setting the Link Bandwidth Management Interrupt Enable (LBWINTEN) bit in the PCIELCTL register of switch downstream switch ports.

1. Note that the 'Autonomous Change' bit is located in bit 6 of the fourth symbol in the training sets. This bit has multiple meanings depending on the LTSSM state in which it is issued. The switch never sets this bit in LTSSM states in which this bit carries the 'autonomous change' meaning.

Notes

For downstream switch ports, the Link Bandwidth Management Status (LBWSTS) bit in the PCIELSTS register is set when the link speed is changed due to the following reasons:

- Link speed downgrade initiated by a switch port when the PHY layer cannot achieve reliable operation at the higher speed. Note that this does not include link speed downgrading due to failure to achieve symbol lock while trying to upgrade link speed via the Recovery state.
- Link speed change initiated by the link partner that was not indicated as an autonomous change.

Also, the LBWSTS bit is set whenever software sets the LRET bit in the PCIELCTL register, even if the link speed is not changed. Note that the LBWSTS bit is not set during the initial link speed change (i.e., the speed change from Gen 1 to Gen 2 after fundamental reset or a full-link-retrain via the 'Detect' state).

Software can verify the link speed by reading the Current Link Speed (CLS) field of the port's PCI Express Link Status Register (PCIELSTS). If the port operates in a multi-function mode, all functions of the port report the same value in this field.

Link Retraining

Per the PCI Express Base Specification, link retraining can be done autonomously in response to link problems (i.e., repeated TLP replay attempts), or as a result of software setting the link retrain (LRET) bit in the PCI Express Link Control (PCIELCTL) register. Writing a one to the Link Retrain (LRET) bit in a downstream switch port's PCI Express Link Control (PCIELCTL) register forces the downstream link to retrain. **When this occurs the LTSSM transitions to the Recovery state.**

Link retraining does not result in the link going down, unless the LTSSM transitions through the Detect state in its retraining attempt. The speed of the link is not necessarily changed as a result of link retraining. A link that operates at 5.0 GT/s will continue to operate at that speed if the link retraining attempt is successful at that speed. Otherwise, the link speed is changed to 2.5 GT/s.

When link retraining results in the speed of the link being downgraded from 5.0 GT/s to 2.5 GT/s, the Link Bandwidth Management Status (LBWSTS) bit is set in the PCI Express Link Status (PCIELSTS) register. Additionally, the PHY LTSSM remains at the downgraded speed until the link partner requests a link speed upgrade, software sets the LRET bit in the PCIELCTL register, or the link is fully retained via the FLRET bit in the PHYLSTATE0 register.

In addition to link retrain (via the Recovery state), the link may be fully retrained by writing a one to the Full Link Retrain (FLRET) bit in a port's Phy Link State 0 (PHYLSTATE0) register. **When this occurs the LTSSM transitions directly to the Detect state.**

The PHYLSTATE0 register is located in the proprietary port-specific registers located in the PCI-to-PCI bridge function's configuration space (see section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11).

Full link retraining causes the data-link to go down (refer to section Link Down Handling on page 7-10).

- The sudden transition to the Detect state caused by a full link retrain may result in error status bits being set in status registers associated with the port's physical, data-link, and transaction layers.

Note that the LBWSTS bit in the PCIELSTS register is not affected by a full link retrain (i.e., since the data-link indicates a DL_Down status).

Link States

The PES32NT8xG2 ports support the following link states:

- L0
  - Fully operational link state
- L0s
  - Automatically entered low power state with shortest exit latency
- L1
  - Lower power state than L0s

Notes

- May be automatically entered (i.e., ASPM) or directed by software by placing the device in the $D3_{hot}$ state

– L2/L3 Ready
  - The L2/L3 state is entered after the acknowledgement of a Power Management Event Turn Off (PME_Turn_Off) Message.
  - There is no TLP or DLLP communications over a link in this state.
  - The L2/L3 Ready state can only be exited when the port is reset (i.e., as a result of a switch fundamental reset, a partition fundamental reset, or a port reset caused by an operating mode change on the port).
  - In the PES32NT8xG2, the link is considered link-down in the L2/L3 Ready state.

  This allows a power management turn-off event in a partition to be signaled via the event signaling mechanism (see Chapter 16, Switch Events).

  In an upstream port, a link-down due to L2/L3 Ready does not cause a hot reset on the port.

– L3
  - Link is completely unpowered and off

– Link-Down
  - A transitional link-down pseudo-state prior to L0. This pseudo-state is associated with the LTSSM Detect, Polling, Configuration (when coming from Detect → Polling → Configuration), Disabled, Loopback, and Hot Reset states. In these states, the data-link is in the DL_Inactive state and reports a DL_Down condition to the higher layers.

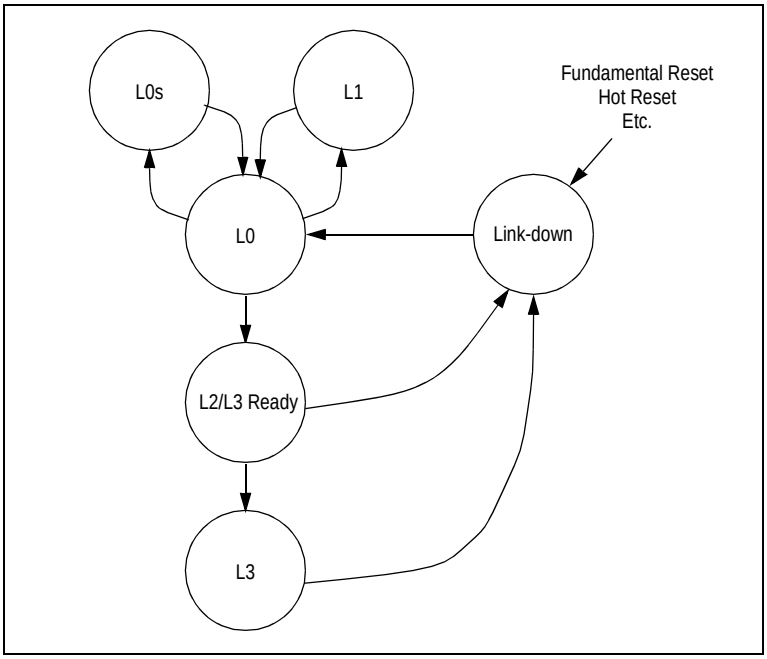

Figure 7.4 PES32NT8xG2 ASPM Link State Transitions

Link Down Handling

When an upstream port's data-link indicates a DL_Down status, it triggers a hot reset in the partition associated with the port, as described in section Partition Hot Reset on page 3-11. In addition:

– All TLPs queued in the port's ingress frame buffer (IFB) are silently discarded.

– All TLPs queued in the port's replay buffer (EFB) are silently discarded.

Notes

When a downstream switch port's data-link indicates a DL_Down status, the following occurs:

- All TLPs queued in the port's ingress frame buffer (IFB) are silently discarded.
- All TLPs queued in the port's replay buffer (EFB) are silently discarded.
- Request TLPs received by other ports and destined to the logical bus number associated with the link that is down are treated as unsupported requests (UR) by the downstream switch port whose link is down.
- All other TLPs received by the other ports and destined to the logical bus number associated with the link that is down are silently discarded.
- The downstream port handles all TLPs that target the port's function normally. It is possible to perform configuration read and write operations to the PCI-to-PCI bridge function associated with that downstream port.

When a link comes up, flow control credits for the configured size of the port's IFB queues are initialized. Following this initialization, the data-link enters the DL_Active state and reports a DL_Up condition to the upper layers. A DL_Down condition on a downstream switch port's link may cause the Surprise Down Error Status (SDOENERR) bit to be set in the port's AER Uncorrectable Error Status (AERUES) register. The conditions under which surprise down is reported are described in Section 3.2.1 of the PCI Express Base Specification.

Note that when a downstream port is directed by a higher layer to the hot reset state (e.g., upstream switch port link receives training sets with the hot reset bit set, upstream switch port reports DL_Down, upstream secondary bus reset, or downstream secondary bus reset), this is not considered a surprise down error.

In addition to the exception conditions listed in Section 3.2.1 of the PCI Express Base Specification, the SDOENERR bit in a port's AERUES register is not set in the following cases:

- The partition associated with the port is placed in Disabled mode (section Partition State on page 5-3).
- The port is placed in Disabled mode (section Switch Port Mode on page 5-5).
- The port's link is fully retrained (i.e., PHY transitions to the Detect state) as a result of a port operating mode change action (section Port Operating Mode Change on page 5-13).
- The port's link is fully retrained via the FLRET bit in the PHYLSTATE0 register.
- The port's clocking mode is modified (section Port Clocking Modes on page 2-3).

Slot Power Limit Support

The Set_Slot_Power_Limit message is used to convey a slot power limit value from a downstream switch port or root port to the upstream port of a connected device or switch.

Upstream Port

When a Set_Slot_Power_Limit message is received by an upstream port, then the fields in the message are written to the PCI Express Device Capabilities (PCIEDCAP) register of that port.[1]

- Byte 0, bits 7:0 of the message payload are written to the Captured Slot Power Limit Scale (CSPLS) field.
- Byte 1, bits 1:0 of the message payload are written to the Captured Slot Power Limit Value (CSPLV) field.

[1] If the port is operating in a multi-function mode, the Set_Slot_Power_Limit messages targets all functions of the port.

Notes

Downstream Switch Port

A Set_Slot_Power_Limit message is generated and transmitted by downstream switch ports when either of the following events occur:

- A configuration write is performed to the corresponding PCIESCAP register when the link associated with the downstream switch port is up.
- A link associated with the downstream switch port transitions from a non-operational state to an operational (i.e., DL_Down to DL_Up) state.

Link Active State Power Management (ASPM)

The operation of link Active State Power Management (ASPM) is orthogonal to device power management. Once ASPM is enabled, ASPM link state transitions are initiated by hardware without software involvement. For a port operating in a multi-function mode, each function of the port has independent ASPM settings. The switch follows the rules governing ASPM policy in multi-function devices, described in Section 5.4.1 of the PCI Express Base Specification.

The switch's ASPM supports the L0s (receiver and transmitter) and L1 states. ASPM is enabled via the ASPM field in the function's link control register (PCIELCTL). Enabled ASPM settings that are common for all functions of the port are enabled for the port as a whole.

In general, ASPM entry and exit conditions are based upon the port's desire to transmit TLPs on the link. For a port in multi-function mode (e.g., upstream switch port with NT function, NT with DMA function, etc.), TLP transfers among functions in the port do not affect the ASPM state of the port, since the TLP transfer is not destined to the multi-function port's link. The only exception to this case is TLPs emitted by the DMA function that map into the partition's multicast BAR aperture, as these TLPs are routed to the port function(s) that claim the TLP as well as the port's link (see section DMA Multicast on page 15-23).

L0s ASPM

L0s entry/exit operates independently for each direction of the link. On the receive side, the PES32NT8xG2 upstream and downstream switch ports always respond to L0s entry/exit requests from the link partner. On the transmit side, the L0s entry conditions must be met for 7µs before the hardware transitions the transmit link to the L0s state.

L0s Entry Conditions

The transmit side L0s entry conditions depend on the port's operational mode (see Chapter 5). A port configured in upstream switch port mode initiates L0s entry when all of the conditions listed below are met:

- L0s ASPM is enabled via the port's PCIELCTL register.
- The following conditions are met for the amount of time specified above:
  - The receive lanes of all of the switch downstream switch ports in the partition which are not in a low power state (i.e., D3) and whose link is not down are in the L0s state.
  - The port has no TLPs to transmit on the link (i.e., the port's EFB is empty) or there are no available flow control credits to transmit a TLP.
  - The port has no DLLPs pending for transmission.

A port configured in downstream switch port mode initiates L0s entry when all of the conditions listed below are met:

- L0s ASPM is enabled via the port's PCIELCTL register.
- The following conditions are met for the amount of time specified above:
  - The receive lanes of the switch partition's upstream port are in the L0s state.
  - The port has no TLPs to transmit on the link (i.e., the port's EFB is empty) or there are no available flow control credits to transmit a TLP.
  - The port has no DLLPs pending for transmission.

Notes

A port configured in NT function mode or NT with DMA function mode initiates L0s entry when all of the conditions listed below are met:

- L0s ASPM is enabled via the PCIELCTL register of all functions in the port.
- The following conditions are met for the amount of time specified above:
  - The port has no TLPs to transmit on the link (i.e., the port's EFB is empty) or there are no available flow control credits to transmit a TLP.
  - The port has no DLLPs pending for transmission.

A port configured in upstream switch port with NT function mode or upstream switch port with NT and DMA function mode initiates L0s entry when the L0s entry conditions are met for all of its functions (i.e., L0s entry conditions for the PCI-to-PCI bridge, NT, and DMA functions are met):

- L0s ASPM is enabled in the PCIELCTL register of all functions in the port.
- The following conditions are met for the amount of time specified above[1]:
  - The receive lanes of all of the downstream switch ports in the switch partition (i.e., associated with the PCI-to-PCI bridge function) which are not in a low power state (i.e., D3) and whose link is not down are in the L0s state.
  - The port has no TLPs to transmit on the link (i.e., the port's EFB is empty) or there are no available flow control credits to transmit a TLP.
  - The port has no DLLPs pending for transmission.

L0s Exit Conditions

The transmit side L0s exit conditions depend on the port's operational mode. A port configured in upstream switch port mode initiates exit from L0s when either of the conditions listed below is met:

- The port has a TLP or DLLP scheduled for transmission on the link.
- A downstream switch port in the switch partition has initiated exit from L0s.

A port configured in downstream switch port mode initiates exit from L0s when either of the conditions listed below is met:

- The port has a TLP or DLLP scheduled for transmission on the link.
- The upstream port in the switch partition has initiated exit from L0s.

A port configured in NT function mode or NT with DMA function mode initiates exit from L0s when the condition listed below is met:

- The port has a TLP or DLLP scheduled for transmission on the link.

Finally, a port configured in upstream switch port with NT function mode or upstream switch port with NT and DMA function mode initiates exit from L0s when either of the conditions listed below is met:

- The port has a TLP or DLLP scheduled for transmission on the link.
- A downstream switch port in the switch partition associated with the PCI-to-PCI bridge function has initiated exit from L0s.

L1 ASPM

L1 entry/exit is initiated by downstream link components (i.e., upstream ports) and affects both directions of the link. Upstream link components (i.e., downstream switch ports) accept or reject L1 entry requests from the link partner.

L1 Entry Conditions

The PES32NT8xG2 downstream switch ports may accept or reject L1 entry requests sent by the link partner. A port configured in downstream switch port mode accepts L1 entry requests when all of the conditions listed below are met. Otherwise, the L1 entry request is rejected.

- L1 ASPM is enabled via the port's PCIELCTL register.
- The port has no TLPs pending for transmission on the link (i.e., the port's EFB is empty).
- The port has no ACK or NAK DLLPs pending for transmission.

[1] Note that there is a single L0ET timer for both functions of the port.

Notes

The PES32NT8xG2 upstream ports request entry into L1 based on the criteria defined below. The L1 entry conditions must be met for 1 ms before the upstream port transitions the link to the L1 state. If these conditions are met and the link is in the L0 or L0s states, then the hardware will request a transition to the L1 state from its link partner. If the link partner acknowledges the transition, then the L1 state is entered. Otherwise, L0s entry is attempted[1].

The upstream port's L1 entry conditions depend on the port's operational mode, as follows.

A port configured in upstream switch port mode initiates L1 entry when all of the conditions listed below are met:

- L1 ASPM is enabled via the port's PCIELCTL register.
- All of the downstream switch ports in the partition which are not in a low power state (i.e., D3) and whose link is not down are in the L1 state.
- The port has no TLPs pending for transmission on the link. The port's EFB is empty.
- The port has no DLLPs pending for transmission.
- The port's receiver is idle (i.e., no TLPs or DLLPs are received) for the amount of time specified above.
- The port has accumulated enough flow-control header and data credits to transmit the largest possible packet of each type (i.e., posted, non-posted, or completion).

A port configured in NT function mode or NT with DMA function mode initiates L1 entry when all of the conditions listed below are met:

- L1 ASPM is enabled in the PCIELCTL register of all functions of the port.
- The port has no TLPs pending for transmission on the link.The port's EFB is empty.
- The port has no DLLPs pending for transmission.
- The port's receiver is idle (i.e., no TLPs or DLLPs are received) for the amount of time specified above.
- The port has accumulated enough flow-control header and data credits to transmit the largest possible packet of each type (i.e., posted, non-posted, or completion).

A port configured in upstream switch port with NT function mode or upstream switch port with NT and DMA function mode will enter L1 when all of the conditions listed below are met:

- L1 ASPM is enabled in the PCIELCTL register of all functions of the port.
- All of the downstream switch ports in the switch partition (i.e., associated with the PCI-to-PCI bridge function) which are not in a low power state (i.e., D3) and whose link is not down are in the L1 state.
- The port has TLPs pending for transmission on the link. The port's EFB is empty.
- The port has no DLLPs pending for transmission.
- The port's receiver is idle (i.e., no TLPs or DLLPs are received) for the amount of time specified above.
- The port has accumulated enough flow-control header and data credits to transmit the largest possible packet of each type (i.e., posted, non-posted, or completion).

L1 Exit Conditions

The L1 exit conditions depend on the port's operational mode. A port configured in upstream switch port mode initiates exit from L1 when either of the conditions listed below is met:

- The port has a TLP scheduled for transmission on the link.
- A downstream switch port in the switch partition has initiated exit from L1. The latency between the downstream switch port's initiated exit from L1 and the upstream port's initiated exit from L1 must not exceed 1 µs.

[1] L0s entry is subject to the rules specified in section L0s ASPM on page 7-12.

Notes

A port configured in downstream switch port mode initiates exit from L1 when either of the conditions listed below is met:

- The port has a TLP scheduled for transmission on the link.
- The upstream port in the switch partition has initiated exit from L1. The latency between the upstream port's initiated exit from L1 and the downstream switch port's initiated exit from L1 must not exceed 1 µs.

A port configured in NT function mode or NT with DMA function mode initiates exit from L1 when the condition listed below is met:

- The port has a TLP scheduled for transmission on the link.

Finally, a port configured in upstream switch port with NT function mode or upstream switch port with NT and DMA function mode will enter L1 when either of the conditions listed below is met:

- The port has a TLP scheduled for transmission on the link.
- A downstream switch port in the switch partition associated with the PCI-to-PCI bridge function has initiated exit from L1. The latency between the downstream switch port's initiated exit from L1 and the upstream port's initiated exit from L1 must not exceed 1 µs.

L1 ASPM Entry Rejection Timer

When L1 is enabled by the ASPM field in the PCI Express Link Control (PCIELCTL) register, the PES32NT8xG2 downstream switch ports respond to link partner requests to enter the L1 ASPM state.

In order to enter the L1 ASPM link state, a downstream device (e.g., endpoint) sends continuous PM_Active_State_Request_L1 DLLPs to its link partner (e.g., a downstream switch port). This process continues until the downstream device receives an acceptance or rejection from its link partner.

A PES32NT8xG2 downstream switch port can choose to accept or reject the request, depending on a variety of conditions (refer to section L1 Entry Conditions on page 7-13). When accepting a request, the PES32NT8xG2 downstream switch port sends continuous PM_Request_Ack DLLPs until the downstream device receives these and sends an electrical idle ordered set, effectively placing the link in L1 state.

When rejecting a request, the PES32NT8xG2 downstream switch port sends a single PM_Active_State_Nak TLP. The downstream device, upon reception of this TLP, should place its transmitter into the L0s state, and exit this state prior to sending a new L1 ASPM entry request. Optionally, the downstream device may keep the link in L0 state, in which case it must wait at least 10 µs before sending a new L1 ASPM entry request.

Some endpoint devices do not meet the required 10 µs gap between consecutive L1 ASPM entry requests. A live-lock situation can develop in the following scenario:

The Endpoint sends continuous PM_Active_State_Request_L1 DLLPs to the downstream switch port of a switch.

The switch receives the request but decides to reject (i.e., due to a TLP already queued for transmission on this link). The switch sends a PM_Active_State_Nak TLP to the endpoint device.

The endpoint device notices the rejection, waits an amount of time (i.e., 8 µs) and resumes transmission of PM_Active_State_Request_L1 DLLPs.

The switch receives PM_Active_State_Request_L1 DLLPs, but does not recognize them as a new L1 ASPM entry request, since there was a violation of the 10 us gap between L1 ASPM entry requests.

The switch does not respond with an acceptance or rejection. Therefore, the endpoint keeps waiting for an acceptance or rejection. A deadlock condition develops.

To avoid this deadlock condition, PES32NT8xG2 downstream switch ports allow programmability of a timer that checks for the 10 µs gap between L1 ASPM entry requests. There is a timer per port. The Minimum Time between L1 Entry Requests (MTL1ER) field in the L1 ASPM Rejection Timer Control (L1ASPMRTC) register may be programmed for this purpose.

Notes

The L1ASPMRTC register is located in the proprietary port-specific registers located in the PCI-to-PCI bridge function's configuration space (see section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11). This timer may be programmed from the nano-second range (i.e., 100 ns) up to the micro-second range (i.e., 64 µs). By default, the timer is set to 9.5 µs (refer to the Implementation note in Section 5.4.1.2 of the PCI Express Base Specification 2.1).

Normally, this timer starts its count after the switch downstream switch port issues an L1 ASPM rejection (i.e., PM_Active_State_Nak TLP), without checking activity on the link. The switch also provides an option to start the timer after the downstream switch port issues an L1 ASPM rejection (i.e., PM_Active_State_Nak TLP) and no activity is detected on the receive-lanes. The Timer Start Control (TSCTL) field in the L1ASPMRTC register controls this behavior.

This feature allows the PES32NT8xG2 downstream switch ports to enter L1 ASPM with a variety of endpoints, even those that don't meet the 10 µs gap between subsequent L1 ASPM entry requests.

Link Status

Associated with each switch port is a Port Link Up (PxLINKUPN) status output and a Port Activity (PxACTIVEN) status output. These outputs are provided on an I/O expander for all ports. In addition, the port 0, port 4, port 8, and port 16 link up and activity outputs are also provided via GPIO alternate functions (refer to Chapter 13).

The PxLINKUPN and PxACTIVEN status outputs may be used to provide a visual indication of system state and activity or for debug. The PxLINKUPN output is asserted when the port's data link layer is up (i.e., when the LTSSM is in the L0, L0s, L1 or recovery states). When the data link layer is down, this output is negated.

In the L2/L3-Ready state, the PES32NT8xG2 considers the link to be down. The PxLINKUPN signal is therefore deasserted in this state.

The PxACTIVEN output is asserted whenever any TLP, other than a vendor defined message, is transmitted or received on the corresponding port's link. Whenever a PxACTIVEN output is asserted, it remains asserted for at least 200 ms. Since an I/O expander output may change no more frequently than once every 40 ms, this translates into five I/O expander update periods.

De-emphasis Negotiation

The PCI Express Base Specification requires that components support the following levels of de-emphasis, depending on the link data rate:

- 2.5 GT/s (Gen 1): De-emphasis = -3.5 dB
- 5.0 GT/s (Gen 2): De-emphasis = -3.5 dB or -6.0 dB

When operating at 5.0 GT/s, the de-emphasis is selected by programming the Selectable De-emphasis (SDE) field in the port's PCI Express Link Control 2 Register (PCIELCTL2). The chosen de-emphasis for the link is the result of a negotiation between the two components of the link. Both components must operate with the same de-emphasis across all lanes of the link.

During normal link operation (i.e, PHY LTSSM not in the polling.compliance state), de-emphasis selection is done during the Recovery state. The downstream component of the link (e.g., switch upstream port or endpoint) advertises its desired de-emphasis by transmission of training sets. The upstream component of the link (e.g., switch downstream switch port or root-complex port) notes its link partner desired de-emphasis, and makes a decision about the de-emphasis to be used in the link.

The PES32NT8xG2's upstream port physical layer advertises its desired de-emphasis based on the setting of the SDE field in the PCIELCTL2 register of function 0 of the port. The upstream port always accepts the link-partners decision on the de-emphasis to be used in the link. The PES32NT8xG2's downstream switch ports ignore the link partner's desired de-emphasis and always choose the de-emphasis setting in the SDE field of the port's PCIELCTL2 register.

Notes

Crosslink

PES32NT8xG2 ports support the optional crosslink capability specified in the PCI Express Base Specification. Per this specification, a crosslink is established between two downstream switch ports or two upstream ports. The device's ports are capable of establishing crosslink with any link partner, including another switch port.

When two PES32NT8xG2 switches are crosslinked to each other, it is recommended that the crosslink connection be done among ports in different port groups, as shown in Table 7.1. In order for ports in the same port group (port 0 and port 4) to form a crosslink, software must set the SEED field in the crosslinked port's Phy PRBS Seed (PHYPRBS) register to different values.

Port Groups	
Group 0	Group 1
0	2
4	6
8	—
12	—
16	—
20	—

Table 7.1 Crosslink Port Groups

Note that when an upstream port is crosslinked to a link-partner's upstream port, neither port may automatically initiate a link speed change to Gen 2, thereby resulting in a Gen 1 link. It is possible to overcome this by setting the ILSCC bit in the upstream port's PHYLCFG0 register. By setting this bit, the upstream port will initiate the link transition to Gen 2 speed.

Crosslink is enabled by default. Crosslink may be disabled by setting the Crosslink Disable (CLINKDIS) bit in the port's Phy Link Configuration 0 (PHYLCFG0) register.

Hot Reset Operation on a Crosslink

When a PES32NT8xG2 port forms a crosslink, hot reset operates as follows.

- For a port operating in downstream switch port mode:
  - Regardless of the port's physical layer mode of operation (i.e., downstream lanes or upstream lanes):

    If a higher layer directs the port to hot reset (e.g., partition hot reset, upstream secondary hot reset, downstream secondary hot reset), the physical layer enters the recovery state and proceeds to the hot reset state, as specified in the PCI Express Base Specification.

    The physical layer responds to the reception of training sets with the hot reset bit set by transitioning to the hot reset state as specified in the PCI Express Base Specification. The hot reset does not reset the configuration registers of the port and does not affect other ports in the partition.
- For a port operating in upstream switch port mode:
  - There is no higher layer mechanism to place the port in hot reset state.
  - Regardless of the port's physical layer mode of operation (i.e., downstream lanes or upstream lanes), the physical layer responds to the reception of training sets with the hot reset bit set by transitioning to the hot reset state. The hot reset has the effect described in section Partition Hot Reset on page 3-11.

Notes

Link Disable Operation on a Crosslink

When a port is crosslinked, link disable operates as follows.

- For a port operating in downstream switch port mode:
  - Regardless of the port's physical layer mode of operation (i.e., downstream lanes or upstream lanes):

    If a higher layer directs the port to disable the link (i.e., the Link Disable (LDIS) bit is set in the port's PCIELCTL register), the physical layer enters the recovery state and proceeds to the disabled state, as specified in the PCI Express Base Specification.

    The physical layer responds to the reception of training sets with the disabled bit set by transitioning to the disabled state as specified in the PCI Express Base Specification.
- For a port operating in upstream switch port mode:
  - There is no higher layer mechanism to place the port's link in the disabled state.[1]
  - Regardless of the port's physical layer mode of operation (i.e., downstream lanes or upstream lanes), the physical layer responds to the reception of training sets with the disabled bit set by transitioning to the disabled state as specified in the PCI Express Base Specification.

Gen 1 Compatibility Mode

PES32NT8xG2 ports may be configured to operate in 'Gen 1 Compatibility Mode'. The intent of this mode is to overcome interoperability problems that arise when PCI Express Base 2.1 devices link train with devices that conform to the PCI Express Base 1.1 or earlier specifications (i.e., Gen 1 devices). Specifically, this mode overcomes the problem in which Gen 1 devices react incorrectly to newly defined bits in the PCI Express Base Specification 2.1 for the PHY training sets. Such bits include bits 2, 6, and 7 in symbol four of the TS1 and TS2 training sets.

A switch port is placed in Gen 1 Compatibility Mode by setting the Gen 1 Compatibility Mode Enable (G1CME) bit in the PHYLCFG0 register and fully retraining the link (i.e., via the FLRET bit the PHYLSTATE0 register). These registers are located in the proprietary port-specific registers located in the PCI-to-PCI bridge function's configuration space (see section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11).

When a switch port operates in Gen 1 Compatibility Mode, the PHY does not set the bits listed in Table 7.2 in the training sets that it transmits.

Training Set	Symbol	Bit	PCI Express Base 1.1 and earlier Definition	PCI Express Base 2.1 Definition
TS1	4	2	Reserved	5.0 GT/s Data Rate Support
		6		Multiple meanings (refer to PCI Express Base 2.1 Specification)
		7		Speed Change
TS2	4	2	Reserved	5.0 GT/s Data Rate Support
		6		Multiple meanings (refer to PCI Express Base 2.1 Specification)
		7		Speed Change

Table 7.2 Gen 1 Compatibility Mode: bits cleared in training sets

1. Note that a port that is placed in the disabled operating mode (see section Switch Ports on page 5-5) does not place its physical layer in the disabled state, but rather transitions the physical layer directly to the detect state.

Notes

A switch port exits Gen 1 Compatibility Mode by clearing the G1CME field in the PHYLCFG0 register and fully retraining the link (i.e., via the FLRET bit the PHYLSTATE0 register). When this occurs, the training set bits listed in Table 7.2 behave per the definition in the PCI Express Base Specification.

Notes

SerDes

Notes

Overview

This chapter describes the controllability of the Serializer-Deserializer (SerDes) block associated with each PES32NT8xG2 port. A SerDes block is composed of the serializing/deserializing logic for four PCI Express lanes (i.e., a SerDes "quad"), plus a central unit that controls the quad as a whole. This central unit is called CMU, and contains functionality such as a PLL to generate a high-speed clock used by each lane, initialization of the quad, etc.

In order to improve signal integrity across the high-speed PCI Express links, the PES32NT8xG2 allows per-lane programmability of several SerDes settings. These include the following.

- Transmitter drive level
- Transmitter de-emphasis level
- Receiver equalization

In addition, the PES32NT8xG2 supports the optional "low-swing mode" specified by the PCI Express Base Specification 2.1. This mode is intended for power-sensitive applications. This chapter describes these controls, their intended use, and the manner in which they are programmed. Before this is discussed, the topic of SerDes numbering and port association is introduced. To modify the SerDes driver and receiver settings for a port, the SerDes quad and specific lanes associated with the port must be identified as described in section SerDes Numbering and Port Association on page 8-1.

SerDes Numbering and Port Association

The PES32NT8xG2 contains eight SerDes quads, numbered 0 to 7. Tables 8.1 through 8.4 list the ports in the switch and the SerDes quads with which they are associated. The SerDes/port association depends on the configuration of the corresponding stack[1], as shown in the tables.

- SerDes / Port association for stack configurations not shown in the tables can be easily derived from the basic configurations shown in the table.

Note that in some stack configurations, several ports whose width is less than x4 share a SerDes quad (i.e., a SerDes quad could be shared by four x1 ports or by two x2 ports). Still, the SerDes lanes associated with different ports operate independently, such that the ports can operate at different data rates, power states, drive and de-emphasis levels, etc.

- The exception to the above assertion is that all ports that share a SerDes quad must operate in the same clocking mode. See section Port Clocking Modes on page 2-3 for further details on this.

To modify some of the SerDes driver and receiver settings (e.g., drive swing, receiver equalization) for a port, the SerDes quad and specific lanes associated with the port must be identified via the tables shown below.

- For example, as shown Table 8.1, lanes 3 through 0 of SerDes quad 1 are associated with port 2, lanes 0 through 3. Therefore, to modify the SerDes driver and receiver settings of port 2, the configuration registers associated with SerDes quad 1, lanes 3 through 0 should be modified. The next sections describe the settings and corresponding configuration registers in detail.

1. As mentioned in section Stack Configuration on page 3-5, the switch contains four stacks, each of which is associated with 2 ports. Refer to Stack Configuration for further details.

Notes

Stack 0 Configuration	SerDes Quad 1				SerDes Quad 0			
	Lane3	Lane 2	Lane 1	Lane 0	Lane3	Lane 2	Lane 1	Lane 0
x8	Port 0							
	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x4, x4	Port 2				Port 0			
	Lane 3	Lane 2	Lane 1	Lane 0	Lane 3	Lane 2	Lane 1	Lane 0

Table 8.1 SerDes / Port Association for Ports in Stack 0

Stack 1 Configuration	SerDes Quad 3				SerDes Quad 2			
	Lane3	Lane 2	Lane 1	Lane 0	Lane3	Lane 2	Lane 1	Lane 0
x8	Port 4							
	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x4, x4	Port 6				Port 4			
	Lane 3	Lane 2	Lane 1	Lane 0	Lane 3	Lane 2	Lane 1	Lane 0

Table 8.2 SerDes / Port Association for Ports in Stack 1

Stack 2 Configuration	SerDes Quad 5				SerDes Quad 4			
	Lane3	Lane 2	Lane 1	Lane 0	Lane3	Lane 2	Lane 1	Lane 0
x8	Port 8							
	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x4, x4	Port 12				Port 8			
	Lane 3	Lane 2	Lane 1	Lane 0	Lane 3	Lane 2	Lane 1	Lane 0

Table 8.3 SerDes / Port Association for Ports in Stack 2

Stack 3 Configuration	SerDes Quad 7				SerDes Quad 6			
	Lane3	Lane 2	Lane 1	Lane 0	Lane3	Lane 2	Lane 1	Lane 0
x8	Port 16							
	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x4, x4	Port 20				Port 16			
	Lane 3	Lane 2	Lane 1	Lane 0	Lane 3	Lane 2	Lane 1	Lane 0

Table 8.4 SerDes / Port Association for Ports in Stack 3

Notes

SerDes Transmitter Controls

The PES32NT8xG2 allows programmability of SerDes transmitter voltage level and de-emphasis, including support for the PCI Express optional low-swing mode, as well as a proprietary “amplitude boost” feature to increase the drive strength above its normal operating level (e.g., for operation across long traces).

Except for low-swing mode, which is defined by PCI Express as a per-link function, all the other controls are proprietary and provided on a per-lane basis. This allows a system designer to customize the SerDes transmitter settings for each lane independently. At the 5.0 GT/s speed (i.e., Gen 2) and above, small differences in the channel characteristics among lanes may result in noticeable differences in the quality of the signal at the receiver and per-lane controllability is an important tool in improving the bit-error rate on the link.

Driver Voltage Level and Amplitude Boost

The PCI Express Base Specification requires that each port support the ‘transmit margining’ feature. This feature allows the selection of several voltage settings across the link and is intended for compliance testing and debug. In addition to this, the PES32NT8xG2 offers proprietary fine grain controllability of the SerDes transmitter voltage level, across a wide range of settings. The PES32NT8xG2 places no restrictions on the time at which these settings can be modified (e.g., they can be modified during normal operation of the link or while the link is being tested).

By default, the SerDes transmit level can be programmed in the range from 980 mV to 120 mV, at steps of ~15 mV each. In addition, there is an “amplitude-boost” control that increases the drive level by ~5%.

Together, the controls for drive-level and amplitude boost allow the system designer to select, on a per-lane basis if desired, the appropriate drive strength for the channel. For power sensitive applications, the drive level can be reduced with fine granularity to the desired level, without compromising link reliability.

Note that the PCI Express Base Specification requires that at 5.0 GT/s, receivers accept incoming signals in the range 1.2 V to 0.120 V. Thus, the transmitter voltage settings may be modified without requiring any modification of the link partner’s receiver settings.

Refer to section Programming of SerDes Controls on page 8-4 for procedural details on modifying the default SerDes settings and to section SerDes Transmitter Control Registers on page 8-5 for details on programming the transmitter voltage level and amplitude boost controls.

De-emphasis

The PCI Express Base Specification supports three de-emphasis levels: -3.5 dB (at 2.5 GT/s or 5.0 GT/s speeds), -6.0 dB (only for 5.0 GT/s), and 0 dB (low-swing mode). The de-emphasis selected for the link is controlled by the Selectable De-emphasis bit in each port’s PCI Express Link Control 2 register[1]. This field is set by hardware or firmware (e.g., EEPROM) during boot time and remains unchanged during normal system operation.

To allow the de-emphasis setting to be modified and customized on the link, the PES32NT8xG2 contains proprietary per-lane coarse and fine de-emphasis adjustment controls. Together, these controls allow the nominal de-emphasis setting (i.e., -3.5 dB or -6.0 dB) to be modified with a granularity of ~0.6 dB per setting[2]. The desired de-emphasis setting can be achieved across the range of driver level settings described in the previous section. The PES32NT8xG2 places no restrictions on the time at which the de-emphasis setting can be modified. Refer to section SerDes Transmitter Control Registers on page 8-5 for details on programming the transmitter de-emphasis.

1. In low-swing mode, de-emphasis is automatically set to 0 dB and the Selectable De-emphasis bit in the port’s PCI Express Link Control 2 register is ignored.

2. Note that the PCI Express Base Specification allows a deviation of +/- 0.5 dB from the nominal setting.

Notes

PCI Express Low-Swing Mode

PES32NT8xG2 ports support the optional low-swing transmit voltage mode defined in the PCI Express Base Specification. In this mode, the port's transmitter voltage level is set to approximately half the value of the full-swing (default) mode, which results in reduced power consumption in the SerDes. In addition, signal de-emphasis is turned off. Low-swing mode is a per-link feature, meaning that all lanes of the port operate low-swing simultaneously. Refer to section Low-Swing Transmitter Voltage Mode on page 8-12 for details on enabling low-swing mode on a port.

Receiver Equalization

In addition to the transmitter controls described above, the switch SerDes also contains a receiver equalizer to compensate for effects of channel loss on received signal (i.e., high-speed signal degradation due to the combined effects of board traces, vias, connectors, and cables in the physical link). In general, the channel has low-pass filter characteristics, which results in the degradation of high speed signals. Receiver equalization may be used to compensate for the lossy attenuation effects of the channel on high-speed signals.

Receiver equalization can be controlled on a per-lane basis. Each SerDes lane contains a receiver equalization circuit. This circuit is a multi-stage programmable amplifier, where each stage is a peaking equalizer with a different center frequency and programmable gain. Varying amounts of gain may be applied depending on the overall frequency response of the channel loss.

For details on programming the receiver equalizer, refer to section Receiver Equalization Controls on page 8-14. The PES32NT8xG2 places no restrictions on the time at which the equalizer settings may be modified (e.g., the settings can be modified during normal operation of the link or while the link is being tested).

Programming of SerDes Controls

The SerDes controls described above may be programmed by accessing IDT proprietary registers within the switch. The registers may be programmed via any of the mechanisms allowed by the PES32NT8xG2 (i.e., via PCI Express configuration accesses from a root, via EEPROM loading at boot-time, or via the switch's SMBus slave interface). The following sections describe in detail the control registers associated with the SerDes and the manner in which the SerDes controls are programmed.

Programmable Voltage Margining and De-Emphasis

The PES32NT8xG2 contains SerDes transmitter voltage controls, which operate on a per-port, per-quad, or per-lane basis. There are two mechanisms to control the SerDes transmitter voltage level:

- Via the Transmit Margin (TM) field of the associated port's Link Control 2 Register (PCIELCTL2).
- Via proprietary SerDes transmitter control registers
  - These registers are associated with each SerDes quad. Each SerDes quad has independent transmitter control registers. To modify the settings for the lanes of a port, the SerDes quad associated with the port must first be determined. The association between SerDes quads and ports is described in section SerDes Numbering and Port Association on page 8-1. As indicated in that section, the SerDes lanes associated with each port depend on the configuration of the stack associated with the port.
  - The SerDes Lane Transmitter Control Registers (S[x]TXLCTL0 and S[x]TXLCTL1, where 'x' refers to the SerDes quad number) are the registers that control the transmit settings of the corresponding SerDes quad. S[0]TXLCTL0 and S[0]TXLCTL1 are associated with SerDes quad 0, S[1]TXLCTL0 and S[1]TXLCTL1 are associated with SerDes quad 1, and so on.
  - The S[x]TXLCTL0 and S[x]TXLCTL1 registers may be used to control transmit driver settings per-lane.[1]

1. The S[x]TXLCTL0 and S[x]TXLCTL1 registers are used in conjunction with the SerDes Control (S[x]CTL) register in order to apply the settings to a particular lane or all lanes of the SerDes. Please refer to the description of the S[x]CTL register for further details.

Notes

The selection of which of the two mechanism controls the SerDes transmit voltage is based on the setting of the TM field in the associated port's PCIELCTL2 register.

When the Transmit Margin (TM) field in the port's PCIELCTL2 register is set to 'Normal Operating Range', the transmitter voltage level for each SerDes lane of the port is controlled via the corresponding S[x]TXLCTL0 and S[x]TXLCTL1 registers. Otherwise, the TM field controls the SerDes voltage directly for all SerDes lanes associated with the port.

- For instance, when port 0 is configured as a x4 port, it is associated with SerDes quad 0, lanes 3 to 0 (see Table 8.1). If the TM field in the port's PCIELCTL2 register is set to 'Normal Operating Range', then the S[0]TXLCTL0 and S[0]TXLCTL1 registers control the operating voltage of the port's SerDes lanes. If the TM field is set to another value, the voltage on the SerDes lanes associated with port 0 is set to the value in the port's PCIELCTL2.TM field.
- As another example, when port 4 is configured as a x4 port, it is associated with SerDes quad 2, lanes 0 through 3 (see Table 8.2). If the TM field in the port's PCIELCTL2 register is set to 'Normal Operating Range', then the S[2]TXLCTL0 and S[2]TXLCTL1 registers control the operating voltage of the port's SerDes lanes. These registers must be configured to operate on lanes 0 through 3 in order to affect the lanes associated with port 4. If the TM field is set to another value, the voltage on the SerDes lanes associated with port 4 is set to the value in the port's PCIELCTL2.TM field.
- As a final example, when port 16 is configured as a x8 port, it is associated with SerDes quads 6 and 7 (see Table 8.4). If the TM field in the port's PCIELCTL2 register is set to 'Normal Operating Range', then the S[6]TXLCTL0, S[6]TXLCTL1, S[7]TXLCTL0, and S[7]TXLCTL1 registers control the operating voltage of the port's SerDes lanes. If the TM field is set to another value, the voltage on the SerDes lanes associated with port 16 is set to the value in the port's PCIELCTL2.TM field.

De-emphasis levels may also be adjusted on a per-lane basis, using the above mentioned transmitter control registers. Nominally, de-emphasis levels are set to -3.5 dB, -6.0 dB, or 0 dB (in low-swing mode). The S[x]TXLCTL0 and S[x]TXLCTL1 registers can be used to modify the nominal values by coarse or fine steps.

SerDes Transmitter Control Registers

As described above, each switch SerDes quad is associated with two transmitter control registers (S[x]TXLCTL0 and S[x]TXLCTL1). Together, these registers allow full programmability of the SerDes transmitter voltage levels and de-emphasis. These registers are segmented into fields that allow programmability of the transmit driver levels under the following *PHY operating modes*:

- Full-Swing Mode, in Gen 1 data rate, with -3.5 dB de-emphasis
- Full-Swing Mode, in Gen 2 data rate, with -3.5 dB de-emphasis
- Full-Swing Mode, in Gen 2 data rate, with -6.0 dB de-emphasis
- Low-Swing Mode, in Gen 1 data rate (no de-emphasis)
- Low-Swing Mode, in Gen 2 data rate (no de-emphasis)

The S[x]TXLCTL0 and S[x]TXLCTL1 registers have default values that select the appropriate transmit driver settings for each of the above modes. These default values may be modified to adjust the drive levels. When the Physical layer of the port associated with the SerDes transitions dynamically across these operating modes, the appropriate driver settings are applied to the SerDes automatically.

- For example, when the PHY operates in Full-swing mode at Gen 1 data rate with -3.5 dB de-emphasis, the SerDes transmit settings are set to the values specified in the S[x]TXLCTL0 and S[x]TXLCTL1 registers corresponding to that operating mode (e.g., Full-Swing mode at Gen 1 data rate with -3.5 dB de-emphasis). As the PHY changes data rate to Gen 2, the SerDes transmit settings are automatically modified to the values specified in the S[x]TXLCTL0 and S[x]TXLCTL1 registers corresponding to the new operating mode (e.g., Full-Swing mode at Gen 2 data rate with -3.5 dB de-emphasis).

Table 8.5 shows the register fields that control the SerDes transmit levels for the operation modes listed above.

Notes

PHY Operation Mode			Relevant fields in S[x]TXLCTL0	Relevant fields in S[x]TXLCTL1
Voltage Swing	Data Rate	De-emphasis	Fine De-emphasis Control	Drive Level / Fine De-emphasis Control
Full-Swing	2.5 GT/s	-3.5 dB	FDC_FS3DBG1	TDVL_FS3DBG1 / CDC_FS3DBG1
Full-Swing	5.0 GT/s	-3.5 dB	FDC_FS3DBG2	TDVL_FS3DBG2 / CDC_FS3DBG2
Full-Swing	5.0 GT/s	-6.0 dB	FDC_FS6DBG2	TDVL_FS6DBG2 / CDC_FS6DBG2
Low-Swing	2.5 GT/s	0 dB	N/A	TDVL_LSG1
Low-Swing	5.0 GT/s	0 dB		TDVL_LSG2

Table 8.5 SerDes Transmit Level Controls in the S[x]TXLCTL0 and S[x]TXLCTL1 Registers

As shown in Table 8.5, there are three parameters that may be programmed to adjust the transmitter drive levels (per-lane). These are:

- Fine De-emphasis Control (in the S[x]TXLCTL0 register)
- Coarse De-emphasis Control (in the S[x]TXLCTL1 register)
- Drive Level Control (in the S[x]TXLCTL1 register).

Modification of these settings takes an immediate effect on the SerDes. Therefore, the link does not need to be retrained explicitly (i.e., via the link-retrain (LRET) bit in the port's PCIELCTL register) in order for these settings to take effect. Still, the user must be careful when modifying SerDes settings while the port is in normal operating mode, as this may result in link instability.

Table 8.6 shows a number of possible settings for the drive and de-emphasis in Gen 1 mode[1]. These can be used as guidance when adjusting the SerDes transmit drive swing. The default setting is highlighted. Note that in Gen 1 mode, de-emphasis is ideally -3.5 dB with +/- 0.5 dB error (refer to Section 4.3.3.5 of the PCI Express Base Base 2.1 Specification). The settings listed in the table ensure that the de-emphasis is kept within the allowable range.

Transmit Levels			**Settings of Relevant Fields in the S[x]TXLCTL0 & S[x]TXLCTL1 Registers**			
Drive Level (mV)	**De-emphasis (dB)**	**De-emphasized Drive Level (mV)**	**TDVL_FS3DBG1**	**CDC_FS3DBG1**	**FDC_FS3DBG1**	**TX_SLEW_G1**
887	-3.5	590	0x13	0x3	0x2	0x1
872	-3.6	576	0x12	0x3	0x2	0x1
840	-3.5	558	0x11	0x3	0x2	0x1
808	-3.5	541	0x10	0x3	0x2	0x1

Table 8.6 SerDes Transmit Driver Settings in Gen 1 Mode with -3.5 dB de-emphasis (Part 1 of 2)

1. Table values are based on simulations using the Snowbush SerDes HSPICE model and device package s-parameters. Values are sampled at the device pins. The simulation assumes typical conditions, with VddPEA = VddPETA = 1.0V, VddPEHA = 2.5V, and TX_AMPBOOST = 0x0. The values in the tables may differ from those of post-silicon characterization. Please refer to the device data sheet for post-silicon device characterization data.

Notes

Transmit Levels			Settings of Relevant Fields in the S[x]TXLCTL0 & S[x]TXLCTL1 Registers			
Drive Level (mV)	De-emphasis (dB)	De-emphasized Drive Level (mV)	TDVL_FS3DBG1	CDC_FS3DBG1	FDC_FS3DBG1	TX_SLEW_G1
776	-3.4	523	0xF	0x3	0x1	0x1
744	-3.4	505	0xE	0x3	0x1	0x1
712	-3.3	487	0xD	0x3	0x1	0x1
674	-3.4	457	0xC	0x3	0x1	0x1
635	-3.5	427	0xB	0x3	0x1	0x1
597	-3.6	397	0xA	0x3	0x1	0x1
558	-3.7	366	0x9	0x3	0x1	0x1
512	-3.6	336	0x8	0x3	0x1	0x1
465	-3.6	306	0x7	0x3	0x0	0x1
419	-3.6	275	0x6	0x3	0x0	0x1
372	-3.6	245	0x5	0x2	0x3	0x1
324	-3.8	211	0x4	0x2	0x3	0x1
276	-4.0	177	0x3	0x2	0x2	0x1

Table 8.6 SerDes Transmit Driver Settings in Gen 1 Mode with -3.5 dB de-emphasis (Part 2 of 2)

Table 8.7 shows a number of possible settings for the drive and de-emphasis in Gen 2 mode with -3.5 dB de-emphasis[1]. The default setting is highlighted.

Transmit Levels			Settings of Relevant Fields in the S[x]TXLCTL0 & S[x]TXLCTL1 Registers			
Drive Level (mV)	De-emphasis (dB)	De-emphasized Drive Level (mV)	TDVL_FS3DBG2	CDC_FS3DBG2	FDC_FS3DBG2	TX_SLEW_G2
858	-3.7	558	0x1C	0x3	0x3	0x0
855	-3.6	563	0x1B	0x3	0x3	0x0
852	-3.5	569	0x1A	0x3	0x3	0x0
849	-3.4	574	0x19	0x3	0x3	0x0
846	-3.3	580	0x18	0x3	0x3	0x0
837	-3.2	578	0x17	0x3	0x3	0x0
828	-3.1	577	0x16	0x3	0x3	0x0

Table 8.7 SerDes Transmit Driver Settings in Gen 2 Mode with -3.5 dB de-emphasis (Part 1 of 2)

1. Table values are based on simulations using the Snowbush SerDes HSPICE model and device package s-parameters. Values are sampled at the device pins. The simulation assumes typical conditions, with VddPEA = VddPETA = 1.0V, VddPEHA = 2.5V, and TX_AMPBOOST = 0x0. The values in the tables may differ from those of post-silicon characterization. Please refer to the device data sheet for post-silicon device characterization data.

Notes

Transmit Levels			Settings of Relevant Fields in the S[x]TXLCTL0 & S[x]TXLCTL1 Registers			
Drive Level (mV)	De-emphasis (dB)	De-emphasized Drive Level (mV)	TDVL_FS3DBG2	CDC_FS3DBG2	FDC_FS3DBG2	TX_SLEW_G2
820	-3.1	576	0x15	0x3	0x3	0x0
811	-3.0	574	0x14	0x3	0x3	0x0
794	-3.0	562	0x13	0x3	0x3	0x0
776	-3.0	550	0x12	0x3	0x3	0x0
750	-3.0	529	0x11	0x3	0x3	0x0
724	-3.1	508	0x10	0x3	0x3	0x0
699	-3.1	487	0xF	0x3	0x3	0x0
673	-3.2	467	0xE	0x3	0x3	0x0
647	-3.2	446	0xD	0x3	0x3	0x0
615	-3.4	419	0xC	0x3	0x3	0x0
583	-3.5	392	0xB	0x3	0x3	0x0
551	-3.6	364	0xA	0x3	0x3	0x0
519	-3.7	337	0x9	0x3	0x3	0x0
474	-3.7	310	0x8	0x3	0x3	0x0
430	-3.6	283	0x7	0x3	0x3	0x0
386	-3.5	256	0x6	0x3	0x2	0x0
341	-3.5	229	0x5	0x3	0x2	0x0
295	-3.4	198	0x4	0x3	0x2	0x0
248	-3.4	167	0x3	0x3	0x2	0x0
201	-3.4	136	0x2	0x3	0x1	0x0
155	-3.4	104	0x1	0x3	0x1	0x0
108	-3.4	73	0x0	0x3	0x1	0x0

Table 8.7 SerDes Transmit Driver Settings in Gen 2 Mode with -3.5 dB de-emphasis (Part 2 of 2)

Table 8.8 shows a number of possible settings for thedrive and de-emphasis in Gen 2 mode with -6.0 dB de-emphasis[1]. The default setting is highlighted. As mentioned above, the PCI Express Base Specification allows an error of up to +/- 0.5 dB on the de-emphasis. All settings listed in the table ensure that the de-emphasis is kept within the allowable range.

1. Table values are based on simulations using the Snowbush SerDes HSPICE model and device package s-parameters. Values are sampled at the device pins. The simulation assumes typical conditions, with VddPEA = VddPETA = 1.0V, VddPEHA = 2.5V, and TX_AMPBOOST = 0x0. The values in the tables may differ from those of post-silicon characterization. Please refer to the device data sheet for post-silicon device characterization data.

Notes

Transmit Levels			Settings of Relevant Fields in the S[x]TXLCTL0 & S[x]TXLCTL1 Registers			
Drive Level (mV)	De-emphasis (dB)	De-emphasized Drive Level (mV)	TDVL_FS6DBG2	CDC_FS6DBG2	FDC_FS6DBG2	TX_SLEW_G2
845	-6.3	407	0x19	0x6	0x3	0x0
842	-6.4	402	0x18	0x6	0x3	0x0
838	-6.3	407	0x17	0x6	0x3	0x0
835	-6.1	411	0x16	0x6	0x3	0x0
831	-6.0	416	0x15	0x6	0x3	0x0
827	-5.9	421	0x14	0x6	0x3	0x0
818	-5.9	416	0x13	0x6	0x3	0x0
808	-5.9	412	0x12	0x6	0x3	0x0
784	-5.9	398	0x11	0x6	0x3	0x0
760	-5.9	385	0x10	0x6	0x3	0x0
736	-5.9	371	0xF	0x5	0x3	0x0
712	-6.0	358	0xE	0x5	0x3	0x0
688	-6.0	344	0xD	0x5	0x3	0x0
652	-5.9	329	0xC	0x5	0x3	0x0
617	-5.8	315	0xB	0x5	0x3	0x0
581	-5.7	300	0xA	0x5	0x2	0x0
545	-5.6	285	0x9	0x5	0x2	0x0
501	-5.7	260	0x8	0x5	0x2	0x0
458	-5.8	235	0x7	0x5	0x2	0x0
414	-5.9	210	0x6	0x5	0x2	0x0
370	-6.0	185	0x5	0x5	0x2	0x0
319	-5.9	161	0x4	0x5	0x2	0x0
268	-5.8	136	0x3	0x5	0x2	0x0
217	-5.7	112	0x2	0x5	0x1	0x0
166	-5.6	88	0x1	0x5	0x1	0x0
115	-5.5	63	0x0	0x5	0x1	0x0

Table 8.8 SerDes Transmit Driver Settings in Gen 2 Mode with -6.0 dB de-emphasis

When the PHY operates in low-swing mode, de-emphasis is automatically turned off. Therefore, the fine and coarse de-emphasis controls in the S[x]TXLCTL0 and S[x]TXLCTL1 registers have no effect. In this mode, the TDVL_LSG1 and TDVL_LSG2 fields in the S[x]TXLCTL1 register control the transmitter voltage swing for Gen 1 and Gen 2 modes respectively. Refer to section Low-Swing Transmitter Voltage Mode on page 8-12 for further details.

Notes

In addition to the SerDes settings described above, the user may apply an amplitude boost to the drive swing by setting the TX_AMPBOOST field in the S[x]TXLCTL0 register. Amplitude boost may be applied on a per-lane basis. Amplitude boost may be applied to increase the drive swings above the values shown in Tables 8.6, 8.7, and 8.8. Refer to the description of the TX_AMPBOOST field for further details.

Programmable De-emphasis Adjustment

The tables shown in the previous section list different settings to control the SerDes drive swing while keeping the de-emphasis within the nomina range, depending on the PHY operating mode (e.g., Gen 1 data rate and -3.5 dB de-emphasis, Gen 2 data rate and -3.5 dB de-emphasis, or Gen 2 data rate with -6.0 dB de-emphasis).

It is possible to modify the de-emphasis in fine or coarse increments on a per-lane basis, using the appropriate fields in the S[x]TXLCTL0 and S[x]TXLCTL1 registers. Table 8.5 shows the register fields that control fine and coarse de-emphasis for each PHY operating mode.

Figure 8.1 shows the qualitative relationship between the coarse and fine de-emphasis controls and the resulting de-emphasis.

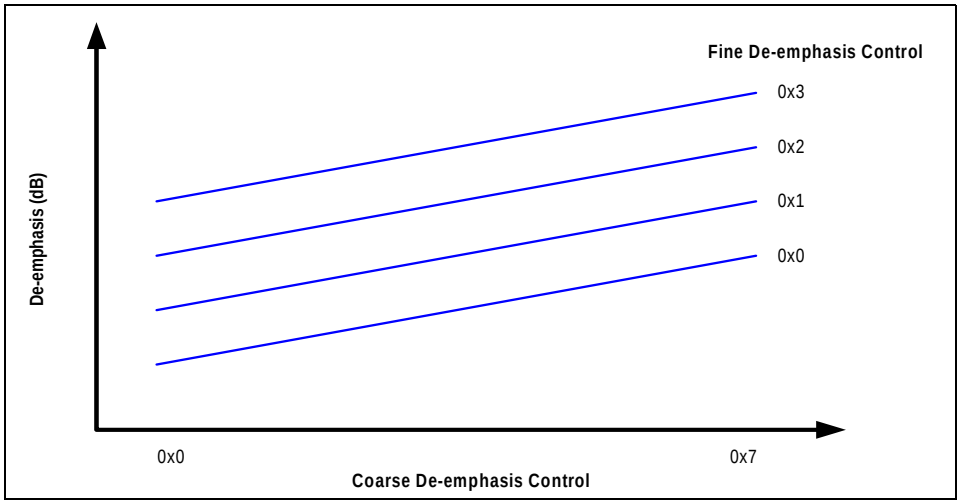

Figure 8.1 Relationship Between Coarse and Fine De-emphasis Controls

When using the de-emphasis controls, it is important to understand that the actual de-emphasis applied on the link is a function of the de-emphasis controls, the transmit drive swing controls, and the data rate of the SerDes.

The coarse de-emphasis controls should generally be set as shown in Tables 8.6, 8.7, and 8.8. Note that there is a coarse de-emphasis control per PHY operating mode.

- When the PHY operates in Gen 1 data rate with -3.5 dB de-emphasis, the coarse de-emphasis is controlled by the CDC_FS3DBG1 field in the S[x]TXLCTL1 register.
- When the PHY operates in Gen 2 data rate with -3.5 dB de-emphasis, the coarse de-emphasis is controlled by the CDC_FS3DBG2 field in the S[x]TXLCTL1 register.
- When the PHY operates in Gen 2 data rate with -6.0 dB de-emphasis, the coarse de-emphasis is controlled by the CDC_FS6DBG2 field in the S[x]TXLCTL1 register.

The coarse de-emphasis settings shown in Tables 8.6, 8.7, and 8.8 ensure that the de-emphasis falls within the nominal range mandated by the PCI Express Base Specification.

When using the de-emphasis controls, it is important to understand that the actual de-emphasis applied on the link is a function of the de-emphasis controls, the transmit drive swing controls, and the data rate of the SerDes. The coarse de-emphasis controls should generally be set as shown in Tables 8.6, 8.7, and 8.8. Note that there is a coarse de-emphasis control per PHY operating mode. The coarse de-emphasis settings shown in the above tables ensure that the de-emphasis falls within the nominal range mandated by the PCI

Notes

Express Base Specification. As shown in the tables, the coarse de-emphasis setting is dependent on the transmit drive swing setting. Therefore, modifying the transmit drive swing must be done in conjunction with modifying the coarse de-emphasis setting.

The fine de-emphasis registers allow modification of the de-emphasis in fine steps. There is a fine de-emphasis control per PHY operating mode.

- When the PHY operates in Gen 1 data rate with -3.5 dB de-emphasis, the fine de-emphasis is controlled by the FDC_FS3DBG1 field in the S[x]TXLCTL0 register.
- When the PHY operates in Gen 2 data rate with -3.5 dB de-emphasis, the fine de-emphasis is controlled by the FDC_FS3DBG2 field in the S[x]TXLCTL0 register.
- When the PHY operates in Gen 2 data rate with -6.0 dB de-emphasis, the fine de-emphasis is controlled by the FDC_FS6DBG2 field in the S[x]TXLCTL0 register.

Figure 8.2 shows a plot of the de-emphasis seen at the SerDes transmitter pins as a function of the fine de-emphasis control and transmit drive level controls, when the PHY operates in Gen 2 data rate with -6.0 dB de-emphasis. In the figure, chX_tx_lev[4:0] refers to the TDVL_FS6DBG2 control.

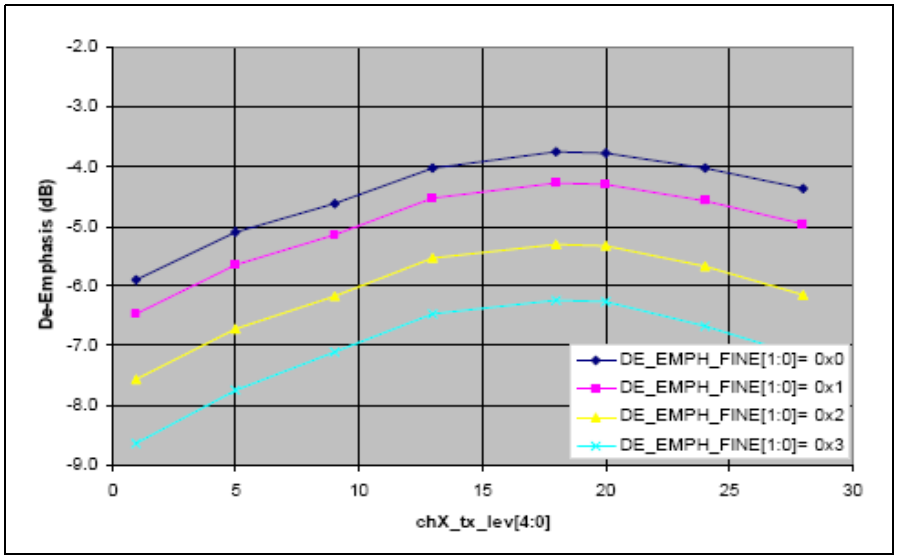

Figure 8.2 Effect of Fine de-emphasis Control at Gen 2 with -6.0 dB Nominal de-emphasis

As shown in Figure 8.2, the de-emphasis applied on the line varies depending on the setting of the transmit drive level field. Thus, when modifying the transmit drive level of the SerDes, the fine de-emphasis control must be adjusted appropriately to ensure that the de-emphasis on the line falls within the range mandated by the PCI Express Base Specification.

Finally, note that it is possible to turn off the de-emphasis (i.e., 0 db de-emphasis) by setting both the coarse and fine de-emphasis settings to a value of 0x0.

Transmit Margining Using the PCI Express Link Control 2 Register

When the Transmit Margin (TM) field in the port's PCIELCTL2 register is set to a value other than 'Normal Operating Range', the transmitter voltage levels are controlled by hardware based on the setting of the TM field, and not by the S[x]TXLCTL0 and S[x]TXLCTL1 registers[1]. Per the PCI Express Base Specification, transmit margining may be done in full-swing mode or in low-swing mode. Table 8.9 shows the transmit margining settings supported by the switch.

[1] The TX_AMPBOOST field in the S[x]TXLCTL0 register does have an effect during transmit margining.

Notes

Full Swing Mode (mV)	Low Swing Mode (mV)
900	500
700	400
500	300
300	200
200	100

Table 8.9 PCI Express Transmit Margining Levels Supported by the PES32NT8xG2

Note that in compliance mode (i.e., when the associated port's PHY LTSSM is in the Polling.Compliance state), the SerDes transmit level is controlled by the TM field in the associated port's PCIELCTL2 register, and the de-emphasis setting is controlled by the LTSSM based on the rules described in Section 4.2.6.2.2 of the PCI Express Base Specification.

- When the LTSSM enters the Polling.Compliance state in full-swing mode, the values for full-swing margining are applied.
- When the LTSSM enters the Polling.Compliance state in low-swing mode, the values for low-swing margining are applied.

Finally, when the TM field is modified, the newly selected value is not applied until the PHY LTSSM transitions through the states in which it is allowed to modify the transmit margin setting on the line (e.g., Recovery.RcvrLock). Therefore, after modifying this field, it is recommended that the link be retrained by setting the LRET bit in the port's PCIELCTL register.

Low-Swing Transmitter Voltage Mode

PES32NT8xG2 ports support the optional low-swing transmit voltage mode defined in the PCI Express Base Specification. In this mode, the port's transmitter voltage level is set to approximately half the value of the full-swing (default) mode, reducing power consumption in the SerDes. This mode is enabled by setting the Low-Swing Enable (LSE) bit in the port's SerDes Configuration (SERDESCFG) register.

- The LSE bit in the port's SERDESCFG register affects all SerDes lanes associated with the port.

When Low-Swing mode is enabled, the transmitter drive level is reduced and de-emphasis is automatically turned off. Therefore, the Selectable De-emphasis (SDE) and Compliance De-emphasis (CDE) fields in the PCIELCTL2 register have no effect. Additionally, the Current De-emphasis (CDE) field in the PCIELSTS2 register becomes invalid. The low-swing mode transmitter voltage swing may be adjusted via the TDVL_LSG1 (when operating in Gen 1 mode) and TDVL_LSG2 (when operating in Gen 2 mode) fields in the S[x]TXLCTL1 register.

Table 8.10 shows the transmitter's drive swing for different values of TDVL_LSG1, when the port operates in low swing mode at Gen 1 speed[1]. Table 8.11 shows the transmitter's drive swing for different values of TDVL_LSG2, when the port operates in low swing mode at Gen 2 speed. The default setting is highlighted.

[1] Table values are based on simulations using the Snowbush SerDes HSPICE model and device package s-parameters. Values are sampled at the device pins. The simulation assumes typical conditions, with VddPEA = VddPETA = 1.0V, VddPEHA = 2.5V, and TX_AMPBOOST = 0x0. The values in the tables may differ from those of post-silicon characterization. Please refer to the device data sheet for post-silicon device characterization data.

Notes

Drive Level (mV)	TDVL_LSG1
671	0x0F
640	0x0E
609	0x0D
573	0x0C
537	0x0B
501	0x0A
465	0x09
426	0x08
388	0x07
349	0x06
311	0x05
270	0x04
230	0x03
189	0x02
148	0x01
108	0x00

Table 8.10 SerDes Transmit Drive Swing in Low Swing Mode at Gen 1 speed

Drive Level (mV)	TDVL_LSG2
600	0x0F
573	0x0E
547	0x0D
514	0x0C
482	0x0B
449	0x0A
416	0x09
382	0x08
347	0x07
312	0x06
278	0x05
241	0x04
203	0x03

Table 8.11 SerDes Transmit Drive Swing in Low Swing Mode at Gen 2 Speed (Part 1 of 2)

Notes

Drive Level (mV)	TDVL_LSG2
166	0x02
129	0x01
91	0x00

Table 8.11 SerDes Transmit Drive Swing in Low Swing Mode at Gen 2 Speed (Part 2 of 2)

When the PHY enters the Polling.Compliance state and low-swing mode is enabled, the following occurs:

- The transmit drive level is selected by the Transmit Margin (TM) field in the PCIELCTL2 register. This field has specific transmit margin levels for full-swing and low-swing mode. The values corresponding to low-swing mode are applied.
- De-emphasis is turned off.

Receiver Equalization Controls

PES32NT8xG2 contains SerDes receiver equalization controls on a per-lane basis. The receiver equalization circuit has two controls which may be programmed via the SerDes Receiver Equalization Lane Control (S[x]RXEQLCTL) register. These are:

- Receiver Equalization Zero (RXEQZ): Increases the high-frequency gain of the equalizer.
- Receiver Equalization Boost (RXEQB): Reduces the low-frequency gain of the equalizer.

Together, RXEQZ and RXEQB provide wide programmability and fine grain control over the equalizer's boost. Refer to the definition of the S[x]RXEQLCTL register for further details on programming these controls.

SerDes Power Management

In order to maximize power savings in the SerDes, the PES32NT8xG2 adheres to the following guidelines. For SerDes quads that are used, their power state depends on the state of the port(s) associated with the SerDes, as described below. When a port is disabled:

- For a x4 or x8 port, the SerDes quad(s) associated with the disabled port are placed in a deep low power state.
  - There is one SerDes quad associated with a x4 port.
  - There are two SerDes quads associated with a x8 port.
- For a x1 or x2 port, the SerDes lanes associated with the disabled port are placed in a deep low power state.
  - If all lanes of a SerDes quad are associated with disabled ports, the entire SerDes quad is placed in a deep low power state.

When a port is not disabled:

- The SerDes quad(s) associated with the port are turned-on.
- Unused lanes are powered down.
  - Lanes that form the initial link width (i.e., lanes on which the PHY LTSSM detected the presence of a link partner in the Detect state) are considered used. All other lanes associated with the port are unused.[1]
- Used lanes are active and fully powered.
- Dynamic link width downconfigure (i.e., change of link width while the link is up) is handled per the rules in the PCI Express Base Specification. In this case, inactive lanes place their transmitter in electrical idle and enable receiver termination[2].

1. Note that unused lanes may become used when the PHY LTSSM transitions to the Detect state and retrains the link.
2. In the switch, these lanes are placed in the P1 power state.

Notes

It is possible to explicitly power-down a SerDes quad by setting the POWERDN bit in the corresponding SerDes Control (S[x]CTL) register. Refer to the definition of this field for further details. Powering-down a SerDes shared by multiple ports results in all such ports being affected. Refer to section SerDes Numbering and Port Association on page 8-1 for a list of port/SerDes associations.

Notes

Power Management

Notes

Overview

This chapter describes the PES32NT8xG2 device power management support. This chapter does not describe link active state power management (ASPM). For a description of this topic, refer to section Link Active State Power Management (ASPM) on page 7-12.

Located in the configuration space of each function in the PES32NT8xG2 (i.e., PCI-to-PCI Bridge, NT, and DMA functions) is a power management capability structure. The switch's functions support the following device power management states:

- D0 ($D0_{uninitialized}$ and $D0_{active}$)
- $D3_{Hot}$
- $D3_{Cold}$.

A power management state transition diagram for the states supported by the switch is provided in Figure 9.1 and described in Table 9.1. Transitioning a function's power management state from $D3_{hot}$ to $D0_{uninitialized}$ does not result in any logic being reset or re-initialization of register values. Thus, the default value of the No Soft Reset (NOSOFTRST) bit in the function's PCI Power Management Control and Status (PMCSR) register corresponds to the functional context being maintained in the $D3_{hot}$ state.

The power management capability structure associated with each function affects the power state of that function only. The link's power state is derived from the power state of the function(s) in the port. When a function enters the D0 state (i.e., $D0_{uninitialized}$ or $D0_{active}$), the function transitions the port's link to the L0 state. When a function enters the $D3_{Hot}$ state, the function transitions the port's link to the L1 state.

- When the upstream port operates in a multi-function mode, the port's link enters the L1 power state only when all functions in the port are placed in a non-D0 state. For example, a port operating in upstream switch port with NT function mode places its link in the L1 state when both functions of the port are placed in the $D3_{Hot}$ state. A port operating in upstream switch port with NT and DMA functions places its link in the L1 state when all three functions of the port are placed in the $D3_{Hot}$ state.
- A port configured in 'Downstream Switch Port' mode always accepts entry into L1 when requested by the link partner.[1]

[1] This applies when entry into L1 is a result of the link partner being placed in D3hot. This does not apply for entry into L1-ASPM, where the L1 entry request may be rejected by the downstream switch port. Refer to section Link Active State Power Management (ASPM) on page 7-12 for further details on L1 ASPM.

Notes

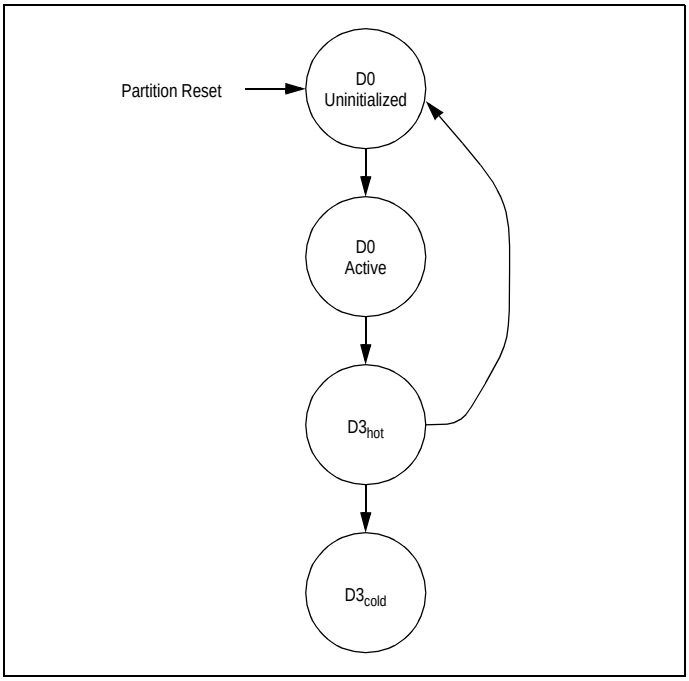

Figure 9.1 PES32NT8xG2 Power Management State Transition Diagram

From State	To State	Description
any	D0 Uninitialized	Partition reset (any type).
D0 Uninitialized	D0 Active	Function configured by software
D0 Active	$D3_{hot}$	The Power Management State (PMSTATE) field in the PCI Power Management Control and Status (PMCSR) register is written with the value that corresponds to the $D3_{hot}$ state.
$D3_{hot}$	D0 Uninitialized	The Power Management State (PMSTATE) field in the PCI Power Management Control and Status (PMCSR) register is written with the value that corresponds to D0 state.
$D3_{hott}$	$D3_{cold}$	Power is removed from the device.

Table 9.1 PES32NT8xG2 Power Management State Transition Diagram

PCI-to-PCI bridge functions have the following behavior when in the $D3_{hot}$ power management state:

- The bridge accepts, processes and completes all type 0 configuration read and write requests.
- Accepts and processes all message requests that target the bridge.
  - Vendor Defined Type 1 messages are silently dropped.
- All requests received by the bridge on the primary interface, except as noted above, are treated as unsupported requests (UR).
  - Vendor Defined Type 1 messages are silently dropped.

Notes

- Any error message resulting from the reception of a TLP is reported in the same manner as when the bridge is not in $D3_{hot}$ (e.g, generation of an ERR_NONFATAL message to the root).
  - This requires transitioning the link to the L0 state when error reporting is enabled and the link is not in L0.
- Error messages resulting from any event other than the reception of a TLP are discarded (i.e., no error message is generated).
- All received completions that target the bridge are treated as unexpected completions (UC).
- Completions flowing in either direction through the bridge are routed as normal. This behavior of the bridge does not differ from that of the bridge when it is in the D0 power management state.
  - This requires transitioning the link to the L0 state when the completion needs to be transmitted on the link by the bridge function and the link is not in L0.
  - There is not need to transition the link to the L0 state when the completion is not transmitted on the link by the bridge function (e.g., when the completion flows from the PCI-to-PCI bridge function to the NT function in a port configured in 'Upstream Switch Port with NT Endpoint' mode).
- All request TLPs received on the secondary interface are treated as unsupported requests (UR).

NT functions have the following behavior when in the $D3_{hot}$ power management state.

- The function accepts, processes and completes all type 0 configuration read and write requests.
- Accepts and processes all message requests that target the function.
- All requests received by the function, except as noted above, are treated as unsupported requests (UR).
- Any error message resulting from the receipt of a TLP is reported in the same manner as when the function is not in $D3_{hot}$ (e.g, generation of an ERR_NONFATAL message to the root).
  - This requires transitioning the link to the L0 state when error reporting is enabled and the link is not in L0.
- Error messages resulting from any event other than the receipt of a TLP are discarded (i.e., no error message is generated).
- All received completions whose destination ID match the NT function's bus/dev/function are treated as unexpected completions (UC).
- All received completions transferred on the NT interconnect are routed as normal.
- All request TLPs received by another NT function that target the partition associated with the NT function in $D3_{hot}$ are treated as unsupported requests (UR) by the NT function that receives the TLP.
- All completion TLPs received by another NT function that target the partition associated with the NT function in $D3_{hot}$ are silently dropped.
- Note that it is the responsibility of software (e.g., the NT driver) to ensure that inter-partition communication associated with the NT function is properly quiesced prior to placing this function into $D3_{hot}$.

DMA functions have the following behavior when in the $D3_{hot}$ power management state.

- The function accepts, processes and completes all type 0 configuration read and write requests.
- Accepts and processes all message requests that target the function.
- All requests received by the function, except as noted above, are treated as unsupported requests (UR).
- Any error message resulting from the receipt of a TLP is reported in the same manner as when the function is not in $D3_{hot}$ (e.g, generation of an ERR_NONFATAL message to the root).

Notes

- This requires transitioning the link to the L0 state when error reporting is enabled and the link is not in L0.

– Error messages resulting from any event other than the receipt of a TLP are discarded (i.e., no error message is generated).

– All received completions that target the DMA function are treated as unexpected completions (UC).

– Note that the DMA function does not automatically quiesce traffic as a result of being placed in $D3_{hot}$. It is the responsibility of software (e.g., the DMA driver) to ensure that the DMA function is properly quiesced prior to entry into $D3_{hot}$.

Power Management Event (PME) Messages

The PES32NT8xG2 does not support generation of PME messages from the $D3_{cold}$ state. PME message generation is only supported by downstream switch ports (i.e., the PCI-to-PCI bridge, NT function, and DMA function associated with an upstream port do not support PME message generation).

Downstream switch ports support the generation of hot-plug PME events (i.e., a PM_PME power management message) from the $D3_{hot}$ state (see section Hot-Plug Events on page 11-7). This includes both the case when the downstream switch port is in the $D3_{hot}$ state or the entire switch partition is in the $D3_{hot}$ state.

PCI Express Power Management Fence Protocol

The root complex takes the following steps to turn off power to a system.

– The root places all devices in the D3 state

– Upon entry to D3, all devices transition their links to the L1 state

– The root broadcasts a PME_Turn_Off message.

- The links temporarily transition to L0 in order to transfer the message.

– Devices acknowledge the PME_Turn_Off message by returning a PME_TO_ACK message

- After transmitting a PME_TO_ACK, a device places its link in L2/L3-Ready state.

The PME_Turn_Off / PME_TO_Ack protocol may be initiated by the root when the switch function's are in any power management state. The port's handling of the power management fence protocol depends on its operating mode as described below.

Upstream Switch Port or Downstream Switch Port Mode

– When a port configured in upstream switch port mode receives a PME_Turn_Off message, it broadcasts the PME_Turn_Off message on all downstream switch ports of the associated partition. The upstream port transmits a PME_TO_Ack message and transitions its link state to L2/L3 Ready after it has received a PME_TO_Ack message on each of the downstream switch ports of the partition. This process is called PME_TO_Ack aggregation.

- In the PES32NT8xG2, the link is considered down in the L2/L3 Ready state (see section Link States on page 7-9). This allows a power management turn-off event in a partition to be signaled via the event signaling mechanism (see Chapter 16, Switch Events).

– The aggregation of PME_TO_Ack messages on downstream switch ports is abandoned by the upstream switch port when this port receives a TLP after having previously received a PME_Turn_Off message but before having responded with a PME_TO_Ack message. Once a PME_TO_Ack message has been scheduled for transmission on the upstream switch port and the PME_TO_Ack aggregation process has completed, received TLPs at that point are discarded.

– If the TLP that causes PME_TO_Ack aggregation to be abandoned targets a PES32NT8xG2 function, then the targeted function responds to the TLP normally. If the TLP that causes aggregation to be abandoned is routed to a downstream switch port's link and the link is in L0, then the TLP is transmitted on the downstream switch port. If the downstream switch port's link is not in L0 (e.g., it is in L2/L3 Ready), then the port transitions the link to Detect and then to L0. Once the link is reaches L0, the TLP is transmitted on the downstream switch port.

Notes

– When PME_TO_Ack aggregation is abandoned, the PES32NT8xG2 makes no attempt to abandon the PME_Turn_Off and PME_TO_Ack protocol on downstream switch ports. Devices downstream of the switch are allowed to respond with a PME_TO_Ack and transition to L2/L3 Ready. When a TLP is received that needs to be routed to the downstream switch port's link, then the switch transitions the link to Detect and then to L0. Once the link reaches L0, the TLP is transmitted on the downstream switch port.

NT Function Mode or NT with DMA Function Mode

When a port configured in NT function mode or NT with DMA function mode receives a PME_Turn_Off message, it immediately generates a PME_TO_Ack message for transmission on its link. After transmitting the PME_TO_Ack message, the port transitions its link state to L2/L3 Ready.

- In the PES32NT8xG2, the link is considered down in the L2/L3 Ready state (see section Link States on page 7-9). This allows a power management turn-off event in a partition to be signaled via the event signaling mechanism (see Chapter 16, Switch Events).

All TLPs received by the port after the PME_Turn_Off message is received (before the link is placed in L2/L3 Ready) are discarded. Note that it is the responsibility of software (e.g., the NT driver) to ensure that inter-partition communication associated with the NT function is properly quiesced prior to initiating the PME_Turn_Off / PME_TO_Ack handshake.

Finally, note that the DMA function does not automatically quiesce traffic as a result of the PME_Turn_Off / PME_TO_Ack handshake on the port. It is the responsibility of software (e.g., the DMA driver) to ensure that the DMA function is properly quiesced prior to initiating the PME_Turn_Off / PME_TO_Ack handshake.

Upstream Switch Port with NT and/or DMA Function Mode

When a port configured in upstream switch port with NT function mode, upstream switch port with DMA function mode, or upstream switch port with NT and DMA function mode receives a PME_Turn_Off message, the port transmits a PME_TO_Ack message when the PCI-to-PCI bridge function has completed PME_TO_Ack aggregation, as described in section Upstream Switch Port or Downstream Switch Port Mode on page 9-4 above.

After the PME_Turn_Off message is received, but before the PME_TO_Ack message has been scheduled for transmission, the reception of a TLP by the port causes PME_Turn_Off aggregation to be abandoned. If the TLP that causes PME_TO_Ack aggregation to be abandoned targets a switch function (i.e., PCI-to-PCI bridge, NT, or DMA), then the targeted function responds to the TLP normally.

Once the PME_TO_Ack message has been scheduled for transmission by the port, the port discards all received TLPs. After transmitting the PME_TO_Ack message, the port transitions its link state to L2/L3 Ready. In the PES32NT8xG2, the link is considered down in the L2/L3 Ready state (see section Link States on page 7-9). This allows a power management turn-off event in a partition to be signaled via the Event Signaling mechanism described in Chapter 16, Switch Events.

Notes

Transparent Switch Operation

Notes

Overview

As noted in Chapter 1, each PES32NT8xG2 switch partition operates logically as a completely independent PCI Express switch that implements the behavior and capabilities required of a switch by the PCI Express Base Specification Revision 2.1.

- A PCI Express switch contains one upstream port and one or more downstream switch ports. Each port is associated with a PCI-to-PCI bridge function. All PCI-to-PCI bridges associated with a PCI Express switch are interconnected by a virtual PCI bus.
- In this document, the term *transparent switch operation* refers to the operation of such a switch.

This chapter describes switch-specific architectural features for the transparent switch associated with each switch partition (i.e., the PCI-to-PCI bridge functions and their interaction in the switch). As discussed in Chapter 1, the upstream port of a switch partition may be configured to include a Non-Transparent Bridge (NTB) function (for inter-partition communication), as well as a DMA function. The operation of these functions is not discussed in this chapter.

- For details on non-transparent operation, refer to Chapter 14.
- For details on the DMA operation, refer to Chapter 15.

Transaction Routing

The PES32NT8xG2 PCI-to-PCI bridge functions support routing of all transaction types defined in PCI Express Base Specification Revision 2.1. This includes routing of specification defined transactions as well as those that may be used in vendor defined messages and in future revisions of the PCI Express Base Specification. Specifically, the PCI-to-PCI bridge function supports the following type of routing:

- Address routing with 32-bit or 64-bit format
- ID based routing using bus, device and function numbers.
- Implicit routing utilizing
  - Route to root
  - Broadcast from root
  - Local - terminate at receiver
  - Gathered and routed to root

A summary of TLP types that use the above routing methods is provided in Table 10.1.

Routing Method	TLP Type Using Routing Method
Route by Address	MRd, MrdLk, MWr, IORd, IOWr, Msg, MsgD
ID Based Routing	CfgRd0, CfgWr0, CfgRd1, CfgWr1, TCfgRd, TCfgWr, Cpl, CpdD, CplLk, CplDLk, Msg, MsgD
Imlicit Routing - Route to Root	Msg, MsgD
Implicit Routing - Broadcast from Root	Msg, MsgD
Implicit Routing - Local	Msg, MsgD
Implicit Routing - Gathered and Routed to Root[1]	Only supported for PME_TO_Ack messages in response to a root initiated PME_Turn_Off message.

Table 10.1 Switch Routing Methods

1. The only Gathered and Routed to Root message supported is a PME_TO_Ack message received on a downstream switch port.

Notes

Virtual Channel Support

In section Virtual Channel Support on page 4-5 there is a description of virtual channel support in the PES32NT8xG2 ports. The PCI-to-PCI bridge function contains a VC Capability Structure that provides architected port arbitration and TC/VC mapping for VC0. For port operating modes in which the PCI-to-PCI bridge function is function 0 of the port, the VC Capability Structure in this function provides architected port arbitration and TC/VC mapping for all functions of the port. For other port operating modes, the registers in the PCI-to-PCI bridge function's VC Capability Structure are 'reserved' [1] and must not be programmed.

Maximum Payload Size

The PES32NT8xG2 requires that the Maximum Payload Size (MPS) field in the PCI Express Device Control (PCIEDCTL) register be set identically in all functions (i.e., PCI-to-PCI bridge, NT, and DMA) of a partition.

Note that a port with a maximum link width of x1 supports a Maximum Payload Size (MPS) of up to 1 KB. Ports with maximum link width of x2, x4, or x8 support an MPS of up to 2 KB. The MPAYLOAD field in the PCI Express Device Capabilities (PCIEDCAP) register is automatically set by the hardware based on the port's maximum link width to reflect this.

Upstream Port Device Number

In the switch, the upstream port of a partition is assumed to have device number zero. Type 0 configuration requests received by the upstream port must always target device 0 in the port. In order to meet this requirement, Alternative Routing ID (ARI) Forwarding must be disabled in the root port or switch downstream port immediately above the switch partition's upstream port.

Bus Locking

The switch supports locked transactions, allowing legacy software to run without modification on PCI Express. Locked transactions are only supported between an upstream switch port (i.e., PCI-to-PCI bridge function) and a downstream switch port in the same partition. Only one locked transaction sequence may be in progress at a time.

- A locked transaction sequence is requested by the root complex by issuing a Memory Read Request - Locked (MRdLk) transaction. A lock is established when a lock request is successfully completed with a Completion with Data - Locked (CplDLk). A lock is released with an Unlock message (Msg) sent by the root complex.

When the switch receives a MRdLk transaction on a partition's upstream switch port, it forwards the MRdLk transaction to the appropriate downstream switch port and locks the downstream switch port so that all subsequent TLPs destined to the locked port from other ports (except the upstream port) are blocked until the lock is released.

- Bus locking only affects TLPs that map to VC0 at the egress port. TLPs that do not map to VC0 are not affected by the lock.[2]
- The MRdLK transaction obeys PCI Express ordering rules meaning that all queued posted requests for the downstream switch port are transmitted prior to the MRdLK being transmitted. The MRdLK is allowed by bypass queued completions.[3]
- Locking of a downstream switch port does not affect transactions destined to any other port (e.g., transactions from the other downstream switch ports to the upstream port and peer-to-peer transactions among other downstream switch ports are not blocked).

1. Reading from a reserved address returns and undefined value. Writes to a reserved address complete successfully but produce undefined behavior on the register.
2. In the PES32NT8xG2, only VC0 is supported. TLPs that don't map to VC0 are treated as malformed. Refer to section Error Detection and Handling by the PCI-to-PCI Bridge Function on page 10-11.
3. Refer to section Packet Ordering on page 4-6 for further details on ordering rules.

Notes

When a CplDLk is received by the locked downstream switch port, it forwards the CplDLk transaction to the upstream port and locks the upstream port so that all subsequent TLPs destined to the locked port from other ports (except the locked downstream switch port) are blocked until the lock is released.

- Bus locking only affects TLPs that map to VC0 at the egress port. TLPs that do not map to VC0 are not affected by the lock.
- The CplDLk transaction obeys PCI ordering rules meaning that all queued posted requests at the locked downstream switch port destined to the upstream port are completed prior to the CplDLk being transmitted. The CplDLk is allowed to bypass queued non-posted requests.

When a CplDLk is returned by the locked downstream switch port and the upstream port becomes locked, the partition is said to be 'bus-locked'. While a partition is bus-locked, the following applies:

- It is illegal to read or write any of the PCI Express configuration space headers in ports associated with the partition since the switch can not generate a completion until the partition is unlocked. The behavior of the partition is undefined when a partition's PCI Express configuration space register is read while the partition is bus-locked.
- Any register in the ports associated with the partition may be read or written via the SMBus.
- It is allowed for the root to perform subsequent reads from the locked device (e.g., a legacy endpoint) by issuing a MRdLk requests to the locked device and receiving a CplDLk or CplLk response from the locked device. These transactions do not change the state of the bus-locked partition. Therefore, a CplLk completion received by the downstream switch port of a bus-locked partition in no way "unlocks" the partition.
- It is allowed for the root to perform subsequent writes to the locked device by issuing MWr requests to the locked device. These transactions in no way change the state of the bus-locked partition.
- The locked upstream and downstream switch ports may generate messages (i.e., "insert messages"). These messages include interrupt emulation messages and error messages. The locked ports may also generate MSIs.

The behavior of a bus-locked partition is undefined when:

- Any transaction other than a MWr, MRdLk, and Unlock message is received on the upstream port.
- Any transaction other than a CplLk and a CplDLk is received on the locked downstream switch port.
- A MRdLk TLP is received on the partition's upstream port destined to an unlocked downstream switch port.
- A TLP is received by the upstream port destined to an unlocked downstream switch port.

When an Unlock message is received on the partition's upstream port, the partition is unlocked. This causes the Unlock message to be forwarded to the locked downstream switch port and the unblocking of transactions destined to the previously locked ports.

- The unlock message obeys PCI ordering rules meaning that all queued posted requests from the upstream port are completed prior to the switch becoming unlocked.
- Unlocked ports ignore the reception of the unlock message.

Note that when a TLP received by a port is blocked from being forwarded due to a bus-locked partition, the TLP is delayed until the partition is unlocked. If the partition is locked for an extended period, this may cause TLPs to be discarded due to switch time-outs.

Notes

Interrupts

The switch's PCI-to-PCI bridge functions may be configured to issue interrupts due to several conditions. The interrupt sources each have a corresponding status bit in the PCI-to-PCI bridge function's Interrupt Status (P2PINTSTS) register.

- When an interrupt source requests service, the corresponding bit in the P2PINTSTS register is set.
- An interrupt source may be masked from generating an interrupt by setting the corresponding mask bit in the PCI-to-PCI Bridge Interrupt Mask (P2PINTMSK) register. By default, all interrupt sources are masked.
- Once a bit corresponding to an interrupt source is set in the P2PINTSTS register, interrupts associated with that source are inhibited until the bit is cleared in the P2PINTSTS register.

When a PCI-to-PCI bridge function detects the occurrence of an unmasked interrupt condition, an MSI or legacy interrupt message is generated by the function per the rules in Table 10.2. The removal of the interrupt condition occurs when unmasked status bit(s) causing the interrupt are masked or cleared.

The PES32NT8xG2 assumes that MSIs generated by the PCI-to-PCI bridge function target the root-complex and always routes these transactions to the partition's upstream link. Configuring the address contained in the PCI-to-PCI bridge function's MSIADDR and MSIADDRU registers to an address that does not route to the partition's upstream link and generating an MSI produces undefined results.

- An MSI generated by the PCI-to-PCI bridge function is never multicasted. Software must never configure the address of an MSI generated by a PCI-to-PCI bridge function to fall within an enabled multicast BAR aperture in the partition. Violating this requirement produces undefined results.

Unmasked Interrupt	EN bit in MSICAP Register	INTXD bit in PCICMD Register	Action
Asserted	1	X	MSI message generated
	0	0	Assert_INTA message request generated
	0	1	None
Negated	1	X	None
	0	0	Deassert_INTA message request generated
	0	1	None

Table 10.2 PCI-to-PCI Bridge Function Interrupts

Downstream Port Interrupts

The following are sources of downstream switch port interrupts and MSIs.

- Downstream switch port's hot-plug controller.
- Link bandwidth notification capability (i.e., assertion of the LBWSTS or LABWSTS bits in the PCIELSTS register when interrupt notification is enabled for these bits).

When a port is configured to generate INTx messages, only INTA is used. Note that the Interrupt Pin register (INTRPIN) must be programmed accordingly.

Upstream Port Interrupts

The following are sources of upstream port interrupts and MSIs.

- Switch events
- Failover change initiated by the failover capability associated with the partition
- Failover change completed by the failover capability associated with the partition
- A temperature sensor alarm (see Chapter 19).

Notes

When a port is configured to generate INTx messages, only INTA is used. Note that the Interrupt Pin register (INTRPIN) must be programmed accordingly.

The MSI capability structure associated with the upstream port's PCI-to-PCI bridge function is not by default part of the PCI capability structure linked-list located in the function's configuration space. This capability may be added to the capability structure linked-list by using the serial EEPROM, SMBus, or the Root to unlock registers and setting the Next Pointer (NXTPTR) field in one of the linked capabilities to point to the MSI capability structure.

Legacy Interrupt Aggregation

Each switch partition supports legacy PCI INTx emulation. Rather than use sideband INTx signals, PCI Express defines two messages that indicate the assertion and negation of an interrupt signal. An Assert_INTx message is used to signal the assertion of an interrupt signal and an Deassert_INTx message is used to signal its negation.

Within each partition, the PES32NT8xG2 maintains an aggregated INTx state for each of the four interrupt signals (i.e., A through D) at each port. The aggregation includes INTx interrupts generated by all downstream ports of the partition, as well as INTx interrupts generated by the upstream port's PCI-to-PCI bridge, NT, and DMA functions. Figure 10.1 shows a logical diagram of the INTx aggregation for a sample partition configuration. In this example, the downstream and upstream port PCI-to-PCI bridge functions are configured to generate INTA interrupts, while the NT and DMA functions are each configured to generate INTD interrupts.

As shown in Figure 10.1, the upstream port PCI-to-PCI bridge function re-maps the INTx interrupts from the partition's downstream ports (see Table 10.3 below), and logically "ORs" the INTA interrupt result from the re-mapping logic with the INTA interrupt generated by the function itself. In addition, interrupt aggregation logic at the multi-function upstream port aggregates the INTx interrupts issued by each of the port functions.

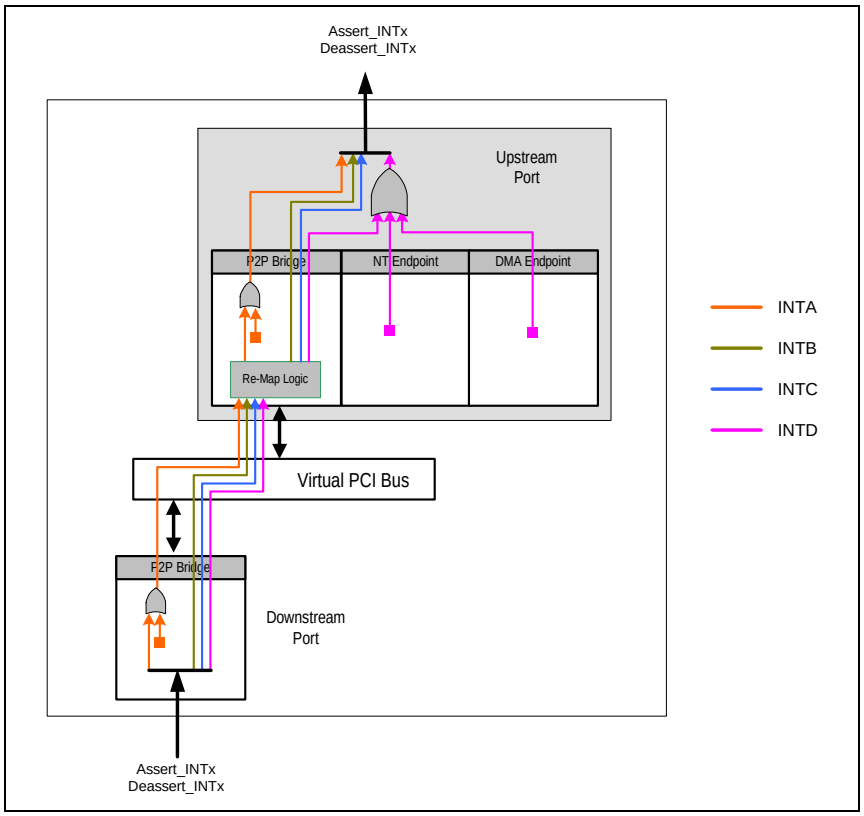

Figure 10.1 Logical Representation of INTx Aggregation

Notes

An Assert_INTx message is sent to the root by the upstream port when the aggregated state of the corresponding interrupt in the upstream port transitions from a negated to an asserted state. A Deassert_INTx message is sent to the root by the upstream port when the aggregated state of the corresponding interrupt in the upstream port function transitions from an asserted to a negated state. The requester ID field in the Assert_INTx and Deassert_INTx messages has the upstream port's bus and device number. The function number is set to 0x0 and must be ignored by receivers per the PCI Express Base Specification.

PCI-to-PCI bridges must map interrupts on the secondary side of the bridge according to the device number of the device on the secondary side of the bridge. No mapping is performed for the PCI-to-PCI bridges corresponding to downstream switch ports as these ports only connect to device zero. A mapping is performed for the upstream port. This mapping is summarized in Table 10.3.

	Upstream Port Interrupt			
	INTA	INTB	INTC	INTD
Downstream Device[1] Interrupt	Device (N mod 4) = 0 INTA	Device (N mod 4) = 0 INTB	Device (N mod 4) = 0 INTC	Device (N mod 4) = 0 INTD
	Device (N mod 4) = 1 INTD	Device (N mod 4) = 1 INTA	Device (N mod 4) = 1 INTB	Device (N mod 4) = 1 INTC
	Device (N mod 4) = 2 INTC	Device (N mod 4) = 2 INTD	Device (N mod 4) = 2 INTA	Device (N mod 4) = 2 INTB
	Device (N mod 4) = 3 INTB	Device (N mod 4) = 3 INTC	Device (N mod 4) = 3 INTC	Device (N mod 4) = 3 INTA

Table 10.3 Downstream to Upstream Port Interrupt Routing Based on Device Number

1. Device X INTy corresponds to external downstream generated INTy interrupts and INTy interrupts generated by the downstream port.

If a downstream switch port goes down (i.e., transition to DL_Down state) or is removed from a partition, the INTx virtual wires associated with that port are negated, and the upstream port's aggregate state is updated accordingly. This may result in the upstream port generating a Deassert_INTx message. Refer to section Port Operating Mode Change on page 5-13 for details on how port operating mode changes affect the aggregated INTx state of a partition.

Access Control Services

The PCI-to-PCI bridge function supports Access Control Services (ACS) as defined in PCI Express Base Specification. ACS functionality is performed by the PCI-to-PCI bridge function when the associated port operates in the following modes:

- Downstream switch port mode
- Upstream switch port with NT function
- Upstream switch port with DMA function
- Upstream switch port with NT and DMA functions

The PCI-to-PCI bridge function does not support ACS checks when the port operates in any other mode. When a port operates in downstream switch port mode, the PCI-to-PCI bridge function supports the following ACS operations:

- ACS Source Validation
- ACS Translation Blocking
- ACS Peer-to-Peer[1] Request Redirect

1. For a port operating in downstream switch port mode, 'peer-to-peer' implies traffic received by the downstream switch port (via the PCI Express link) that is destined towards another downstream switch port in the same partition.

Notes

- ACS Peer-to-Peer Completion Redirect
- ACS Upstream Forwarding
- ACS Peer-to-Peer Egress Control
- ACS Direct Translated Peer-to-Peer

When a port operates in one of the multi-function upstream port modes listed above, the PCI-to-PCI bridge function supports the following ACS operations[1]:

- ACS Peer-to-Peer[2] Request Redirect
- ACS Peer-to-Peer Completion Redirect
- ACS Direct Translated Peer-to-Peer

ACS is programmed via the ACS Capability Structure in the PCI-to-PCI bridge function's configuration space.

The PCI-to-PCI bridge function only applies ACS checks to TLPs flowing in the upstream direction.

- For a downstream switch port, these are TLPs that are received from the port's link, regardless of the final destination of the TLP (e.g., regardless of whether the TLP is going to the upstream port or a peer downstream switch port).
- For a multi-function upstream port, these are TLPs received by the port's PCI-to-PCI bridge function on its secondary side.

When an ACS check causes a TLP to be re-directed, the re-direction is implemented such that TLPs received by a port that are ACS re-directed follow the ordering rules described in section Packet Ordering on page 4-6. The following figures show examples of the effect of applying ACS checks to TLPs on the PCI-to-PCI bridge function.

Figure 10.2 shows an example of ACS source validation at a downstream switch port. In this case, the offending TLP is dropped and a completion with completer-abort status is generated.

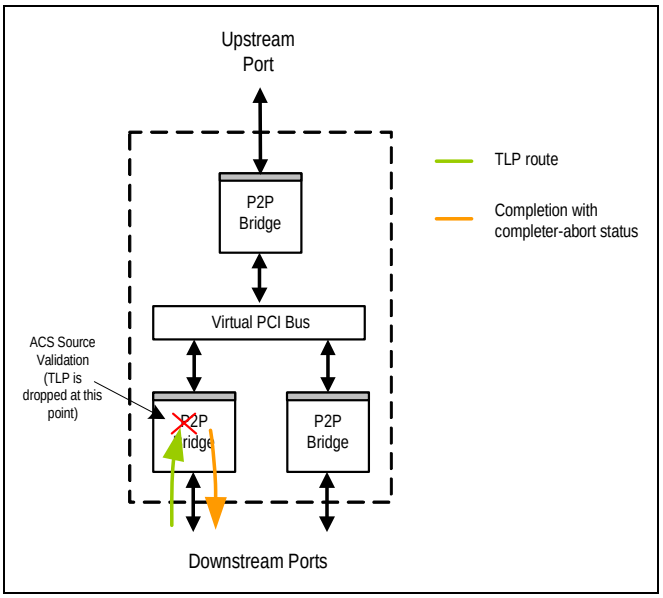

Figure 10.2 ACS Source Validation Example

Figure 10.3 shows an example of ACS peer-to-peer request re-direct at a downstream switch port. In this case, the offending TLP received by the downstream switch port is re-directed towards the root-complex.

1. Note that the switch does not support ACS Peer-to-Peer Egress Control among the functions of a multi-function upstream port.
2. For a port operating in a multi-function upstream port mode, 'peer-to-peer' implies traffic sent from one of the port functions to another (e.g., from the port's PCI-to-PCI bridge function to the port's NT function or vice-versa).

Notes

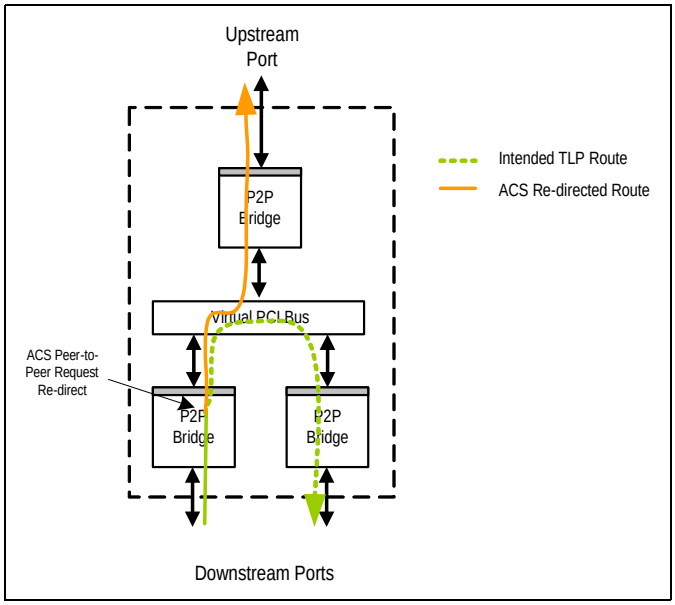

Figure 10.3 ACS Peer-to-Peer Request Re-direct at a Downstream Switch Port

Figure 10.4 shows an example of ACS upstream forwarding at a downstream switch port. As with ACS Peer-to-Peer forwarding, the offending TLP received by the downstream switch port is re-directed towards the root-complex.

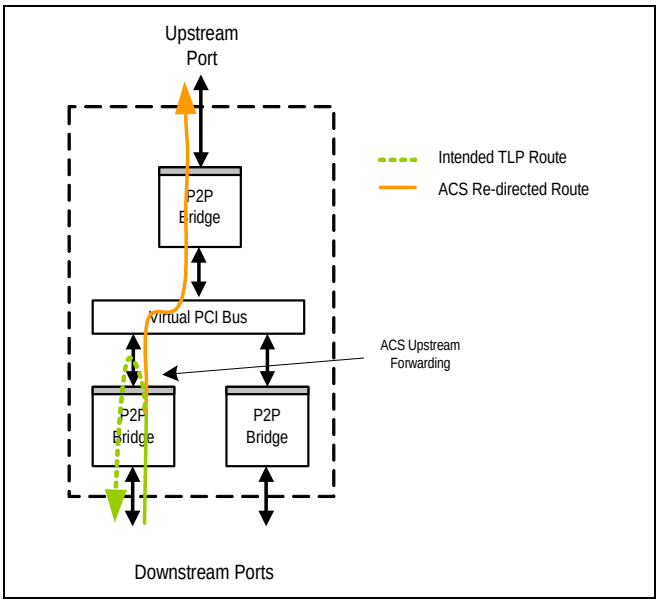

Figure 10.4 ACS Upstream Forwarding Example

Finally, Figure 10.5 shows an example of ACS peer-to-peer request re-direct at the PCI-to-PCI bridge function of a multi-function upstream port. As shown, the offending TLP received by the PCI-to-PCI bridge function on its secondary side is re-directed towards the root-complex.

Notes

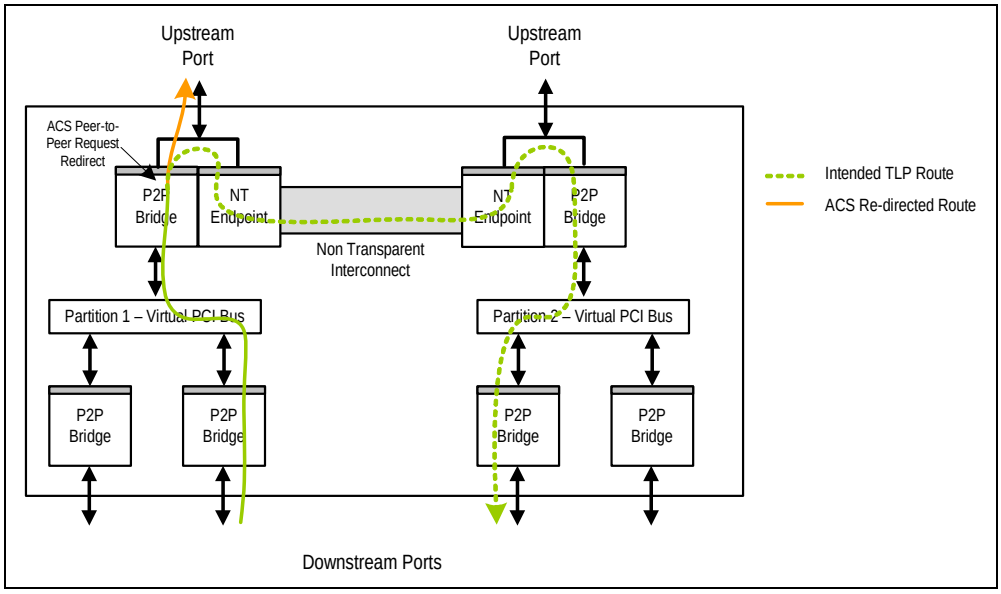

Figure 10.5 ACS Peer-to-Peer Request Re-direct by an Upstream PCI-to-PCI Bridge Function

When multiple ACS checks are enabled in the PCI-to-PCI bridge function, they are prioritized as described below. Table 10.4 shows the prioritization for ACS checks associated with the reception of request TLPs. Table 10.5 shows the prioritization for ACS checks associated with the reception of completion TLPs.

ACS Check	Priority	Comment
ACS Source Validation	4 (Highest)	Applicable to request TLPs received by a downstream switch port on its ingress link.
ACS Translation Blocking	3	Applicable to memory request TLPs received by a downstream switch port on its ingress link.
ACS Upstream Forwarding	2	Applicable to request or completion TLPs received by the downstream switch port on its ingress link that target the port's egress link. This is not considered a peer-to-peer transfer.
ACS Peer-to-Peer Request Redirect	1 (Lowest)	Applicable to peer-to-peer request TLPs only Subject to the interaction rules in Section 6.12.3 of the PCI Express Base Specification.
ACS Peer-to-Peer Egress Control		
ACS Direct Translated Peer-to-Peer		

Table 10.4 Prioritization of ACS Checks for Request TLPs

Notes

ACS Check	Priority	Comment
ACS Upstream Forwarding	2 (Highest)	Applicable to request or completion TLPs received by the downstream switch port on its ingress link that target the port's egress link. This is not considered a peer-to-peer transfer.
ACS Peer-to-Peer Completion Redirect	1 (Lowest)	Applicable to non-relaxed-ordered peer-to-peer completion TLPs only

Table 10.5 Prioritization of ACS Checks for Completion TLPs

ACS checks are only applicable to certain TLP types. Table 10.6 list the ACS checks supported by the PCI-to-PCI bridge function and the TLP types on which they are applied.

ACS Check	Applicable to the Following TLP Type(s)
ACS Source Validation	Request TLPs
ACS Translation Blocking	Memory Request TLPs
ACS Peer-to-Peer (P2P) Request Redirect	Peer-to-Peer Request TLPs
ACS P2P Completion Re-direct	Peer-to-Peer Completion TLPs
ACS Upstream Forwarding	Request or Completion TLPs that target the port's own egress link
ACS P2P Egress Control	Peer-to-Peer Request TLPs
ACS Direct Translated P2P	Peer-to-Peer Memory Request TLPs

Table 10.6 TLP Types Affected by ACS Checks

ACS violations associated with transparent operation are handled as described in section ACS Error Handling on page 10-18. Refer to PCI Express Base Specification for further information on ACS.

ECRC Support

The PCI-to-PCI bridge function supports End-to-End CRC (ECRC) generation and checking. ECRC checking is done for TLPs that are received by the PCI-to-PCI bridge on the port's link, and that either target the bridge function or are forwarded across the bridge. The PCI-to-PCI bridge function only checks and logs ECRC errors when the ECRC Check Enable (ECRCCE) bit is set in the function's AER Control (AERCTL) register.

- ECRC error checking and logging is not performed by PCI-to-PCI bridge functions that do not receive the TLP from the link.
- The PCI-to-PCI bridge function never modifies the ECRC on received TLPs that it forwards across the bridge.
- When ECRC checking is enabled, the reception of a TLP without ECRC is not considered an error (i.e., the TLP is processed normally).
- If the port is operating in a multi-function mode, then ECRC errors are only logged in functions in which ECRC checking is enabled.
- Refer to section Error Detection and Handling by the PCI-to-PCI Bridge Function on page 10-11 for details on the logging and signaling of ECRC errors in the PCI-to-PCI bridge function.

ECRC generation is enabled in the PCI-to-PCI bridge function when the ECRC Generation Enable (ECRCGE) bit is set in the function's AER Control (AERCTL) register. If ECRC generation is enabled in a PCI-to-PCI bridge function, then all TLPs originated by that function contain an ECRC. Otherwise, all TLPs originated by that function do not have ECRC.

Notes

Error Detection and Handling by the PCI-to-PCI Bridge Function

This section describes error conditions detected by the PCI-to-PCI bridge function. This includes physical, data-link, and transaction layer errors detected by the port, as well as routing errors associated with the PCI-to-PCI bridge function in the port.

– Internal switch errors (i.e., parity errors, switch time-out, and internal memory errors) are associated with the switch core and not with a specific port function. These errors are not described here. Refer to section Internal Errors on page 4-16 for a detailed description of these errors.

The errors described here apply to ports that operate in a mode that includes a PCI-to-PCI bridge function (e.g., upstream switch port mode, downstream switch port mode, upstream switch port with NT function mode, etc.) This section focuses specifically on errors related to the PCI-to-PCI bridge function[1]. Errors that affect all functions of the port (i.e., non function-specific errors) are noted where appropriate.

Error detection and handling in the switch follows the requirements in PCI Express Base Specification. The error checking and handling described here is performed by each PES32NT8xG2 PCI-to-PCI bridge function. In cases where the error condition propagates among PCI-to-PCI bridge functions (e.g., a poisoned TLP flowing from the upstream port to a downstream switch port), each PCI-to-PCI bridge function performs error checking and handling independently.

The errors described below are associated with specific actions to log and report the error. The terms ‘uncorrectable error processing’ and ‘correctable error processing’ refer to the processing described in Section 6.2.5 of PCI Express Base Specification. Errors that are not function-specific are logged in the corresponding status and logging registers of all functions in the port. Errors that are function-specific are logged in the status and logging register of the affected function. Signaling of non function-specific errors follows the rules in Section 6.2.4 of PCI Express Base Specification.

Some of the errors described below are marked as function-specific when the “function claims the TLP”. A function claims a TLP in the following cases:

– PCI-to-PCI Bridge function

  Address Routed TLPs: If received on the primary side of the bridge, the TLPs address falls within the address space range(s) programmed in the base/limit registers. If received on the secondary side of the bridge, always.

  ID Routed TLPs: If received on the primary side of the bridge, the TLPs destination ID matches the bus aperture range programmed in the primary/secondary/subordinate registers or matches the bridge function’s bus/device/function assignment. If received on the secondary side of the bridge, always.

  Implicit Route TLPs: Always.

– NT Endpoint function:

  Refer to section Error Detection and Handling by the NT Function on page 14-25.

– DMA function:

  Refer to section PCI Express Error Handling by the DMA Function on page 15-28.

Physical Layer Errors

Table 10.7 lists error checks performed by the physical layer and the action taken when an error is detected. Physical layer errors affect all functions of the port.

1. Errors associated with the NT function (i.e., non-transparent operation errors) are described in Chapter 14. Errors associated with the DMA function are described in Chapter 15.

Notes

Error Condition	PCI Express Base Specification[1] Section	Function-Specific Error	Action Taken
Link Errors (8b/10b, loss of symbol lock, elastic buffer overflow/underflow, lane-to-lane deskew)	4.2.4.6	No	Correctable error processing
Any TLP or DLLP framing rule violation.	4.2.2	No	Correctable error processing

Table 10.7 Physical Layer Errors

1. Refer to PCI Express Base Specification Rev. 2.1.

Data Link Layer Errors

Table 10.8 lists error checks performed by the data link layer and action taken when an error is detected. Data link layer errors affect all functions of the port. Per PCI Express Base Specification, data link layer errors are ignored in cases where the error is associated with a received packet for which the physical layer reports an error. This prevents error pollution.

Error Condition	PCI Express Base Specification[1] Section	Function-Specific Error	Action Taken
Bad TLP[2]	3.5.3.1	No	TLP discarded, Correctable error processing
Bad DLLP[3]	3.5.2.1	No	DLLP discarded, Correctable error processing
Replay time-out	3.5.2.1	No	Correctable error processing
REPLAY NUM rollover	3.5.2.1	No	Correctable error processing
DL Protocol Error[4]	3.5.2.1	No	DLLP discarded, Uncorrectable error processing
Surprise link down (refer to section Link Down Handling on page 7-10.	3.5.2.1 & 3.2.1	No	Uncorrectable error processing

Table 10.8 Data Link Layer Errors

1. Refer to PCI Express Base Specification Rev. 2.1.

2. A Bad TLP is a TLP ending in EDB with LCRC that does not match inverted calculated LCRC, or a TLP with incorrect LCRC, or a TLP received with sequence number not equal to NEXT_RCV_SEQ and this is not a duplicate TLP).

3. A bad DLLP is a DLLP with a bad LCRC.

4. A DL protocol error occurs when an ACK or NAK DLLP is received and the sequence number specified by AckNak_Seq does not correspond to an unacknowledged TLP or to the value in ACKD_SEQ.

Notes

Transaction Layer Errors

Table 10.9 lists non-ACS error checks associated with a PCI-to-PCI bridge function and the action taken when an error is detected. ACS error checks and handling are discussed in section ACS Error Handling on page 10-18. Table 10.9 indicates the conditions under which an error is reported as advisory or non-advisory in AER. This decision does not affect the logging of errors in the PCI compatible registers (e.g., PCISTS and SECSTS).

For some errors, it is necessary to determine if the function that receives the TLP is an "ultimate receiver" or "intermediate receiver". The term "ultimate receiver" refers to a port function that receives a TLP from the link, when the TLP is consumed by a function within the switch partition associated with the receiving port. The term "intermediate receiver" refers to a port function that receives a TLP from the link, when the TLP is not consumed by a function within the switch partition associated with the receiving port.

Examples of ultimate receiver functions include:

- An PCI-to-PCI bridge function that receives a TLP from the link.
- An upstream PCI-to-PCI bridge function that receives a TLP that targets the PCI-to-PCI bridge function in a downstream port in the same partition (i.e., the TLP's bus/device/function destination ID matches that or the downstream port's PCI-to-PCI bridge function).
- A downstream PCI-to-PCI bridge function that receives a TLP that targets a function in the partition's upstream port (i.e., the TLP targets the PCI-to-PCI bridge function, the NT function, or the DMA function).
- A PCI-to-PCI bridge function that receives a TLP that can't be routed to another switch port in the partition (e.g., unsupported request, unexpected completion, etc.).

Examples of intermediate receiver functions include:

- An upstream PCI-to-PCI bridge function that receives a TLP that is destined to the link associated with a downstream port in the same partition.
- A downstream PCI-to-PCI bridge function that receives a TLP that is destined to the link associated with an upstream or downstream port in the partition.

A PCI-to-PCI bridge function that receives a multicast TLP (i.e., transparent multicast) is always considered an intermediate receiver. Per PCI Express Base Specification Revision, transaction layer errors are ignored in cases where the error is associated with a received packet for which the physical or data-link layers report an error. This prevents error pollution across the stack layers. Within the transaction layer, there are error pollution rules that resolve the cases where two or more errors are detected simultaneously. Refer to section Transaction Layer Error Pollution on page 10-20 for details on transaction layer error pollution.

Notes

Error Condition	PCI Express Base Specification[1] Section	Function Specific Error	Role Based (Advisory) Error Reporting Condition	Action Taken
Poisoned TLP received	2.7.2.2	Yes	Advisory when the corresponding error is configured as non-fatal in the AERUESV register	Detected Parity Error (DPE) bit in the PCISTS or SECSTS register set appropriately. Affected packet is forwarded across the bridge function (unless the bridge function consumes the TLP, in which case the TLP is dropped by this function). If the port is the intermediate or ultimate receiver: Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing.
ECRC check failure[2]	2.7.1	No	Advisory when the corresponding error is configured as non-fatal in the AERUESV register and the port is an intermediate receiver	If the port received the TLP from the link: Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. Affected packet is forwarded across the bridge (unless the bridge function is the target of the TLP, in which case the TLP is dropped by this function).
Unsupported request	See Table 10.10 in this chapter	Yes if a function claims the TLP. Else No.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register and the request is non-posted	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. For Non-Posted unsupported requests, the function that claims the TLP generates a completion with UR status. If the request is not claimed, then function 0 generates the completion with UR status.
Completion time-out	2.8	N/A	N/A (always non-advisory)	Not applicable (the bridge never issues requests).
Completer abort	2.3.1	N/A	N/A (always non-advisory)	Not applicable. PES32NT8xG2 ports never issue completions with 'Completer Abort' status except for ACS violations. For the latter, the error is considered an ACS error and is not logged as a completer abort error.

Table 10.9 Transaction Layer Errors Associated with the PCI-to-PCI Bridge Function (Part 1 of 2)

Notes

Error Condition	PCI Express Base Specification[1] Section	Function Specific Error	Role Based (Advisory) Error Reporting Condition	Action Taken
Unexpected completion received	2.3.2	Yes if a function claims the TLP. Else No.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. The unexpected completion is dropped.
Receiver overflow	2.6.1.2	No	N/A (always non-advisory)	Uncorrectable error processing TLP is nullified.
Flow control protocol error	2.6.1	No	N/A (always non-advisory)	Not applicable. PES32NT8xG2 does not check for any flow control protocol errors.
Malformed TLP	See Table 10.12 and Table 10.13 below	No	N/A (always non-advisory)	Uncorrectable error processing TLP is nullified.
Multicast Blocked TLP	6.14.1	Yes	N/A (always non-advisory)	The Signaled Target Abort (STAS) bit is set in the PCISTS or SECSTS register if the TLP was received on the function's primary or secondary side respectively. Uncorrectable error processing TLP is nullified.
Internal Error	6.2	Yes	N/A (always non-advisory)	Refer to section Internal Errors on page 4-16.

Table 10.9 Transaction Layer Errors Associated with the PCI-to-PCI Bridge Function (Part 2 of 2)

1. Refer to PCI Express Base Specification Revision 2.1.

2. Refer to section ECRC Support on page 10-10.

Unsupported Requests

Table 10.10 lists the conditions under which the PCI-to-PCI Bridge function in the PES32NT8xG2 ports handles received requests as unsupported requests (UR).

Conditions Handled as UR	Description	PCI Express Base Specification Section
Routing Errors	Refer to section Routing Errors on page 10-23.	Numerous
Vendor Defined Type 0 message reception[1]	Vendor Defined Type 0 message which targets the PCI-to-PCI bridge function.	2.2.8.6
Messages with invalid message code	Reception of a message TLP with invalid message code that targets the switch port's PCI-to-PCI bridge function.	2.3.1

Table 10.10 Conditions Handled as Unsupported Requests (UR) by the PCI-to-PCI Bridge Function (Part 1 of 2)

Notes

Conditions Handled as UR	Description	PCI Express Base Specification Section
Poisoned IO request, memory write request, type 0 configuration write request, or message with data targeting the bridge function	Reception of a poisoned IO request, memory write request, type 0 configuration write request, or message with data (except Vendor Defined messages) that targets a switch port's PCI-to-PCI bridge function.	2.7.2.2
Function in D3Hot state	Refer to section Overview on page 9-1.	5.3.1.4.1
Downstream Switch Port Link Down	TLPs flowing downstream across a downstream switch port's PCI-to-PCI Bridge whose link is down. Such TLPs are URed by the appropriate downstream switch port.[2]	2.9.1

Table 10.10 Conditions Handled as Unsupported Requests (UR) by the PCI-to-PCI Bridge Function (Part 2 of 2)

1. NOTE: Vendor Defined Type 1 messages which target the PCI-to-PCI bridge function are silently discarded.

2. NOTE: Vendor Defined Type 1 messages are silently discarded

Unexpected Completions

Table 10.11 lists the conditions for which the PCI-to-PCI Bridge function in the PES32NT8xG2 ports handles received completions as unexpected completions.

Conditions Handled as UC	Description	PCI Express Base Specification Section
Completion Routing Errors	Refer to section Completions (Routed by ID) on page 10-24.	2.3.2.
Completion that targets a PCI-to-PCI bridge function	PCI-to-PCI bridge functions in the switch never generate requests. Therefore, a completion that target the PCI-to-PCI bridge function in a switch port is treated as an unexpected completion error by the PCI-to-PCI bridge function targeted by the completion.	

Table 10.11 Conditions Handled as Unexpected Completions (UC) by the PCI-to-PCI Bridge Function

TLP Malformation Checks

Table 10.12 lists the TLP malformation checks performed by a PES32NT8xG2 port on reception of TLPs. These checks are performed whenever the port receives the packet from the link.

Notes

TLP Type	Error Check
All	TLP must have a valid FMT/TYPE combination Data payload length <= Max_Payload_Size (i.e., MPS field in PCIEDCTL register)
All TLPs with data (i.e., FMT[1]=1)	LENGTH field must match actual payload data
All TLPs with ECRC (i.e., TD=1)	Actual TLP length must match calculated length (HEADER + PAYLOAD + ECRC)
I/O read or write request	LENGTH = 1 (doubleword) TC = 0 ATTR = 0 Last DWord BE[3:0] = 0b0000
Configuration read or write request	LENGTH = 1 (doubleword) TC = 0 ATTR = 0 Last DWord BE[3:0] = 0b0000
Message Requests interrupt message Power management message Error signaling message Unlock message Set power limit message	TC = 0
TLPs with Route to Root Complex routing.	May only be received on downstream switch ports
TLPs with Broadcast from Root Complex routing.	May only be received on upstream ports
TLPs with Gathered and Routed to Root Complex routing	May only be received by the downstream switch ports Must be a PME_TO_ACK message (all other TLP types with this routing are illegal)
Interrupt messages (INTx)	May only be received by the downstream switch ports
All	TLP traffic class (TC) must be mapped to VC0. TC to VC mapping is controlled by the TC/VC Map (TCVCMAP) field in the VC Capability Structure associated with function 0 of the ingress port.

Table 10.12 Ingress TLP Formation Checks associated with the PCI-to-PCI Bridge Function

Table 10.13 lists the TLP formation error checks performed whenever a port transmits a packet. Note that TLP malformation errors are non-function specific. Therefore, an TLP formation error (detected at ingress or egress) is logged in all functions of the port.

Notes

TLP Type	Error Check
All	TLP traffic class (TC) must be mapped to VC0. TC to VC mapping is controlled by the TC/VC Map (TCVCMAP) field in the egress port's VC Resource 0 Control (VCR0CTL) register of the PCI-to-PCI bridge function.

Table 10.13 Egress Malformed TLP Error Checks

TLP Header Logging

TLP header logging is subject to the rules outlined in section 6.2 of the PCI Express Base Specification.

Note: The PES32NT8xG2 does not support recording of multiple TLP headers, nor does it support recording of headers for uncorrectable internal errors. When an uncorrectable internal error is reported by AER, a header of all ones is recorded.

The following non function-specific errors require that the offending TLP's header be logged in the AER capability structure of all functions in the port:

- Reception of a TLP with ECRC error on the port's link.
- Reception of a request that is unsupported on the port's link, when no function in the port claims the TLP.
- Reception of an unexpected completion on the port's link, when no function in the port claims the TLP.
- Reception of a malformed TLP on the port's link.

The following function-specific errors require that the offending TLP's header be logged in the PCI-to-PCI bridge function's AER capability structure:

- Reception of a request that is unsupported and is claimed by the PCI-to-PCI bridge function.
- Reception of an unexpected completion that is claimed by the PCI-to-PCI bridge function.
- Reception of a TLP that causes an ACS violation (see section ACS Error Handling on page 10-18).
- Reception of a poisoned TLP on the upstream port's link that is claimed by the PCI-to-PCI bridge function. When the TLP is not received on the link, header logging is not performed.
- Reception of a TLP that causes a multicast-blocking error (see Chapter 17, Multicast).

ACS Error Handling

As described in section Access Control Services on page 10-6, ACS checks are performed ports that operate in downstream switch port mode or upstream switch port with NT endpoint mode. All ACS checks are function-specific (i.e., are logged and handled by the function that detected the error). ACS checks may be divided into two groups: ACS checks that re-direct the routing of TLPs and ACS checks that block the routing of TLPs.

- ACS re-direction of a TLP is not considered an error case and is therefore not logged in the function's AER capability structure.
- ACS blocking of a TLP is considered an error case and is logged in the function's AER capability structure. Such an error case is referred to as "ACS violation".

Table 10.14 lists ACS violation checks performed by the PCI-to-PCI bridge function of a port that operates in Downstream Switch Port mode. Note that PES32NT8xG2's downstream switch ports do not support ACS Source Validation on message requests received by a port with 'Local - Terminate at Receiver' and 'Gathered and Routed to Root Complex' routing type (e.g., INTx, PME_TO_Ack, Vendor Defined messages).

Notes

ACS Check	PCI Express Base Specification[1] Section	Role Based (Advisory) Error Reporting Condition	Action Taken
ACS Source Validation	6.12.1.1	Advisory when the corresponding error is configured as non-fatal in the AERUESV register and an ACS violation is detected on a non-posted request	If TLP is a non-posted request, a completion with 'completer abort' status is generated. Note that this is not considered a completer abort error in AER. The Signaled Target Abort (STAS) bit is set in the SECSTS register. Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. TLP header logged in AER. The offending TLP is dropped.
ACS Translation Blocking		Advisory when the corresponding error is configured as non-fatal in the AERUESV register and an ACS violation is detected on a non-posted request	If TLP is a non-posted request, a completion with 'completer abort' status is generated. Note that this is not considered a completer abort error in AER. The Signaled Target Abort (STAS) bit is set in the SECSTS register. Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. TLP header logged in AER. The offending TLP is dropped.
ACS Peer-to-Peer Egress Control		Advisory when the corresponding error is configured as non-fatal in the AERUESV register and an ACS violation is detected on a non-posted request	If TLP is a non-posted request, a completion with 'completer abort' status is generated. Note that this is not considered a completer abort error in AER. The Signaled Target Abort (STAS) bit is set in the SECSTS register. Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. TLP header logged in AER. The offending TLP is dropped.
ACS Direct Translated Peer-to-Peer		(Refer to the next column)	Offending TLP is subject to ACS Peer-to-Peer Egress Control and ACS Peer-to-Peer Request Redirect rules.

Table 10.14 ACS Violations for Ports Operating in Downstream Switch Port Mode

1. Refer to PCI Express Base Specification Revision 2.1.

The PCI-to-PCI bridge function of a port that operates Upstream Switch Port with NT Endpoint mode does not perform any ACS violation checks. The PCI-to-PCI bridge function in such a port only supports ACS re-direction checks (refer to section Access Control Services on page 10-6).

Notes

Transaction Layer Error Pollution

Per section 6.2.3.2.3 of PCI Express Base Specification 2.1, transaction layer errors may be prioritized to prevent error pollution in AER. Error pollution rules only apply to errors associated with the reception of a TLP. Errors not associated with the reception of a TLP are logged for each occurrence of the error.

– The PES32NT8xG2 does not apply error pollution rules to internal errors detected by the device, even when such errors are associated with the reception of a TLP. As a result, it is possible that more than one AER error be logged on reception of a TLP which causes an internal error. For example, reception of a poisoned TLP which causes an internal double-bit ECC error in a port's ingress buffer memory would result in the poisoned and internal errors logged in the ingress port's AER capability structure.

In addition, the Detected Parity Error bit (DPE) in the PCISTS and SECSTS registers is not subject to error pollution rules and is therefore set when the PCI-to-PCI bridge receives a poisoned TLP on its primary or secondary side respectively, even if error pollution rules indicate that the poisoned TLP received error is superseded by a higher priority error.

Table 10.15 shows the prioritization of transaction layer errors used by the switch ports. All errors listed in the table are associated with the reception of a TLP. Errors not applicable to the switch (e.g., completion timeout and completer-abort) are not shown. Higher priority errors have precedence over lower priority errors. Errors with the same priority are mutually exclusive (the errors can't occur simultaneously).

Error	Associated with Packet Reception	Priority
Receiver Overflow	Yes	7 (Highest)
ECRC Check failure	Yes	6
Malformed TLP received	Yes	5
ACS Violation	Yes	4
Multicast Blocked TLP	Yes	3
Unsupported Request	Yes	2
Unexpected Completion received	Yes	
Poisoned TLP received	Yes	1 (lowest)

Table 10.15 Prioritization of Transaction Layer Errors

The prioritization of errors shown in Table 10.15 determines the error that is logged and reported when multiple errors are detected simultaneously for the received TLP. Higher priority errors inhibit the logging and reporting of lower priority errors in AER. Still, higher priority errors do not inhibit the checking and TLP handling action of lower priority errors, unless the higher priority error results in the TLP being consumed, dropped, or nullified by the detecting function (refer to Table 10.9 and Table 10.14).

Figure 10.6 shows the decision diagram for error checking and logging on a received TLP taking into account the error pollution rules and priorities.

Notes

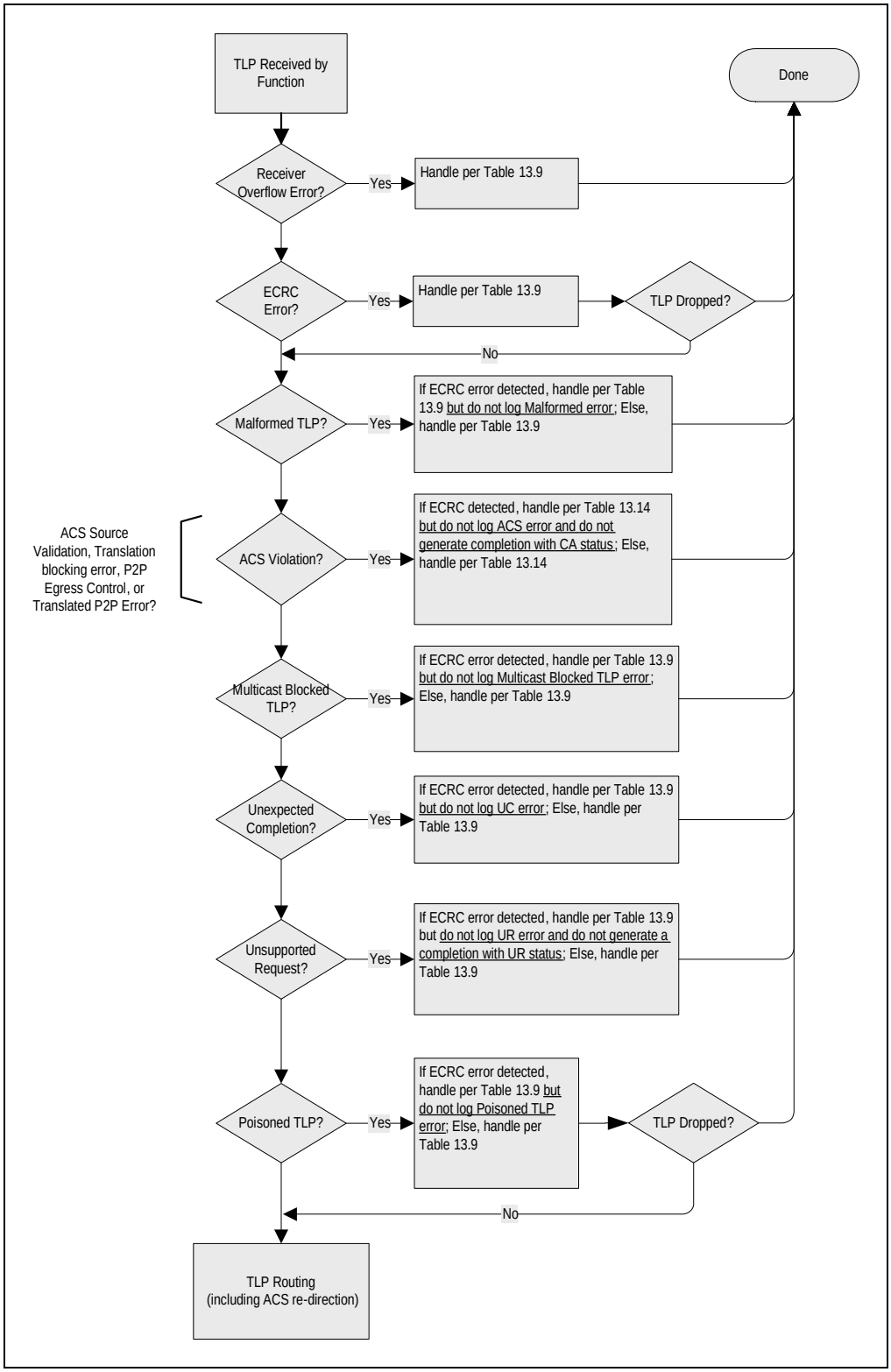

Figure 10.6 Error Checking and Logging on a Received TLP

Notes

Note the following:

- Except for ECRC and Poisoned TLP errors, all other errors detected on the received TLP cause the detecting function to consume, drop, or nullify the TLP.
- Receiver overflow errors are always checked and logged.
- ECRC errors are only checked and logged when the TLP has passed receiver overflow checks.

  Per the error handling rules in Table 10.9, a TLP with ECRC error may result in the TLP being forwarded across the PCI-to-PCI bridge function. Such TLPs are subject to further error checking by the receiving function.
- TLP malformation errors are only logged and reported when the TLP has passed ECRC error checking. Still, a TLP with ECRC error that is not dropped as a result of the ECRC error is subject to TLP malformation error checking since the higher priority ECRC error does not inhibit the checking of the lower-priority malformation error.

  In case the TLP with ECRC error is malformed, the TLP is nullified (per the error handling rules in Table 10.9) but the malformed TLP error is not logged in AER.
- ACS errors violations are only logged and reported when the TLP has passed malformation and ECRC checks. Still, a TLP with ECRC error that is not dropped as a result of the higher priority errors is subject to ACS blocking or re-direction since the higher priority ECRC error does not inhibit the checking of the lower-priority ACS checks.

  In case the TLP with ECRC error is blocked by an ACS check (e.g., ACS Source Validation), the blocking action takes place but the ACS error is not logged and a completion with 'completer-abort' status is not generated.
- Multicast blocking errors are only logged and reported when the TLP has passed ECRC, malformation, and ACS violation checks. Still, a TLP with ECRC error that is not dropped as a result of the higher priority errors is subject to multicast blocking checks since the higher priority ECRC error does not inhibit the checking for multicast blocking errors.

  In case the TLP with ECRC error is blocked by the multicast blocking check, the blocking action takes place but a multicast blocking error is not logged.
- Unsupported request errors are only logged and reported when the request TLP has passed ECRC, malformation, ACS violation, and multicast blocking checks. Still, an unsupported request TLP that is not dropped as a result of the higher priority errors is subject to unsupported request handling.

  In case the TLP with ECRC error is an unsupported request, the TLP is handled per the rules in Table 10.9 but the unsupported is not logged in AER and a completion TLP is not generated.
- Finally, poisoned TLP errors are only logged when the TLP has passed ECRC, malformation, ACS violation, multicast blocking, unsupported request, and unexpected completion checks (i.e., the TLP is a valid request or completion claimed by a port function).

  Per the error handling rules in Table 10.9, poisoned TLP may be forwarded across the PCI-to-PCI bridge function.

For example, when a downstream switch port receives a posted memory request TLP from the link with an ECRC error[1], the port's PCI-to-PCI bridge function will handle the TLP as described in Table 10.9 and Table 10.14. Because ECRC error has higher priority than other errors, only the ECRC error is logged in the port's AER Capability Structure. If the TLP targets the receiving port (i.e., the port is the 'ultimate receiver' of the TLP), the TLP is dropped and no further checking is required. If the port that received the TLP is an intermediate receiver, the TLP is not dropped due to the ECRC error and thus the port performs lower priority error checks and takes the appropriate action. In this example, if the TLP results in an unsupported request error (e.g., the BME bit in the function's PCICMD register is cleared), the port consumes the TLP, does not log the UR error, and does not generate a completion TLP as a result of the UR error.

1. Assuming that the reception of the TLP did not cause a receiver overflow error on the port.

Notes

Routing Errors

This section lists TLP routing errors that are detected by the PCI-to-PCI bridge function in the PES32NT8xG2 ports. Except for completions (section Completions (Routed by ID) on page 10-24), all of these errors are treated as unsupported requests.

Address Routed TLPs

- TLPs received by an upstream port that are not claimed by any function in the upstream port.
- TLPs received by an upstream port that match the upstream port's address range but which do not match a downstream switch port's address range within the partition (i.e., TLPs that do not route through the partition).
- TLPs received by a downstream switch port that do not match the address range of any other downstream switch port within the partition, but match the address range of the partition's upstream port.
- TLPs received by a downstream switch port whose address decoding indicates they are to route back to the port on which they were received, if ACS Upstream Forwarding is disabled on the port. When ACS Upstream Forwarding is enabled, such TLPs are not considered errors and are forwarded upstream.
- TLPs received by the primary side of a port that is not enabled for such transactions.
  - For prefetchable memory and non-prefetchable memory transactions the Memory Access Enable (MAE) bit must be set in the port's PCI Command (PCICMD) register.
  - For I/O transactions the I/O Access Enable (IOAE) bit must be set in the port's PCI Command (PCICMD) register.
- MEM or IO TLPs received on a downstream switch port and the port's Bus Master Enable (BME) bit in the PCICMD register is cleared.
- MEM or IO TLPs from downstream switch ports that target the upstream port and the Bus Master Enable (BME) bit is cleared in the upstream port's PCICMD register.
- A VGA route from a VGA enabled downstream switch port.
- IO TLPs blocked by the ISA Enable (ISAEN) bit in the port's Bridge Control (BCTL) register.
- TLPs with 64-bit address format that target an address below 4 GB (i.e., the upper 32-bits of the address are all zeroes).

Configuration Requests (Routed by ID)

- Type 0 requests that arrive on a downstream switch port.
- Type 1 requests that arrive on a downstream switch port.
- Type 1 requests that do not route through the upstream port's PCI-to-PCI bridge.
- Type 1 requests that are converted to Type 0 requests at the upstream port but which do not target an enabled downstream switch port device number (i.e., target a PCI-to-PCI bridge device number that doesn't exist in the partition).
- Type 1 requests that routed through the PES32NT8xG2, target a downstream switch port's link partner (i.e., are converted to a Type 0 request at the downstream switch port), and which do not target device zero. Note that this check is disabled when the Alternative Routing ID (ARI) function is enabled via the ARIFEN bit in the PCIEDCTL2 register.

Notes

Completions (Routed by ID)

Completions for which there is no valid route across the switch (i.e., the completion can't be forwarded) are treated as unexpected completions. This includes the following cases:

- Completions that attempt to route back onto the link on which they were received, if ACS Upstream Forwarding is disabled. When ACS Upstream Forwarding is enabled, such completion TLPs are not treated as unexpected completions and are forwarded upstream.
- Completions received by a switch port that target a non-existent function in the upstream port.
- Completions received by a switch port that target a non-existent device in the upstream port's bus number.
- Completions received by a switch port that target a non-existent device or function in the switch's virtual PCI bus.
- Completions received by a switch port that fall within the bus aperture of the upstream port but are not claimed by any downstream switch ports.

In addition, all completions that terminate within the PES32NT8xG2 (i.e., ones that target the upstream switch port or any device/function on the virtual PCI bus within the switch) are treated as unexpected completions by the port being targeted.

ID Routed Messages

- Messages that attempt to route back onto the link on which they were received, if ACS Upstream Forwarding is disabled. When ACS Upstream Forwarding is enabled, such TLPs are not considered errors and are forwarded upstream.
- Messages that do not have a valid route through the switch.
- Messages that target a downstream switch port device number that does not exist or is not enabled in the bond option.
- A Vendor Defined Type 0 message which targets an enabled switch port. Vendor Defined Type 1 messages that target a switch port are silently discarded by that port.

Error Emulation Control in the PCI-to-PCI Bridge Function

The PES32NT8xG2 provides the capability to emulate error occurrence in the AER uncorrectable and correctable error status registers. Associated with the PCI-to-PCI bridge function are two error emulation registers. The PCI-to-PCI Bridge Uncorrectable Error Emulation (P2PUEEM) and PCI-to-PCI Bridge Correctable Error Emulation (P2PCEEM) registers allow emulation of errors in the PCI-to-PCI bridge function.

When a bit in these registers is set, it causes the hardware to emulate the detection of the corresponding error. The detection of the error is handled as shown in Figure 6-2 of the PCI Express 2.1 base specification (i.e., the corresponding error is logged in the AER status registers (i.e., AERUES or AERCES), and reported to the root-complex).

- To allow emulation of advisory errors, the P2PUEEM register contains a bit named ADVISORYNF. When this bit is set in conjunction with another bit in the P2PUEEM register, the hardware flags the error as an advisory error and handles it according to Figure 6-2 of the PCI Express 2.1 base specification. Refer to the description of this bit for details.

Since the error emulation does not involve an actual TLP, the AER Header Log registers (AERHL[1:4]DW) in the switch have RWL type, such that they may be modified by software to emulate the capturing of the TLP's header.

Error Emulation Usage and Limitations

The following are some usage guidelines and limitations associated with error emulation.

- To emulate the detection of a correctable error:
  - The desired error bit must be set in the P2PCEEM register.
- To emulate the detection of an uncorrectable fatal error:
  - The desired error bit must be set in the P2PUEEM register.

Notes

- The severity of the error must be set to fatal in the AERUESV register.

– To emulate the detection of an advisory uncorrectable non-fatal error:

– The desired error bit must be set in the P2PUEEM register. The error bit selected must qualify for advisory handling as specified in the PCI Express 2.1 specification. Otherwise, the operation of the emulation logic is undefined.

– The ADVISORYNF bit must be set in the P2PUEEM register.

– The severity of the error must be set to non-fatal in the AERUESV register.

Due to a limitation in the hardware, it is not possible to emulate the detection of a non-advisory uncorrectable non-fatal error.

Notes

Hot-Plug and Hot-Swap

Notes

Overview

As illustrated in Figures 11.1 through 11.3, a PCI Express switch may be used in one of three hot-plug configurations. Figure 11.1 illustrates the use of the PES32NT8xG2 switch in an application in which two downstream switch ports are connected to slots into which add-in cards may be hot-plugged.

Figure 11.2 illustrates the use of the switch in an add-in card application. Here the downstream switch ports are hardwired to devices on the add-in card and the upstream port serves as the add-in card's PCI Express interface. In this application the upstream port may be hot-plugged into a slot on the main system.

Finally, Figure 11.3 illustrates the use of the switch in a carrier card application. In this application, the downstream switch ports are connected to slots which may be hot-plugged and the entire assembly may be hot-plugged into a slot on the main system. Since this application requires nothing more than the functionality illustrated in both Figures 11.1 and 11.2, it will not be discussed further.

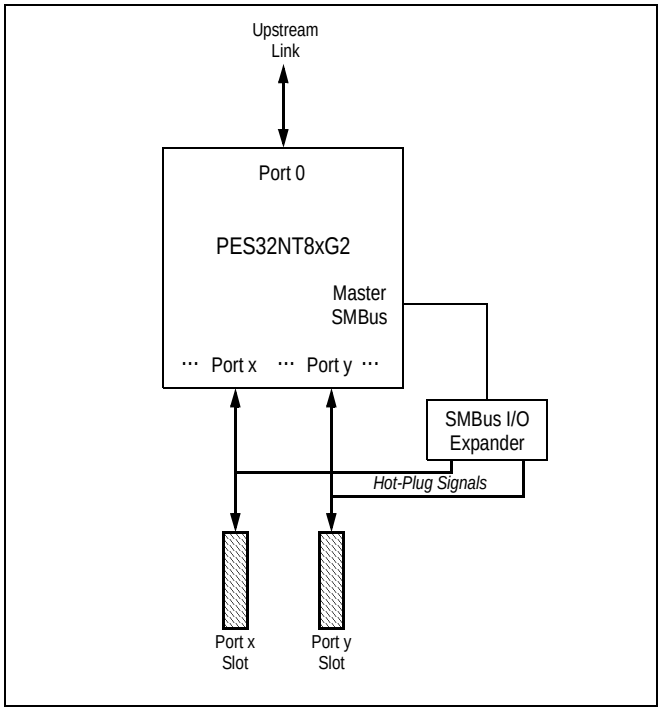

Figure 11.1 Hot-Plug on Switch Downstream Slots Application

Notes

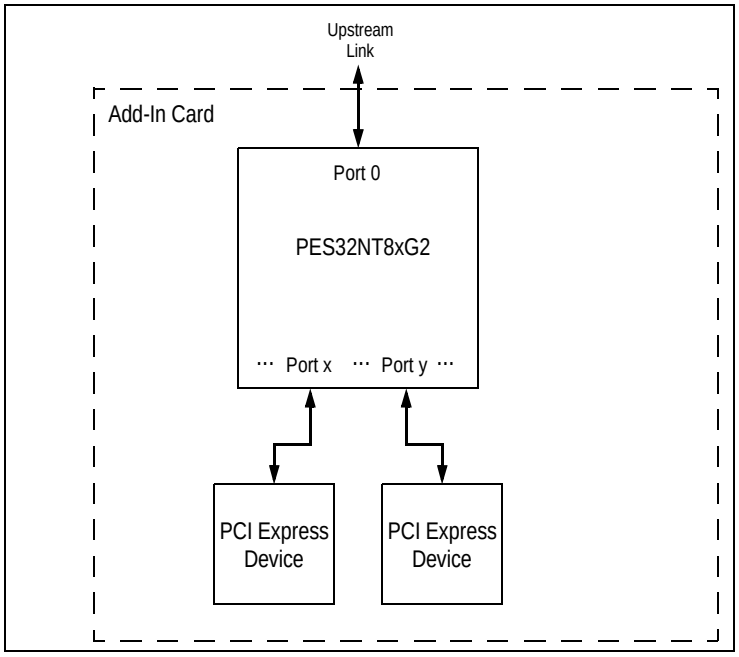

Figure 11.2 Hot-Plug with Switch on Add-In Card Application

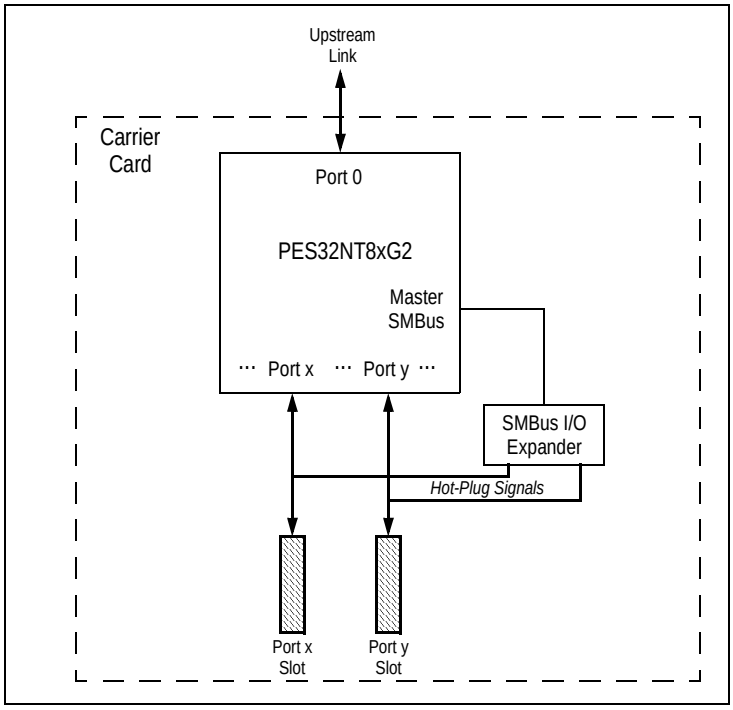

Figure 11.3 Hot-Plug with Carrier Card Application

The PCI Express Base Specification revision 1.0a allowed a hot-plug attention indicator, power indicator and attention button to be located on the board on which the slot is implemented or on the add-in board. When located on the add-in board, state changes are communicated between the hot-plug controller associated with the slot and the add-in card via hot-plug messages. This capability was removed in revision 1.1 of the PCI Express Base Specification and is not supported in the PES32NT8xG2.

Notes

Associated with all PES32NT8xG2 ports is a hot-plug controller. However, hot-plug is only supported when a port is configured to operate in downstream switch port mode. In all other port operating modes, hot-plug is not supported and the hot-plug signals associated with the port are placed in a negated state. Refer to Chapter 5 for details on port operating modes.

Hot-plug is supported within switch partitions. When hot-plug is enabled in a downstream switch port of a switch partition, the behavior is identical do that expected if the switch partition were a stand-alone PCI Express switch. In a port configured to operate in downstream switch port mode, the hot-plug controller may be enabled by setting the HPC bit in the PCI Express Slot Capabilities (PCIESCAP) register associated with that port.

– The HPC bit may be set at any time, but is typically set during a switch fundamental reset via serial EEPROM.

The switch allows hot-plug sensor inputs and indicator outputs to be located next to the slot or on the plug-in module. Regardless of the physical location, the indicators are controlled by the PES32NT8xG2 downstream switch port.

Hot-Plug Signals

All PCI Express defined hot-plug signals are supported on each port using low cost external I/O expanders. The switch requires these I/O expanders for hot-plug operation. Table 11.1 lists the hot-plug inputs and outputs that may be associated with a slot. When enabled during configuration in the PCIESCAP register, these inputs and outputs are made available to external logic using an external I/O expander located on the master SMBus interface.

Signal	Type	Name/Description
PxAIN	O	**Port x[1] Attention Indicator Output.**
PxAPN	I	**Port x Attention Push button Input.**
PxILOCKP	O	**Port x Electromechanical Interlock.**
PxILOCKST	I	**Port x Electromechanical Interlock Status.**
PxMRLN	I	**Port x Manually-operated Retention Latch (MRL) Input.**
PxPDN	I	**Port x Presence Detect Input.**
PxPEP	O	**Port x Power Enable Output.**
PxPFN	I	**Port x Power Fault Input.**
PxPIN	O	**Port x Power Indicator Output.**
PxPWRGDN	I	**Port x Power Good Input (asserted when slot power is good).**
PxRSTN	O	**Port x Reset Output.**

Table 11.1 Port Hot Plug Signals

1. x corresponds to port number (i.e., 0, 2, 4, 6, 8, 12, 16, 20).

The negated value for an unused hot-plug I/O expander is the value shown in Table 11.2. The value is equal to the default value as indicated by the signal name suffix (i.e., N for active low and P for active high) modified, if applicable, as indicated by the corresponding invert polarity bit in the Hot-Plug Configuration Control (HPCFGCTL) register.

Notes

Signal	Negated Output Value with Non-Inverted Polarity (IPXxxx = 0)	Negated Output Value with Inverted Polarity (IPXxxx = 1)
PxAIN	1 (high)	0 (low)
PxILOCKP	0 (low)	1 (high)
PxPEP	0 (low)	1 (high)
PxPIN	1 (high)	0 (low)
PxRSTN[1]	1 (high)	Not Applicable

Table 11.2 Negated Value of Unused Hot-Plug Output Signals

1. PxRSTN signal polarity inversion is not supported.

The switch utilizes external SMBus/I^{2}C-bus I/O expanders connected to the master SMBus interface for hot-plug related signals associated with downstream switch ports. See section I/O Expanders on page 12-11 for details on the operation of the I/O expanders and for the mapping of hot-plug signals to I/O expander inputs and outputs.

SMBus I/O expander transactions are automatically initiated when the state of a hot-plug input signal changes or a new value needs to be driven on a hot-plug output signal. When an I/O Expander is initialized (i.e., the IOEXPADDR field in the IOEXPADDR[4:0] registers is written), the hot-plug controller for the corresponding port initiates an SMBus access to configure the I/O Expander and updates the status bits in the PCI Express Slot Status (PCIESSTS) register. During this initial access, the Presence Detect Changed (PDC) and MRL Sensor Changed (MRLSC) bits in the hot-plug port's PCIESSTS register are not set, since this access is used to determine the initial state of the I/O Expander signals.

The switch supports presence detect signaling via assertion of the Presence Detect Input signal in the external I/O Expander module and through "in-band" presence detect (i.e., the port's PHY detects the presence of a link-partner). The Presence Detect Control (PDETECT) field in the Hot-Plug Configuration Control (HPCFGCTL) register may be used to control the mechanism used for presence detect. The settings in the HPCFGCTL register are globally applied to all hot-plug ports in the switch.

Since the polarity of hot-plug signals has been defined differently in various specifications, each hot plug signal except PxRSTN has a corresponding control bit in the Hot-Plug Configuration Control (HPCFGCTL) that allows the polarity of that signal to be inverted. Inversion affects the corresponding signal in all ports.

When a one is written to the Electromechanical Interlock Control (EIC) bit in the port's PCI Express Slot Control (PCIESCTL) register, then the PxILOCKP signal is pulsed for a length greater than 100 ms and less than 150 ms (i.e., it transitions from negated to asserted, maintains an asserted state for 100 to 150 ms, and then transitions back to negated). When the Toggle Electromechanical Interlock Control (TEMICTL) bit in the HPCFGCTL register is set, writing a one to the EIC bit inverts the state of the PxILOCKP signal.

When the MRL Automatic Power Off (MRLPWROFF) bit is set in the HPCFGCTL register and the Manual Retention Latch Sensor Present (MRLP) bit is set in the PCI Express Slot Capability (PCIESCAP) register, then power to the slot is automatically turned off when the MRL sensor indicates that the MRL is open. This occurs regardless of the state of the Power Controller Control (PCC) bit in the hot-plug port's PCIESCTL register.

The state of a port's Power Fault (PxPFN) input is not latched. For proper operation the system designer should ensure that once the PxPFN signal is asserted, it remains asserted until the power enable (PxPEP) signal is toggled. This is required adapter behavior for the PCI Express ExpressModule form factor.

The default value of hot-plug registers following a switch fundamental reset may be configured via serial EEPROM initialization. Since hot-plug I/O Expander initialization occurs after serial EEPROM initialization, the Command Completed (CC) bit is not set in the hot-plug port's PCIESSTS register as a result of serial EEPROM initialization.

Notes

The default value of fields in the PCIESCTL register following any reset other than a switch fundamental reset (e.g., a partition fundamental reset, partition hot reset, partition upstream secondary bus, or partition downstream secondary bus reset) is determined by the value of the corresponding field in the port's PCI Express Slot Control Initial Value (PCIESCTLIV) register when the corresponding hot-plug capability is enabled.

The SLOT bit in the PCIECAP register and hot-plug capability bits in the PCIESCAP register are SWSticky and, therefore, only get reset to their initial value due to a switch fundamental reset. This means that the hot-plug capability is preserved across all types of partition resets. Following a partition reset, the initial value of control fields in the PCIESCTL register are determined by the PCIESCTLIV register. For example, if the SLOT bit set in the PCIECAP register and the PWRIP bit is set in the PCIESCAP register, then the value of both fields will be preserved across any type of partition reset (i.e., since both of these fields are of RWL type). This means that if the value of the PCC field in the PCIESCTLIV register is zero, then the initial value of the corresponding field in the PCIESCTL register will be zero.

For fields in the PCIESCAP register that control hot-plug output signals, if the value of the field prior to the occurrence of a partition reset is equal to the initial value of the field after a partition reset completes, then the state of the hot-plug output signal is maintained and does not glitch. Continuing the example in the above paragraph, if the PCC field was zero prior to a partition reset and the initial value of the PCC field is zero following a partition fundamental reset, then the PxPEP hot-plug output remains asserted through the partition fundamental reset and does not glitch.

Following a partition hot reset, a partition upstream secondary bus reset, or a downstream secondary bus reset, each downstream switch port's PHY will transition the link to the hot-reset state and subsequently re-train the link starting from the Detect state. When this occurs, the Hot-Plug controller for the port does not set the Presence Detect Changed (PDC) bit in the PCIESSTS register.

Port Reset Outputs

Individual port reset outputs are provided via the PxRSTN hot-plug I/O expander output. Port reset outputs may be configured to operate in one of two modes. These modes are Power Enable Controlled Reset Output and Power Good Controlled Reset Output. The port reset output mode for all downstream switch ports is determined by the Reset Mode (RSTMODE) field in the Hot-Plug Configuration Control (HPCFGCTL) register.

In addition to a port reset output being asserted as determined by the Reset Mode (RSTMODE) field, a port reset output is also asserted under the following circumstances.

- When the partition with which the port is associated experiences a partition fundamental reset. See section Partition Fundamental Reset on page 3-10 for more information on partition fundamental resets.
- When the operating mode of a port is modified and the OMA field is set to reset. See section Reset Mode Change Behavior on page 5-21 for more information on the actions that occur when the OMA field is set to reset.

Hardware ensures that the minimum port reset output assertion pulse width is no less than 200 µS.

Power Enable Controlled Reset Output

In this mode, a downstream switch port reset output state is controlled as a side effect of slot power being turned on or off. The operation of this mode is illustrated in Figure 11.4. A downstream switch port's slot power is controlled by the Power Controller Control (PCC) bit in the PCI Express Slot Control (PCIESCTL) register

Notes

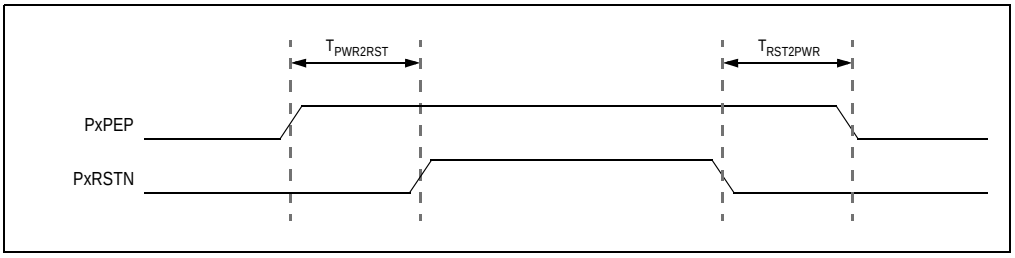

Figure 11.4 Power Enable Controlled Reset Output Mode Operation

While slot power is disabled, the corresponding downstream switch port reset output is asserted. When slot power is enabled by writing a zero to the PCC bit, the Port x Power Enable Output (PxPEP) is asserted and then power to the slot is enabled and the corresponding downstream switch port reset output is negated. The time between the assertion of the PxPEP signal and the negation of the PxRSTN signal is controlled by the value in the Slot Power to Reset Negation (PWR2RST) field in the HPCFGCTL register.

While slot power is enabled, the corresponding downstream switch port reset output is negated. When slot power is disabled by writing a one to the PCC bit, the corresponding downstream switch port reset output is asserted and then slot power is disabled. The time between the assertion of the PxRSTN signal and the negation of the PxPEP signal is controlled by the value in the Reset Negation to Slot Power (RST2PWR) field in the HPCFGCTL register.

Power Good Controlled Reset Output

As in the Power Enable Controlled Reset mode, in this mode a downstream switch port reset output state is controlled as a side effect of slot power being turned on or off. However, the timing in this mode depends on the power good state of the slot's power supply. The operation of this mode is illustrated in Figure 11.5.

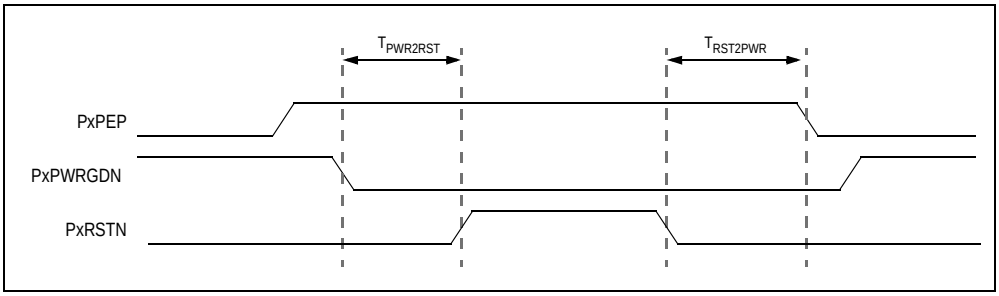

Figure 11.5 Power Good Controlled Reset Output Mode Operation

The operation of this mode is similar to that of the Power Enable Controlled Reset mode except that when power is enabled, the negation of the corresponding port reset output occurs as a result of and after assertion of the slot's Power Good (PxPWRGDN) signal is observed. The time between the assertion of the PxPWRGDN signal and the negation of the PxRSTN signal is controlled by the value in the Slot Power to Reset Negation (PWR2RST) field in the HPCFGCTL register.

When slot power is disabled by writing a one to the PCC bit, the corresponding downstream switch port reset output is asserted and then slot power is disabled. The time between the assertion of the PxRSTN signal and the negation of the PxPEP signal is controlled by the value in the Reset Negation to Slot Power (RST2PWR) field in the HPCFGCTL register.

If at any point while a downstream switch port is not being reset (i.e., PxRSTN is negated) the power good signal (i.e., PxPWRGDN) is negated, then the corresponding port reset output is immediately asserted. Since the PxPWRGDN signal may be configured to be an I/O expander input, it may not be

Notes

possible to meet a profile's power level invalid to reset asserted timing specification (i.e., PxPWRGDN to PxRSTN). Systems that require a shorter time interval may implement this functionality external to the switch.

Hot-Plug Events

The hot-plug controller associated with a downstream switch port slot may generate an interrupt or wakeup event.

Note: Interrupts and wakeup events only affect the partition with which the downstream switch port is associated.

Hot-plug interrupts are enabled when the Hot Plug Interrupt Enable (HPIE) bit is set in the corresponding port's PCI Express Slot Control (PCIESCTL) register.

The following bits, when set in the PCI Express Slot Status (PCIESSTS) register, generate an interrupt if not masked by the corresponding bit in the PCI Express Slot Control (PCIESCTL) register or by the HPIE bit: the Attention Button Pressed (ABP), Power Fault Detected (PFD), MRL Sensor Changed (MRLSC), Presence Detected Changed (PDC), Command Completed (CC), and Data Link Layer Active State Change (DLLLASC).

When an unmasked hot-plug interrupt is generated, the action taken is determined by the MSI Enable (EN) bit in the MSI Capability (MSICAP) register and the Interrupt Disable (INTXD) bit in the downstream switch port's PCI Command (PCICMD) register. Refer to section Interrupts on page 10-4 for details.

When the downstream switch port is in the $D3_{hot}$ state, then the hot-plug controller generates a wakeup event using a PM_PME message instead of an interrupt when the following conditions are satisfied.

- The status bit for an enabled hot-plug event listed below transitions from not set to set.
  - Attention button pressed
  - Power fault detected
  - MRL sensor changed
  - Presence detect changed
  - Command completed event
  - Data link layer state change event
- The PME Enable (PMEE) bit in the downstream switch port's PCI Power Management Control and Status (PMCSR) is set.

It is not required that the Hot Plug Interrupt Enable (HPIE) bit be set in the corresponding port's PCI Express Slot Control (PCIESCTL) register in order to generate a wakeup event using a PM_PME message. Software may clear the HPIE bit to disable interrupt generation while keeping wakeup event generation enabled. If a hot-plug event occurs while a downstream switch port is in $D3_{hot}$ and the corresponding interrupt is enabled, the port will generate an interrupt if the corresponding event's status bit is set in the PCIESCTL is set and the state of the port is transitioned from $D3_{hot}$ to D0 without a reset.

Legacy System Hot-Plug Support

Some systems require support for operating systems that lack PCI Express hot-plug support. The PES32NT8xG2 supports these systems by providing a General Purpose Event (GPEN) output as a GPIO alternate function that can be used instead of the INTx, MSI, and PME mechanisms defined by PCI Express hot-plug.

Note: Because the PES32NT8xG2 only supports a single GPEN output signal, the GPEN signal is associated with partition 0. Partitions other than partition 0 do not support this feature.

Associated with each downstream switch port's hot-plug controller is a bit in the General Purpose Event Control (GPECTL) register. When this bit is set, the corresponding PCI Express Base Specification 2.1 hot-plug event notification mechanisms are disabled for that port and INTx, MSI, and PME events will not be generated by that port due to hot-plug events. Instead, hot-plug events are signaled through assertion of the GPEN signal.

Notes

GPEN is a GPIO alternate function. The GPIO pin will not be asserted when GPEN is asserted unless it is configured to operate as an alternate function. Whenever a port signals a hot-plug event through assertion of the GPEN signal, the corresponding port's status bit is set in the General Purpose Event Status (GPESTS) register. A bit in the GPESTS register can only be set if the corresponding port's hot plug controller is configured to signal hot-plug events using the general purpose event (GPEN) signal assertion mechanism.

The hot-plug event signaling mechanism is the only thing that is affected when a port is configured to use general purpose events instead of the PCI Express defined hot-plug signaling mechanisms (i.e., INTx, MSI, and PME). Thus, the PCI Express defined capability, status, and mask bits defined in the PCI Express slot capabilities, status, and control registers operate as normal and all other hot-plug functionality associated with the port remains unchanged. INTx, MSI, and PME events from other sources are also unaffected.

Hot-Swap

The switch is hot-swap capable and meets the following requirements:

- All of the I/Os are tri-stated on reset (i.e., SerDes, GPIO, SMBuses, etc.)
- All I/O cells function predictably from early power. This means that the device is able to tolerate a non-monotonic ramp-up as well as a rapid ramp-up of the DC power.
- All I/O cells are able to tolerate a precharge voltage.
- Since no clock is present during physical connection, the device will maintain all outputs in a high-impedance state even when no clock is present.
- The I/O cells meet VI requirements for hot-swap.
- The I/O cells respect the required leakage current limits over the entire input voltage range.

In summary, the PES32NT8xG2 meets all of the I/O requirements necessary to build a PICMG compliant hot-swap board or system. The hot-swap I/O buffers of the switch may also be used to construct proprietary hot-swap systems. See the PES32NT8xG2 Data Sheet for a detailed specification of I/O buffer characteristics.

SMBus Interfaces

Notes

Overview

The PES32NT8xG2 has two SMBus interfaces. The slave SMBus interface provides full access to all software-visible registers, allowing every register in the device to be read or written by an external SMBus master. The slave SMBus may also be used to program the serial EEPROM used for initialization. The Master SMBus interface provides connection for an optional external serial EEPROM used for initialization and optional external I/O expanders.

Note: While this document makes reference to a Master SMBus interface, the current implementation of this interface is that of a basic I^2C master.

Six pins make up each of the two SMBus interfaces. These pins consist of an SMBus clock pin, an SMBus data pin, and four SMBus address pins. As shown in Figure 12.1, the master and slave SMBuses may only be used in a split configuration.

The PES32NT8xG2 SMBus master interface does not support SMBus arbitration. As a result, the switch's SMBus master must be the only master in the SMBus lines that connect to the serial EEPROM and I/O expander slaves. In the split configuration, the master and slave SMBuses operate as two independent buses; thus, multi-master arbitration is not required.

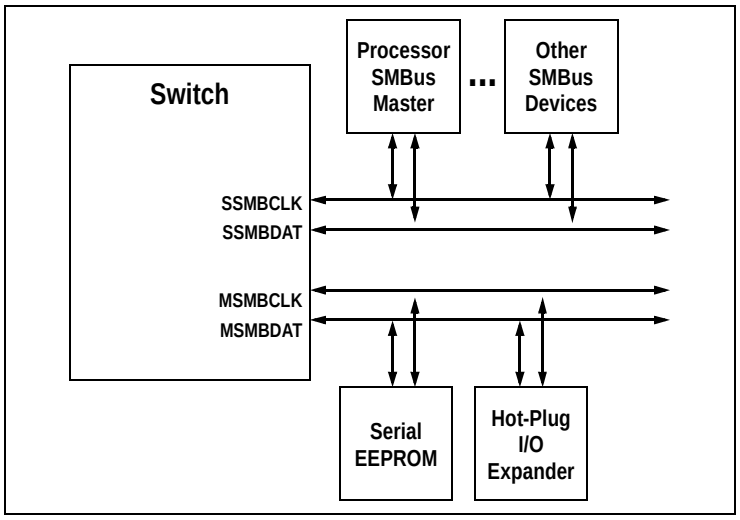

Figure 12.1 Split SMBus Interface Configuration

Master SMBus Interface

The master SMBus interface is used during a switch fundamental reset to load configuration values from an optional serial EEPROM. It is also used to support I/O expanders used for hot-plug and link status signals. This interface operates using a basic I^2C protocol, at a rate of 400 KHz (i.e., I^2C Fast Mode).

Initialization and I^2C Reset

Master SMBus initialization occurs during a switch fundamental reset (see section Switch Fundamental Reset on page 3-2). If the Switch Mode (SWMODE) signal in the boot vector selects one of the following modes, an I^2C reset procedure is performed by the Master SMBus interface during the initialization period.

- Multi-partition with Unattached ports and I^2C Reset
- Multi-partition with Unattached ports and Serial EEPROM initialization with I^2C Reset.

Notes

The goal of the reset procedure is to ensure interoperability with serial EEPROM or IO expander devices that do not have a reset signal input. When the switch is reading from these devices and a fundamental reset is applied to the switch (e.g., via assertion of the PERSTN input signal), the I^2C bus may be left in an unpredictable state that prevents the switch's master SMBus interface from creating a START condition on the bus.

This problem occurs when the I^2C slave device is driving the data signal (MSMBDAT) to a logic 0 at the time the reset occurs. In this case, after the reset is deasserted and the switch starts booting, the switch's master SMBus interface will not be able to drive the data signal to a logic 1 value (since the slave is pulling the signal in the opposite direction), which is a pre-requisite to create a START condition.

The reset procedure described below ensures that the bus is gracefully returned to a state that allows the switch's master SMBus interface to generate a START condition on the bus and re-establish communication with the slaves.

The I^2C reset procedure consists of the switch's master SMBus interface performing the steps shown below.

1. The master SMBus interface drives the MSMBCLK signal with a maximum of nine clock pulses.
2. At each clock pulse, when the MSMBCLK signal is high, the master SMBus interface samples the MSMBDAT signal. If the sampled value is a logic 1, then the next step is performed. Otherwise, step 3 is performed during the ninth clock pulse.

   This step ensures that a slave who was driving the MSMBDAT signal prior to the reset of the switch, releases the MSMBDAT signal during the I^2C acknowledge cycle following the reading of a byte.
3. The master SMBus interface holds the MSMBCLK signal high. This signal is held high until the master SMBus interface decides to create a START condition on the I^2C bus in preparation to access the Serial EEPROM slave or the IO expanders.
   - The START condition is created by driving the MSMBDAT signal low while the MSMBCLK signal remains high.
   - Note that the reception of a START (or repeated START) condition on the I^2C bus causes a slave to reset its bus logic and anticipate the reception of an address, regardless of the positioning of the START condition with respect to prior commands.

Serial EEPROM

During a switch fundamental reset, an optional serial EEPROM may be used to initialize any software-visible register in the device. Serial EEPROM loading occurs if the Switch Mode (SWMODE) signal in the boot vector selects a mode that performs serial EEPROM initialization. The address used by the SMBus master interface to access the serial EEPROM is shown in Table 12.1.

Address Bit	Address Bit Value
1	0
2	0
3	0
4	0
5	1
6	0
7	1

Table 12.1 Serial EEPROM SMBus Address

Notes

Initialization from Serial EEPROM

During initialization from the optional serial EEPROM, the master SMBus interface reads configuration blocks from the serial EEPROM and updates corresponding registers in the switch. Any software-visible register in the device may be initialized with values stored in the serial EEPROM. All software-visible registers have a system address in the PES32NT8xG2's global address space. Configuration blocks stored in the serial EEPROM use this system address shifted right two bits (i.e., configuration blocks in the serial EEPROM use DWord addresses and not byte addresses).

See Chapter 19 for the details on the system address and the global address space address map.

DWord addresses stored in the serial EEPROM are 16-bits wide (i.e., system address bits [17:2]). Therefore, the serial EEPROM may be used to initialize the first 64 K DWords (256 KB) of the global address space. All software-visible registers are located within this region and are therefore accessible via the Serial EEPROM. Since configuration blocks are used to store only the value of those registers that are initialized, a serial EEPROM much smaller than the total size of all of the configuration spaces may be used to initialize the device. Any serial EEPROM compatible with those listed in Table 12.2 may be used to store initialization values.

Serial EEPROM	Size
24C32	4 KB
24C64	8 KB
24C128	16 KB
24C256	32 KB
24C512	64 KB

Table 12.2 PES32NT8xG2 Compatible Serial EEPROMs

During serial EEPROM initialization, the master SMBus interface begins reading bytes starting at serial EEPROM address zero. These bytes are interpreted as configuration blocks and sequential reading of the serial EEPROM continues until the end of a configuration done block is reached or the serial EEPROM address rolls over from 0xFFFF to 0x0. When a serial EEPROM address roll over is detected, loading of the serial EEPROM is aborted, the Serial EEPROM Rollover (ROLLOVER) bit is set in the SMBus Status (SMBUSSTS) register, and the RSTHALT bit is set in the SWCTL register.

A blank serial EEPROM contains 0xFF in all data bytes. When the switch is configured to initialize from serial EEPROM and the first 256 bytes read from the EEPROM all contain the value 0xFF, then loading of the serial EEPROM is aborted, the computed checksum is ignored, the Blank Serial EEPROM (BLANK) bit is set in the SMBus Status (SMBUSSTS) register, and normal device operation begins (i.e., the device operates in the same manner as though it were not configured to initialize from the serial EEPROM).

This is not considered an error. This behavior allows a board manufacturing flow that utilizes uninitialized serial EEPROMs. See section Programming the Serial EEPROM on page 12-10 for information on in-system initialization of the serial EEPROM.

All register initialization performed by the serial EEPROM is performed in DWord quantities. Byte values may be modified by writing the entire DWord.

If during serial EEPROM initialization, an attempt is made to initialize a register that is not defined in a configuration space (i.e., not defined in Chapter 19), then the Unmapped Register Access (URA) bit is set in the SMBUSSTS register and the write is ignored. This is not considered an error.

Notes

There are five configuration block types that may be stored in the serial EEPROM.

- Single double-word initialization sequence
- Sequential double-word initialization sequence
- Jump block
- Wait block
- Configuration done sequence

The first type is a single double-word initialization sequence. A single double-word initialization sequence occupies seven bytes in the serial EEPROM and is used to initialize a single double-word register quantity. A single double-word initialization sequence consists of three fields and its format is shown in Figure 12.2. The CFG TYPE field indicates the type of the configuration block. For single double-word initialization sequence, this value is always 0x0. The SYSADDR field contains the upper 16-bits of the global system address of the double-word to be initialized. The actual global system address, which is a byte address, equals this value with two lower zero bits appended. The final DATA field contains the double-word initialization value.

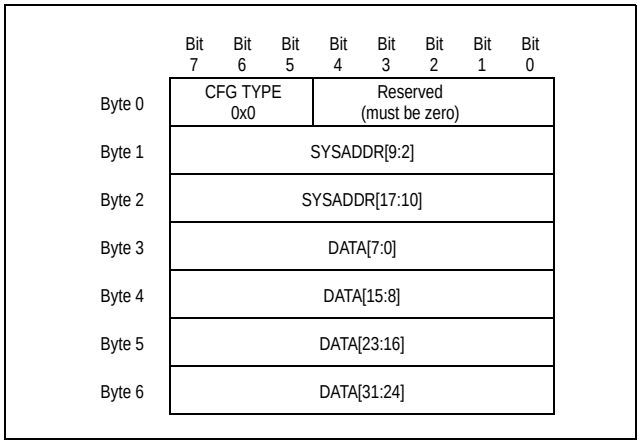

Figure 12.2 Single Double-word Initialization Sequence Format

The second type of configuration block is the sequential double-word initialization sequence. It is similar to a single double-word initialization sequence except that it contains a double-word count that allows multiple sequential double-words to be initialized in one configuration block.

A sequential double-word initialization sequence consists of four required fields and one to 65535 double-word initialization data fields. The format of a sequential double-word initialization sequence is shown in Figure 12.3. The CFG TYPE field indicates the type of the configuration block. For sequential double-word initialization sequences, this value is always 0x1. The SYSADDR field contains the starting double-word system address to be initialized. The NUMDW field specifies the number of double-words initialized by the configuration block. This is followed by the number of DATA fields specified in the NUMDW field.

Notes

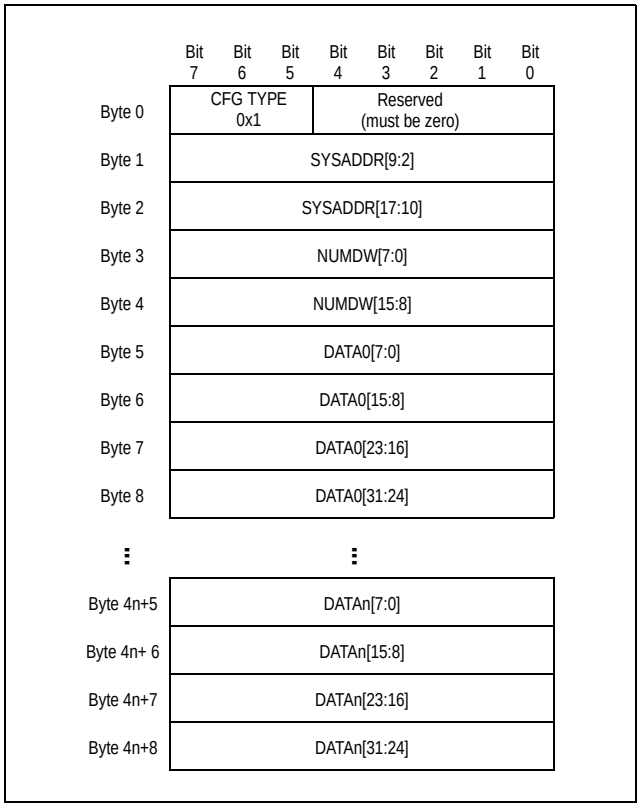

Figure 12.3 Sequential Double-word Initialization Sequence Format

The third type of configuration block is the Jump block. The jump configuration block allows non-sequential execution of the EEPROM initialization sequence. The format of a jump configuration block is shown in Figure 12.4. The CFG TYPE field indicates the type of the configuration block. For jump configuration blocks, this value is always 0x2. The JMP CODE field indicates a jump code that is used in conjunction with the switch mode in order to determine if the jump is taken (see description below). The EEADDR field contains the serial EEPROM's byte address at which execution will continue if the jump is taken (i.e., the jump target address within the serial EEPROM).

A Jump configuration block must indicate a forward jump. That is, the EEADDR field in the Jump configuration block must specify a byte address that is greater than the address of the Jump configuration block itself. For example, a Jump configuration block at byte address 0x10 must specify a jump address no less than 0x12 (since the Jump configuration block comprehends byte addresses 0x10 to 0x12).

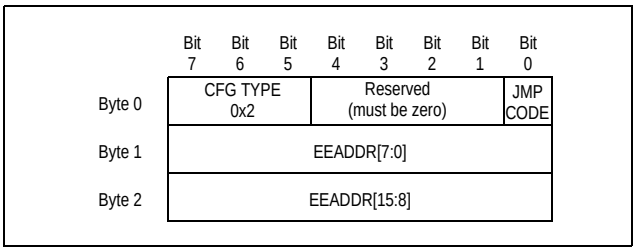

Figure 12.4 Jump Configuration Block

The jump configuration block works in conjunction with the switch mode selected during switch fundamental reset (refer to section Partition Resets on page 3-10). Specifically, the chosen switch mode determines if the jump configuration block is executed or ignored.

Notes

During serial EEPROM initialization, when the SMBus master interface reads a jump configuration block, it evaluates the switch mode to decide if the jump should be taken as shown in pseudo code in Figure 12.5. When the jump is not taken, sequential execution of the Serial EEPROM initialization continues at the address immediately following the jump configuration block. When the jump is taken, execution of the Serial EEPROM continues at the target address specified in the jump configuration block.

```
if ((JMP_CODE = 0x0) & (SWMODE = Single_Partition_With_Serial_EEPROM_Jump0)) {
        Read_Addr = EEADDR; // jump to target EEPROM byte address
}
else if ((JMP_CODE = 0x1) & (SWMODE = Single_Partition_With_Serial_EEPROM_Jump1) {
        Read_Addr = EEADDR; // jump to target EEPROM byte address
}
else {
        Read_Addr = Read_Addr + 1; // sequential execution
}
```

Figure 12.5 Execution of a Jump Configuration Block

The jump configuration block allows programming of the serial EEPROM with up to 3 switch configuration 'images', as shown in Figure 12.6. One configuration image (i.e., configuration A) is executed when the selected switch mode causes all jump configuration blocks to be ignored. Another configuration image (i.e., configuration C) is executed when the switch mode causes jump configuration blocks with jump code 0 to execute. And the third configuration image (i.e., configuration B) is executed when the switch causes configuration blocks with jump code 1 to execute.

Each configuration image initializes the PES32NT8xG2 switch in a user-specified manner. For example, one configuration image could be used to initialize the partition and port control registers to create a switch configuration with four switch partitions and no NTB, while another could create a switch configuration with two partitions connected with an NTB. This allows a single serial EEPROM to be used to create multiple switch configurations, and using the switch mode to choose among configurations.

Notes

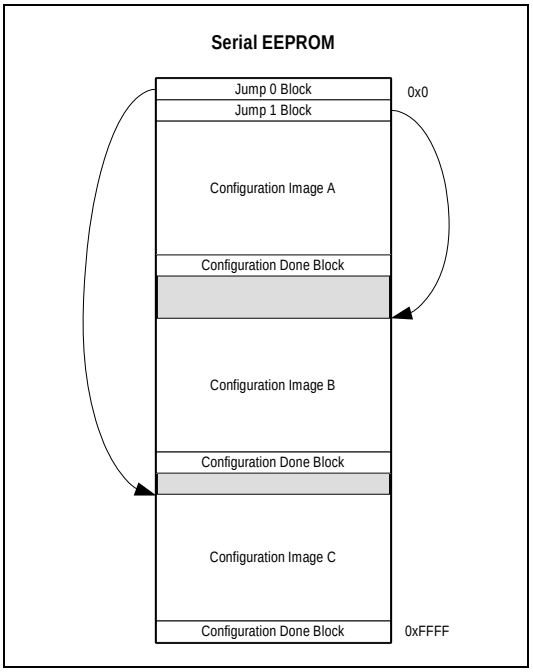

Figure 12.6 Example of Multiple Configuration Images in Serial EEPROM

The PES32NT8xG2 imposes no limitations on the number of jump configuration blocks that may be executed while reading the serial EEPROM. Jump configuration blocks may be located at any byte address within the serial EEPROM.

The fourth type of configuration block is the Wait block. The format of a Wait configuration block is shown in Figure 12.7. The CFG TYPE field indicates the type of the configuration block. For Wait configuration blocks, this value is always 0x3.

As shown in the figure, this configuration block has fields that contain a double-word system address, a 32-bit data value, and a 32-bit mask. The wait block causes the loading of subsequent EEPROM blocks to be suspended until the internal device register at the specified system address contains a value that matches the 32-bit data value provided in the Wait block. The 32-bit mask register is a bit-array that dictates which bits are compared. Bits in the 32-bit data whose corresponding mask bit is set to 1 are masked from comparison (i.e., not compared). Unmasked bits participate in comparison.

Upon reading the Wait configuration block, the SMBus master interface stops reading subsequent configuration blocks from the EEPROM and internally “polls” the value of the register specified SYSADDR field in the Wait block. When the value at this internal register matches the 32-bit data value in the Wait block (masked bits do not participate in the comparison), the SMBus master resumes EEPROM loading and reads the next configuration block.

If the SYSADDR field in the Wait configuration block points to a register that is not defined in the global address space (i.e., not defined in Chapter 19), then the Unmapped Register Access (URA) bit is set in the SMBUSSTS register and EEPROM loading resumes at the instruction following the Wait configuration block.

When executing a Wait configuration block, the maximum wait time allowed by the SMBus master can be configured via the SMBUSCTL register. Each time a Wait configuration block is read from the EEPROM, an internal timer is reset and activated. This internal timer is incremented each clock cycle (i.e., 250 MHz) until the Wait block completes execution or a wait timeout occurs. If the timer reaches the value in the Wait Configuration Block Time Out (WCBT) field in the SMBUSCTL register, a wait timeout occurs and the following actions are taken:

- The Wait Configuration Block Timeout (WCBTO) bit is set in the SMBUSSTS register.
- Loading of the serial EEPROM is aborted and the RSTHALT bit is set in the SWCTL register.

Notes

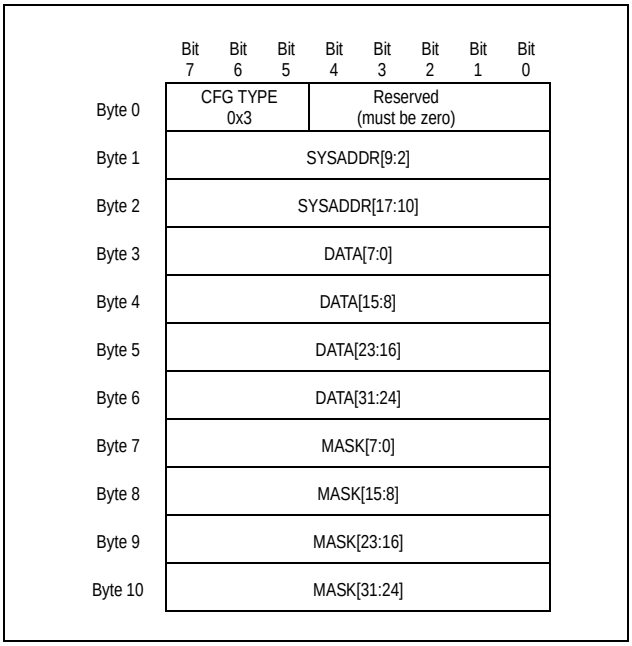

Figure 12.7 Wait Configuration Block

When configuring the PES32NT8xG2 device via EEPROM, the Wait configuration block can be used to ensure that actions performed by previously loaded configuration blocks have completed before proceeding with EEPROM loading. This capability is particularly useful during partition and port configuration (see Chapter 5), as it makes it possible to program the EEPROM instruction sequence to wait for a partition state change or port mode change to complete (i.e., by waiting on the appropriate bits in the SWPARTxSTS and SWPORTxSTS registers respectively) before proceeding with subsequent configurations.

Care must be taken when using the Wait configuration block to check for the completion of partition state changes or port operating mode changes. In particular, note that writing to the STATE field in the SWPARTxCTL register without modifying the value of this field does not constitute a partition state change; therefore, it would be an error to use the Wait configuration block to check that the state change has completed. Similarly, writing to the MODE, SWPART, or DEVNUM fields in the SWPORTxCTL register without modifying their values does not constitute a port operating mode change.

Care must be taken when using the Wait configuration block, as excessive waiting periods will elongate the time it takes to load the EEPROM. This can potentially cause the switch to remain in quasi-reset mode for periods approximating the 1 second limit imposed by the PCI Express Base Specification for devices to successfully complete a configuration request. Refer to section Switch Fundamental Reset on page 3-2 for details.

The final type of configuration block is the configuration done sequence which is used to signify the end of a serial EEPROM initialization sequence.

The configuration done sequence consists of two fields and its format is shown in Figure 12.8. The CFG TYPE field is always 0x7 for configuration done sequences. The CHECKSUM field contains the checksum of all of the bytes in all of the fields read from the serial EEPROM from the first configuration block to the end of this done sequence, including configuration jump blocks.

Notes

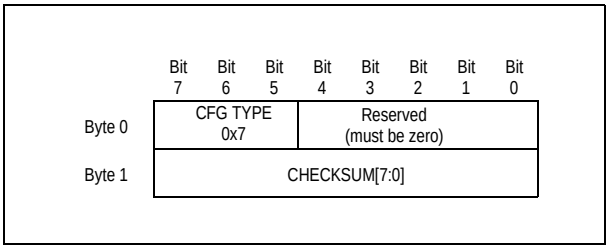

Figure 12.8 Configuration Done Sequence Format

The checksum in the configuration done sequence enables the integrity of the serial EEPROM initialization to be verified. The checksum is computed in the following manner. An 8-bit counter is initialized to zero and the 8-bit sum is computed over the configuration bytes stored in the serial EEPROM, including the entire contents of the configuration done sequence, with the checksum field initialized to zero.[1] The 1's complement of this sum is placed in the checksum field.

Configuration jump blocks processed while reading serial EEPROM are included in the checksum calculation, whether the jump is taken or ignored. For example, in the EEPROM layout shown in Figure 12.6, the checksum in the configuration done block of Configuration A would include the two jump blocks at the beginning of the EEPROM (i.e., to reach Configuration A, these jump blocks are processed even though the jumps are not taken). In this same figure, the checksum in the configuration done block of Configuration B would include also include the two jump blocks at the beginning of the EEPROM. Finally the configuration done block of Configuration C would include the jump 0 block but not the jump 1 block, as only the former is processed before the jump occurs.

The checksum is verified in the following manner. An 8-bit counter is cleared and the 8-bit sum is computed over the bytes read from the serial EEPROM, including the entire contents of the configuration done sequence.[2] The correct result should always be 0xFF (i.e., all ones). Checksum checking may be disabled by setting the Ignore Checksum Errors (ICHECKSUM) bit in the SMBus Control (SMBUSCTL) register.

A summary of possible errors during serial EEPROM initialization and specific action taken when detected is summarized in Table 12.3.

- The detection of any error causes the EEPROM Error Detected (EED) bit to be set in the SMBus Status (SMBUSSTS) register, the loading of the serial EEPROM to be aborted and the RSTHALT bit to be set in the SWCTL register. This allows debugging of the error condition via the slave SMBus interface but prevents normal system operation with a potentially incorrectly initialized device. Specific error information is recorded in the SMBUSSTS register. In addition, the first 3 bytes of the last configuration block processed normally by the master SMBus interface is recorded in the SMBus Configuration Block Header Log (SMBUSCBHL) register.
- The contents of the SMBUSCBHL register can be read to determine if an error exists in the last configuration block that was successfully processed by the master SMBus interface. For example, if a Jump configuration block has an invalid jump address (i.e., a target address at which no valid configuration block is found), the SMBUSCBHL register would log the first 3 bytes of the jump configuration block, thereby capturing the incorrect jump address.

Note that when loading of the serial EEPROM is aborted due to an error, EEPROM configuration blocks loaded prior to the block on which the error is detected are executed normally. The configuration block on which the error is detected may be fully executed, partially executed, or not executed depending on the type of error found.

Once serial EEPROM initialization completes or when it is aborted, the EEPROM Done (EEPROMDONE) bit is set in the SMBUSSTS register.

1. This includes the byte containing the TYPE field.
2. This includes the checksum byte as well as the byte that contains the type and reserved field.

Notes

Error	Action Taken
Configuration Done Sequence checksum mismatch with that computed	- Set RSTHALT bit in SWCTL register - ICSERR bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
Invalid configuration block type	- Set RSTHALT bit in SWCTL register - ICB bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
An unexpected NACK is observed during a master SMBus transaction	- Set RSTHALT bit in SWCTL register - NAERR bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
Serial EEPROM address rollover error detected	- Set RSTHALT bit in SWCTL register - ROLLOVER bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
Wait Configuration Block Timeout	- Set RSTHALT bit in SWCTL register - WCBTO bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
Misplaced START or STOP condition during a master SMBus transaction	- Set RSTHALT bit in SWCTL register - MSS bit is set in the SMBUSSTS register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register
Other error (i.e., an unforeseen error condition is detected)	- Set RSTHALT bit in SWCTL register - EED bit is set in the SMBUSSTS register - Abort initialization, set EEPROMDONE bit in the SMBUSSTS register

Table 12.3 Serial EEPROM Initialization Errors

Programming the Serial EEPROM

The serial EEPROM may be programmed prior to board assembly or in-system via the slave SMBus interface or a PCI Express root. Programming the serial EEPROM via the slave SMBus is described in section Serial EEPROM Read or Write Operation on page 12-22.

A PCI Express root may read and write the serial EEPROM by performing configuration read and write transactions to the Serial EEPROM Interface (EEPROMINTF) register.

To read a byte from the serial EEPROM, the root should configure the Address (ADDR) field in the EEPROMINTF register with the byte address of the serial EEPROM location to be read and the Operation (OP) field to “read.” The Busy (BUSY) bit should then be checked. If the EEPROM is not busy, then the read operation may be initiated by performing a write to the Data (DATA) field. When the serial EEPROM read operation completes, the Done (DONE) bit in the EEPROMINTF register is set and the busy bit is cleared. When this occurs, the DATA field contains the byte data of the value read from the serial EEPROM.

Notes

To write a byte to the serial EEPROM, the root should configure the ADDR field with the byte address of the serial EEPROM location to be written and set the OP field to "write." If the serial EEPROM is not busy (i.e., the BUSY bit is cleared), then the write operation may be initiated by writing the value to be written to the DATA field. When the write operation completes, the DONE bit is set and the busy bit is cleared.

Initiating a serial EEPROM read or write operation when the BUSY bit is set produces undefined results.

SMBus errors may occur when accessing the serial EEPROM. If an error occurs, then it is reported in the SMBus Status (SMBUSSTS) register. Software should check for errors before and after each serial EEPROM access.

I/O Expanders

The PES32NT8xG2 utilizes external SMBus/I^2C-bus I/O Expanders connected to the master SMBus interface for hot-plug and port status signals. The switch is designed to work with Phillips PCA9555 compatible I/O Expanders (i.e., PCA9555, PCA9535, and PCA9539). See the Phillips PCA9555 data sheet for details on the operation of this device.

Note: The MAX7311 is recommended since it is Phillips PCA9555-compatible and supports 64 slave addresses. This allows the use of many I/O Expanders on the same SMBus.

An external SMBus I/O Expander provides 16 bit I/O pins that may be configured as inputs or outputs. The PES32NT8xG2 supports up to 10 external I/O expanders. Table 12.4 summarizes the allocation of functions to I/O expanders. I/O expander signals associated with LED control (i.e., link status and activity) are active low (i.e., driven low when an LED should be turned on). I/O expander signals associated with hot-plug signals are not inverted.

SMBus I/O Expander	Section	Functionality
0	Lower	Port 0 hot-plug
	Upper	Port 4 hot-plug
1	Lower	Port 8 hot-plug
	Upper	Port 16 hot-plug
2	Lower	Port 2 hot-plug
	Upper	Port 6 hot-plug
3	Lower	Port 12 hot-plug
	Upper	Port 20 hot-plug
12	Lower / Upper	Hot-plug MRL inputs (Ports 0, 2, 4, 6, 8, 12, 16, 20)
14	Lower / Upper	Hot-plug electromechanical interlock (Ports 0, 2, 4, 6, 8, 12, 16, 20)
17	Lower / Upper	Link status (Ports 0 through 15)
18	Lower / Upper	Link activity (Ports 0 through 15)
19	Lower / Upper	Link status (Ports 16 and 20) / Link activity (Ports 16 and 20)
20	Lower / Upper	Port reset outputs (Ports 0, 2, 4, 6, 8, 12, 16, 20) / Partition fundamental reset inputs (partitions 0 to 7)

Table 12.4 I/O Expander Functionality Allocation

Notes

During the PES32NT8xG2 initialization, the SMBus/I2C-bus address allocated to each I/O expander used in that system configuration should be written to the corresponding I/O Expander Address (IOE[20:0]ADDR) field. The IOExADDR fields are contained in the I/O Expander Address (IOEXPADDR[5:0]) registers.

Hot-plug outputs and I/O expanders may be initialized via serial EEPROM. Since the I/O expanders and serial EEPROM both utilize the master SMBus, no I/O expander transactions are initiated until serial EEPROM initialization completes. Since no I/O expander transactions are initiated until serial EEPROM initialization completes, it is not possible to toggle a hot-plug output through serial EEPROM initialization (i.e., it is not possible to cause a $0 \rightarrow 1 \rightarrow 0$ transition or a $1 \rightarrow 0 \rightarrow 1$ transition).

Whenever the value of an IOEXPADDR field is written, SMBus write transactions are issued to the corresponding I/O expander by the PES32NT8xG2 to configure the device. This configuration initializes the direction of each I/O expander signal and sets outputs to their default value.

The following I/O expander configuration sequence is issued by the switch to I/O expanders 0 through 3, 14, 15, and 16 (i.e., the ones that contain general port hot-plug signals and electromechanical interlock signals). The I/O expander registers in the sequence are located in the I/O expander device.

1. Write the default value of the outputs bits on the lower eight I/O expander pins (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 2.
2. Write the default value of the outputs bits on the upper eight I/O expander pins (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 3.
3. Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
4. Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
5. Write the configuration value to select inputs/outputs in the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 6.
6. Write the configuration value to select inputs/outputs in the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 7.
7. Read value of I/O expander register 0 to obtain the current state of the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7)
8. Read value of I/O expander register 1 to obtain the current state of the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7)

The following I/O expander configuration sequence is issued by the switch to I/O expanders 12 and 13 (i.e., the ones that contain MRL and partition fundamental reset inputs).

1. Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
2. Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
3. Write the configuration value to select all inputs in the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 6.
4. Write the configuration value to select all inputs in the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 7.
5. Read value of I/O expander register 0 to obtain the current state of the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7)
6. read value of I/O expander register 1 to obtain the current state of the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7)

Notes

The following I/O expander configuration sequence is issued by the switch to I/O expanders 17, 18, and 19 (i.e., the one that contains link up and link activity status).

1. Write link up status for all ports to the lower eight I/O expander pins (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 2.
2. Write link activity status for all ports to the upper eight I/O expander pins (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 3.
3. Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
4. Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
5. Write the configuration value to select all outputs in the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 6.
6. Write the configuration value to select all outputs in the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 7.

The following I/O expander configuration sequence is issued by the switch to I/O expander 20 (i.e., the one that contains port reset outputs).

1. Write the reset status for all ports to the lower eight I/O expander pins (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 2.
2. Write the reset status for all ports to the upper eight I/O expander pins (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 3.
3. Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
4. Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
5. Write the configuration value to select all outputs in the lower eight I/O expander bits (i.e., I/O-0.0 through I/O-0.7) to I/O expander register 6.
6. Write the configuration value to select all outputs in the upper eight I/O expander bits (i.e., I/O-1.0 through I/O-1.7) to I/O expander register 7.

While the I/O expander is enabled, the PES32NT8xG2 maintains the I/O bus expander signals and the switch's internal view of the hot-plug signals in a consistent state. This means that whenever that I/O bus expander state and the internal view of the signal state differs, an SMBus transaction is initiated to resolve the state conflict. An example of an event that may lead to a state conflict is a hot reset. When a hot reset occurs, one or more hot-plug register control fields may be re-initialized to its default value. When this occurs, the internal state of the hot-plug signals is in conflict with the state of I/O expander hot-plug output signals. In such a situation, the switch will initiate an SMBus transaction to modify the state of the I/O expander hot-plug outputs.

The switch has one combined I/O expander interrupt input, labeled IOEXPINTN, which is a GPIO alternate function. Associated with each I/O expander is an open drain interrupt output that is asserted when an I/O expander input pin changes state. The open drain I/O expander interrupt output of all I/O expanders should be tied together on the board and connected to the appropriate GPIO. Whenever IOEXPINTN is asserted, the switch reads the state of all I/O expanders.

Since the I/O expander interrupt input is a GPIO alternate function, the corresponding GPIO should be initialized during configuration to operate in alternate function mode. See Chapter 13, General Purpose I/O.

Whenever the switch needs to change the state of an I/O expander signal output, a master SMBus transaction is initiated to update the state of the I/O expander. This write operation causes the corresponding I/O expander to change the state of its output(s). The switch will not update the state of an I/O expander output more frequently than once every 40 milliseconds. This 40 millisecond time interval is referred to as the I/O expander update period.

Whenever an input to the I/O expander changes state from the value previously read, the interrupt output of the I/O expander is asserted. This causes the switch to issue a master SMBus transaction to read the updated state of all I/O expander inputs. Regardless of the state of the interrupt output of an I/O

Notes

expander, the switch will not issue a master SMBus transaction to read the updated state of the I/O expander inputs more frequently than once every 40 milliseconds (i.e., the I/O expander update period). This delay in sampling may be used to eliminate external debounce circuitry.

The I/O expander interrupt request output is negated whenever the input values are read or when the input pin changes state back to the value previously read. The switch ensures that I/O expander transactions are initiated on the master SMBus in a fair manner. This guarantees that all I/O expanders have equal service latencies. Any error detected during I/O expander SMBus read or write transactions is reflected in the status bits of the SMBus Status (SMBUSSTS) register.

System design recommendations:

- I/O expander addresses and default output values may be configured during serial EEPROM initialization. If I/O expander addresses are configured via the serial EEPROM, then the switch will initialize the I/O expanders when normal device operation begins following the completion of the fundamental reset sequence.
- If the I/O expanders are initialized via serial EEPROM, then the data value for output signals during the SMBus initialization sequence will correspond to those at the time the SMBus transactions are initiated. It is not possible to toggle SMBus I/O expander outputs by modifying data values during serial EEPROM initialization.
- During a fundamental reset and before the I/O expander outputs are initialized, all I/O expander output signals default to inputs. Therefore, pull-up or pull-down resistors should be placed on outputs to ensure that they are held in the desired state during this period.
- All hot-plug data value modifications that correspond to hot-plug outputs result in SMBus transactions. This includes modifications due to partition upstream secondary bus resets and partition hot resets.
- I/O expander outputs are not modified when the device transitions from normal operation to a fundamental reset. In systems where I/O expander output values must be reset during a fundamental reset, a PCA9539 I/O expander should be used.

Hot-Plug I/O Expanders 0 through 3

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	I	P*x*APN[2]	Port *x* attention push button input
1 (I/O-0.1)	I	P*x*PDN	Port *x* presence detect input
2 (I/O-0.2)	I	P*x*PFN	Port *x* power fault input
3 (I/O-0.3)	I	PxPWRGDN	Port *x* power good input
4 (I/O-0.4)	O	PxAIN	Port *x* attention indicator output
5 (I/O-0.5)	O	P*x*PIN	Port *x* power indicator output
6 (I/O-0.6)	O	P*x*PEP	Port *x* power enable output
7 (I/O-0.7)	O	PxRSTN	Port *x* reset output
8 (I/O-1.0)	I	P*y*APN	Port *y* attention push button input
9 (I/O-1.1)	I	P*y*PDN	Port *y* presence detect input
10 (I/O-1.2)	I	P*y*PFN	Port *y* power fault input
11 (I/O-1.3)	I	P*y*PWRGDN	Port *y* power good input
12 (I/O-1.4)	O	P*y*AIN	Port *y* attention indicator output

Table 12.5 Pin Mapping for I/O Expanders 0 through 3 (Part 1 of 2)

Notes

SMBus I/O Expander Bit	Type	Signal	Description
13 (I/O-1.5)	O	P*y*PIN	Port *y* power indicator output
14 (I/O-1.6)	O	P*y*PEP	Port *y* power enable output
15 (I/O-1.7)	O	P*y*RSTN	Port *y* reset output

Table 12.5 Pin Mapping for I/O Expanders 0 through 3 (Part 2 of 2)

1. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

2. Refer to Table 12.6 for the mapping of ports to IO expanders.

I/O Expander	Associated Ports
0	Ports 0 and 4
1	Ports 8 and 16
2	Ports 2 and 6
3	Ports 12 and 20

Table 12.6 I/O Expander 0 through 3 Port Mapping

I/O Expander 12

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	I	P0MRLN	Port 0 manually operated retention latch (MRL) input
1 (I/O-0.1)	I	Unused	Reserved
2 (I/O-0.2)	I	P2MRLN	Port 2 manually operated retention latch (MRL) input
3 (I/O-0.3)	I	Unused	Reserved
4 (I/O-0.4)	I	P4MRLN	Port 4 manually operated retention latch (MRL) input
5 (I/O-0.5)	I	Unused	Reserved
6 (I/O-0.6)	I	P6MRLN	Port 6 manually operated retention latch (MRL) input
7 (I/O-0.7)	I	Unused	Reserved
8 (I/O-1.0)	I	P8MRLN	Port 8 manually operated retention latch (MRL) input
9 (I/O-1.1)	I	Unused	Reserved
10 (I/O-1.2)	I	P12MRLN	Port 12 manually operated retention latch (MRL) input
11 (I/O-1.3)	I	Unused	Reserved
12 (I/O-1.4)	I	P16MRLN	Port 16 manually operated retention latch (MRL) input
13 (I/O-1.5)	I	Unused	Reserved
14 (I/O-1.6)	I	P20MRLN	Port 20 manually operated retention latch (MRL) input
15 (I/O-1.7)	I	Unused	Reserved

Table 12.7 Pin Mapping I/O Expander 12

Notes

[1] I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

I/O Expander 14

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	I	P0ILOCKST	Port 0 electromechanical interlock state input
1 (I/O-0.1)	I	P2ILOCKST	Port 2 electromechanical interlock state input
2 (I/O-0.2)	I	P4ILOCKST	Port 4 electromechanical interlock state input
3 (I/O-0.3)	I	P6ILOCKST	Port 6 electromechanical interlock state input
4 (I/O-0.4)	I	P8ILOCKST	Port 8 electromechanical interlock state input
5 (I/O-0.5)	I	P12ILOCKST	Port 12 electromechanical interlock state input
6 (I/O-0.6)	I	P16ILOCKST	Port 16 electromechanical interlock state input
7 (I/O-0.7)	I	P20ILOCKST	Port 20 electromechanical interlock state input
8 (I/O-1.0)	O	P0ILOCKP	Port 0 electromechanical interlock output
9 (I/O-1.1)	O	P2ILOCKP	Port 2 electromechanical interlock output
10 (I/O-1.2)	O	P4ILOCKP	Port 4 electromechanical interlock output
11 (I/O-1.3)	O	P6ILOCKP	Port 6 electromechanical interlock output
12 (I/O-1.4)	O	P8ILOCKP	Port 8 electromechanical interlock output
13 (I/O-1.5)	O	P12ILOCKP	Port 12 electromechanical interlock output
14 (I/O-1.6)	O	P16ILOCKP	Port 16 electromechanical interlock output
15 (I/O-1.7)	O	P20ILOCKP	Port 20 electromechanical interlock output

Table 12.8 Pin Mapping I/O Expander 14

[1] I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

I/O Expander 17

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	O	P0LINKUPN	Port 0 link up status output
1 (I/O-0.1)	O	Unused	Reserved
2 (I/O-0.2)	O	P2LINKUPN	Port 2 link up status output
3 (I/O-0.3)	O	Unused	Reserved
4 (I/O-0.4)	O	P4LINKUPN	Port 4 link up status output
5 (I/O-0.5)	O	Unused	Reserved
6 (I/O-0.6)	O	P6LINKUPN	Port 6 link up status output
7 (I/O-0.7)	O	Unused	Reserved

Table 12.9 Pin Mapping of I/O Expander 17 (Part 1 of 2)

Notes

SMBus I/O Expander Bit	Type	Signal	Description
8 (I/O-1.0)	O	P8LINKUPN	Port 8 link up status output
9 (I/O-1.1)	O	Unused	Reserved
10 (I/O-1.2)	O	Unused	Reserved
11 (I/O-1.3)	O	Unused	Reserved
12 (I/O-1.4)	O	P12LINKUPN	Port 12 link up status output
13 (I/O-1.5)	O	Unused	Reserved
14 (I/O-1.6)	O	Unused	Reserved
15 (I/O-1.7)	O	Unused	Reserved

Table 12.9 Pin Mapping of I/O Expander 17 (Part 2 of 2)

1. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

I/O Expander 18

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	O	P0ACTIVEN	Port 0 Link active status output
1 (I/O-0.1)	O	Unused	Reserved
2 (I/O-0.2)	O	P2ACTIVEN	Port 2 Link active status output
3 (I/O-0.3)	O	Unused	Reserved
4 (I/O-0.4)	O	P4ACTIVEN	Port 4 Link active status output
5 (I/O-0.5)	O	Unused	Reserved
6 (I/O-0.6)	O	P6ACTIVEN	Port 6 Link active status output
7 (I/O-0.7)	O	Unused	Reserved
8 (I/O-1.0)	O	P8ACTIVEN	Port 8 Link active status output
9 (I/O-1.1)	O	Unused	Reserved
10 (I/O-1.2)	O	Unused	Reserved
11 (I/O-1.3)	O	Unused	Reserved
12 (I/O-1.4)	O	P12ACTIVEN	Port 12 Link active status output
13 (I/O-1.5)	O	Unused	Reserved
14 (I/O-1.6)	O	Unused	Reserved
15 (I/O-1.7)	O	Unused	Reserved

Table 12.10 Pin Mapping of I/O Expander 18

1. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

Notes

I/O Expander 19

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	O	P16LINKUPN	Port 16 link up status output
1 (I/O-0.1)	O	Unused	Reserved
2 (I/O-0.2)	O	Unused	Reserved
3 (I/O-0.3)	O	Unused	Reserved
4 (I/O-0.4)	O	P20LINKUPN	Port 20 link up status output
5 (I/O-0.5)	O	Unused	Reserved
6 (I/O-0.6)	O	Unused	Reserved
7 (I/O-0.7)	O	Unused	Reserved
8 (I/O-1.0)	O	P16ACTIVEN	Port 16 Link active status output
9 (I/O-1.1)	O	Unused	Reserved
10 (I/O-1.2)	O	Unused	Reserved
11 (I/O-1.3)	O	Unused	Reserved
12 (I/O-1.4)	O	P20ACTIVEN	Port 20 Link active status output
13 (I/O-1.5)	O	Unused	Reserved
14 (I/O-1.6)	O	Unused	Reserved
15 (I/O-1.7)	O	Unused	Reserved

Table 12.11 Pin Mapping of I/O Expander 19

1. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

I/O Expander 20

IO Expander 20 provides a copy of the port reset outputs driven via the hot-plug I/O expanders (i.e., I/O expanders 0 through 3). Therefore, this I/O expander allows the use of the port reset outputs without needing to enable hot-plug on the port(s).

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	O	P0RSTN	Port 0 reset output
1 (I/O-0.1)	O	P2RSTN	Port 2 reset output
2 (I/O-0.2)	O	P4RSTN	Port 4 reset output
3 (I/O-0.3)	O	P6RSTN	Port 6 reset output
4 (I/O-0.4)	O	P8RSTN	Port 8 reset output
5 (I/O-0.5)	O	P12RSTN	Port 12 reset output
6 (I/O-0.6)	O	P16RSTN	Port 16 reset output
7 (I/O-0.7)	O	P20RSTN	Port 20 reset output

Table 12.12 Pin Mapping of I/O Expander 20

Notes

SMBus I/O Expander Bit	Type	Signal	Description
8 (I/O-1.0)	I	PART0PERSTN	Partition 0 Fundamental Reset Input
9 (I/O-1.1)	I	PART1PERSTN	Partition 1 Fundamental Reset Input
10 (I/O-1.2)	I	PART2PERSTN	Partition 2 Fundamental Reset Input
11 (I/O-1.3)	I	PART3PERSTN	Partition 3 Fundamental Reset Input
12 (I/O-1.4)	I	PART4PERSTN	Partition 4 Fundamental Reset Input
13 (I/O-1.5)	I	PART5PERSTN	Partition 5 Fundamental Reset Input
14 (I/O-1.6)	I	PART6PERSTN	Partition 6 Fundamental Reset Input
15 (I/O-1.7)	I	PART7PERSTN	Partition 7 Fundamental Reset Input

Table 12.12 Pin Mapping of I/O Expander 20

1. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

Slave SMBus Interface

The slave SMBus interface provides the PES32NT8xG2 with a configuration, management, and debug interface. Using the slave SMBus interface, an external master can read or write any software-visible register in the device.

Initialization

Slave SMBus initialization occurs during a switch fundamental reset. During the switch fundamental reset initialization sequence, the slave SMBus address is initialized. The address is specified by the SSMBADDR[2:1][1] signals as shown in Table 12.13.

Address Bit	Address Bit Value PES32NT8AG2	Address Bit Value PES32NT8BG2
1	SSMBADDR[1]	1
2	SSMBADDR[2]	SSMBADDR[2]
3	1	1
4	0	0
5	1	1
6	1	1
7	1	1

Table 12.13 Slave SMBus Address

1. The PES32NT8BG2 device supports only SSMBADDR2.

Notes

SMBus Transactions

The slave SMBus interface responds to the following SMBus transactions initiated by an SMBus master. See the SMBus Specification Version 2.0, August 3, 2000, SBS Implementers Forum for a detailed description of these transactions.

- Byte and Word Write/Read
- Block Write/Read

Initiation of any SMBus transaction other than those listed above to the slave SMBus interface produces undefined results. Associated with each of the above transactions is a command code. The command code format for operations supported by the slave SMBus interface is shown in Figure 12.9 and described in Table 12.14.

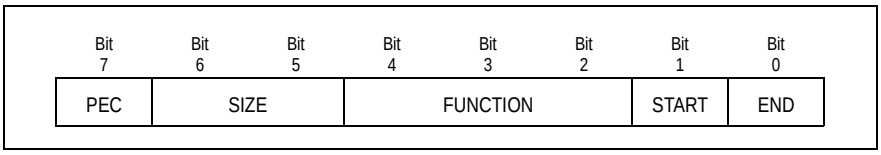

Figure 12.9 Slave SMBus Command Code Format

Bit Field	Name	Description
0	END	End of transaction indicator. Setting both START and END signifies a single transaction sequence 0 - Current transaction is not the last read or write sequence. 1 - Current transaction is the last read or write sequence.
1	START	Start of transaction indicator. Setting both START and END signifies a single transaction sequence 0 - Current transaction is not the first of a read or write sequence. 1 - Current transaction is the first of a read or write sequence.
4:2	FUNCTION	This field encodes the type of SMBus operation. 0 - CSR register read or write operation 1 - Serial EEPROM read or write operation 2 through 7 - Reserved
6:5	SIZE	This field encodes the data size of the SMBus transaction. 0 - Byte 1 - Word 2 - Block 3 - Reserved
7	PEC	This bit controls whether packet error checking is enabled for the current SMBus transaction. 0 - Packet error checking disabled for the current SMBus transaction. 1 - Packet error checking enabled for the current SMBus transaction.

Table 12.14 Slave SMBus Command Code Fields

The FUNCTION field in the command code indicates if the SMBus operation is a system address register read/write or a serial EEPROM read/write operation. Since the format of these transactions is different, they will be described individually in the following sections.

If a command is issued while one is already in progress or if the slave is unable to supply data associated with a command, then the command is NACKed. This indicates to the master that the transaction should be retried.

Notes

CSR Register Read or Write Operation

Table 12.15 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

Byte Position	Field Name	Description
0	CCODE	**Command Code.** Slave Command Code field described in Table 12.14.
1	BYTECNT	**Byte Count.** The byte count field is only transmitted for block type SMBus transactions. SMBus word and byte accesses do not contain this field. The byte count field indicates the number of bytes following the byte count field when performing a write or setting up for a read. The byte count field is also used when returning data to indicate the number of following bytes (including status). In the device, the byte count field must not exceed 4 (i.e., one DWord). Note that the byte count field does not include the PEC byte if PEC is enabled.
2	CMD	**Command.** This field encodes fields related to the CSR register read or write operation.
3	ADDRL	**Address Low.** Lower 8-bits of the double-word system address[1] of register to access.
4	ADDRU	**Address Upper.** Upper 8-bits of the double-word system address of register to access.
5	DATALL	**Data Lower.** Bits [7:0] of data doubleword.
6	DATALM	**Data Lower Middle.** Bits [15:8] of data doubleword.
7	DATAUM	**Data Upper Middle.** Bits [23:16] of data doubleword.
8	DATAUU	**Data Upper.** Bits [31:24] of data doubleword.

Table 12.15 CSR Register Read or Write Operation Byte Sequence

1. Refer to section Overview on page 12-1 for details on PES32NT8xG2's system addresses and the Global Address Space. The SMBus slave interface is capable of addressing only the first 256KB (i.e., 2^{16} DWords) of the Global Address Space.

Table 12.15 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface. Dword addresses and not byte addresses must be used to access all visible software registers. ADDRL and ADDUL represent the lower 8-bit of the doubleword system address and upper 6-bit doubleword system address, respectively. For example, use ADDRU = x00 and ADDRL = 0x00 to access system address 0x00000 (port 0's Vendor/Device ID register). Use ADDRU = x00 and ADDRL = 0x01 to access system address 0x00004 (port 0's Command/Status register).

The format of the CMD field is shown in Figure 12.10 and described in Table 12.16.

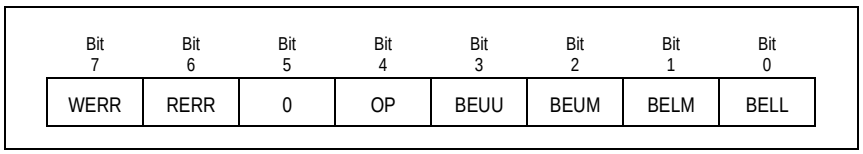

Figure 12.10 CSR Register Read or Write CMD Field Format

Notes

Bit Field	Name	Type	Description
0	BELL	Read/Write	**Byte Enable Lower**. When set, the byte enable for bits [7:0] of the data word is enabled.
1	BELM	Read/Write	**Byte Enable Lower Middle**. When set, the byte enable for bits [15:8] of the data word is enabled.
2	BEUM	Read/Write	**Byte Enable Upper Middle**. When set, the byte enable for bits [23:16] of the data word is enabled.
3	BEUU	Read/Write	**Byte Enable Upper.** When set, the byte enable for bits [31:24] of the data word is enabled.
4	OP	Read/Write	**CSR Operation**. This field encodes the CSR operation to be performed. 0 - CSR write 1 - CSR read
5	0	0	Reserved. Must be zero
6	RERR[1]	Read-Only and Clear	**Read Error**. This bit is set if the last CSR read SMBus transaction was not claimed by the device. Success indicates that the transaction was claimed, not necessarily that the operation completed without error.
7	WERR[1]	Read-Only and Clear	**Write Error**. This bit is set if the last CSR write SMBus transaction was not claimed by the device. Success indicates that the transaction was claimed, not necessarily that the operation completed without error.

Table 12.16 CSR Register Read or Write CMD Field Description

1. The RERR and WERR bits are driven by the switch as status bits that indicate whether or not the switch's SMBus slave interface accepted the register read/write command (the switch accepts the access if it has the correct byte sequence). When a byte sequence refers to a register offset that is not listed or is regarded as a reserve register, the RERR and WERR bits will be set after a read or write operation is performed.

Serial EEPROM Read or Write Operation

Table 12.17 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

Notes

Byte Position	Field Name	Description
0	CCODE	**Command Code.** Slave Command Code field described in Table 12.14.
1	BYTECNT	**Byte Count.** The byte count field is only transmitted for block type SMBus transactions. SMBus word and byte accesses to not contain this field. The byte count field indicates the number of bytes following the byte count field when performing a write or setting up for a read. The byte count field is also used when returning data to indicate the number of following bytes (including status). In the device, the byte count field must not exceed 4 (i.e., one DWord).
2	CMD	**Command.** This field contains information related to the serial EEPROM transaction
3	EEADDR	**Serial EEPROM Address.** This field specifies the address of the Serial EEPROM on the Master SMBus when the USA bit is set in the CMD field. Bit zero must be zero and thus the 7-bit address must be left justified.
4	ADDRL	**Address Low.** Lower 8-bits of the Serial EEPROM byte to access.
5	ADDRU	**Address Upper.** Upper 8-bits of the Serial EEPROM byte to access.
6	DATA	**Data.** Serial EEPROM value read or to be written.

Table 12.17 Serial EEPROM Read or Write Operation Byte Sequence

The format of the CMD field is shown in Figure 12.11 and described in Table 12.18.

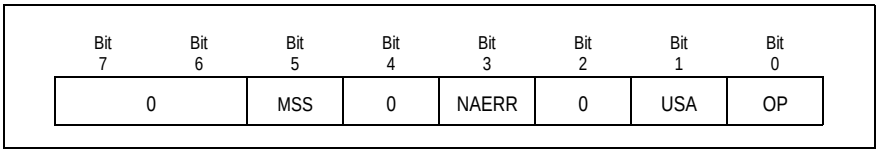

Figure 12.11 Serial EEPROM Read or Write CMD Field Format

Bit Field	Name	Type	Description
0	OP	RW	**Serial EEPROM Operation**. This field encodes the serial EEPROM operation to be performed. 0 - Serial EEPROM write 1 - Serial EEPROM read
1	USA	RW	**Use Specified Address**. When this bit is set the serial EEPROM SMBus address specified in the EEADDR byte is used instead of that specified in the MSMBADDR field in the SMBUSSTS register.
2	Reserved		

Table 12.18 Serial EEPROM Read or Write CMD Field Description (Part 1 of 2)

Notes

Bit Field	Name	Type	Description
3	NAERR	RW1C	**No Acknowledge Error**. This bit is set if an unexpected NACK is observed during a master SMBus transaction when accessing the serial EEPROM. This bit has the same function as the NAERR bit in the SMBUSSTS register. The setting of this bit may indicate the following: that the addressed device does not exist on the SMBus (i.e., addressing error), data is unavailable or the device is busy, an invalid command was detected by the slave, or invalid data was detected by the slave.
4	Reserved		
5	MSS	RW1C	**Misplaced Start & Stop**. This bit is set if a misplaced start & stop error condition is detected by the master SMBus interface when accessing the serial EEPROM. This bit has the same function as the MSS bit in the SMBUSSTS register.
7:6	Reserved	0	Reserved. Must be zero

Table 12.18 Serial EEPROM Read or Write CMD Field Description (Part 2 of 2)

Sample Slave SMBus Operation

This section illustrates sample Slave SMBus operations. Shaded items are driven by the switch's slave SMBus interface and non-shaded items are driven by an SMBus host.

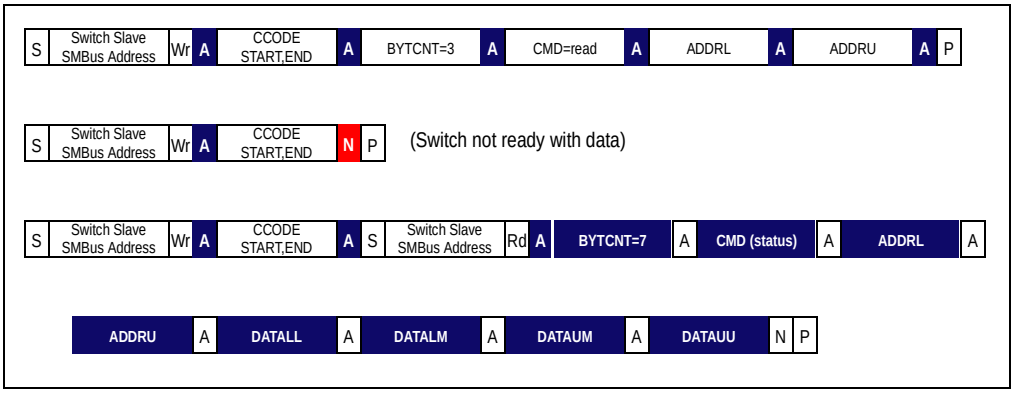

Figure 12.12 CSR Register Read Using SMBus Block Write/Read Transactions with PEC Disabled

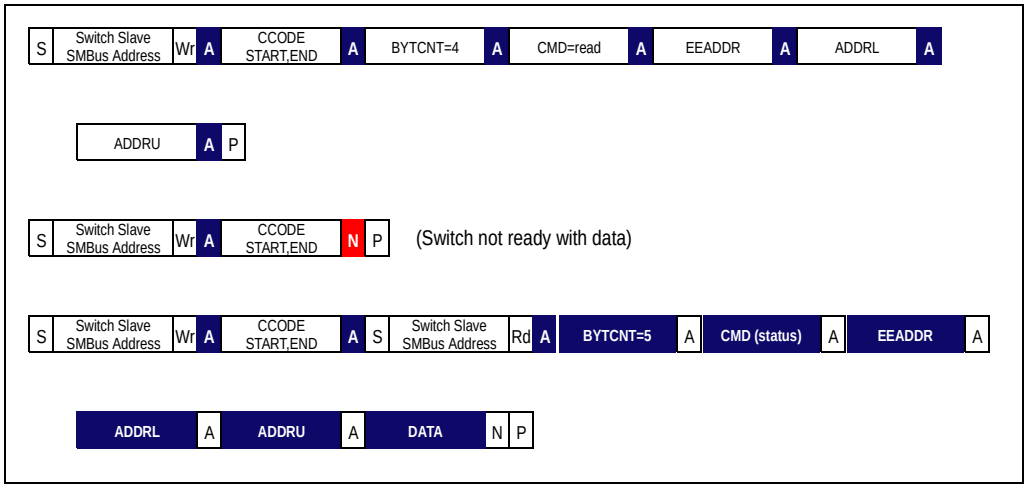

Figure 12.13 Serial EEPROM Read Using SMBus Block Write/Read Transactions with PEC Disabled

Notes

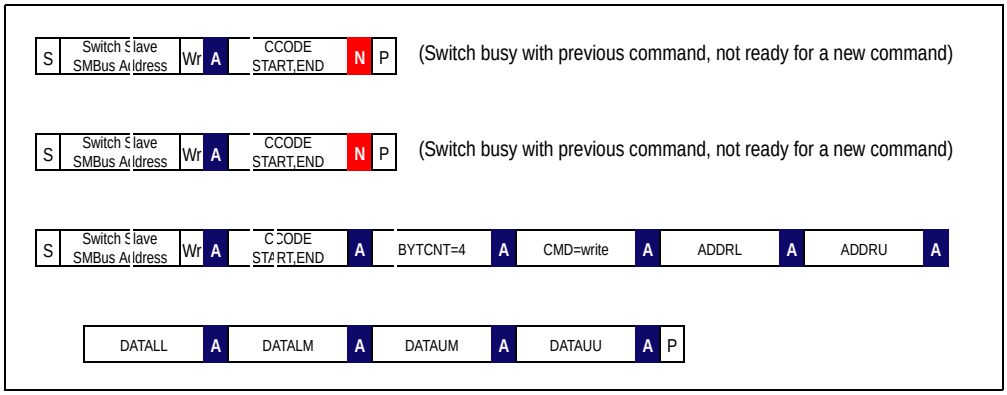

Figure 12.14 CSR Register Write Using SMBus Block Write Transactions with PEC Disabled

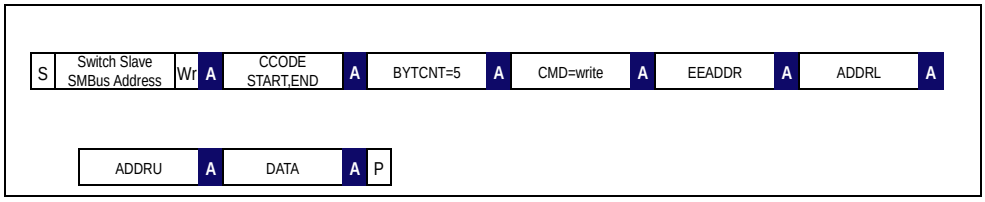

Figure 12.15 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Disabled

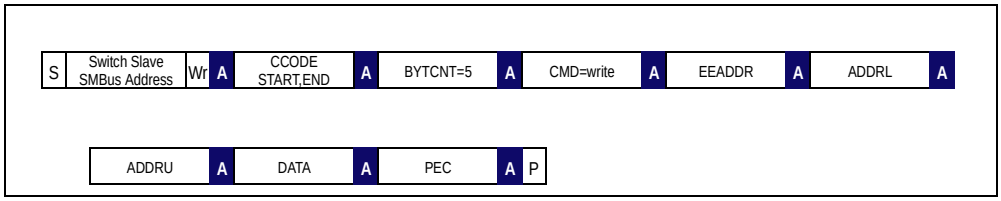

Figure 12.16 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Enabled

Notes

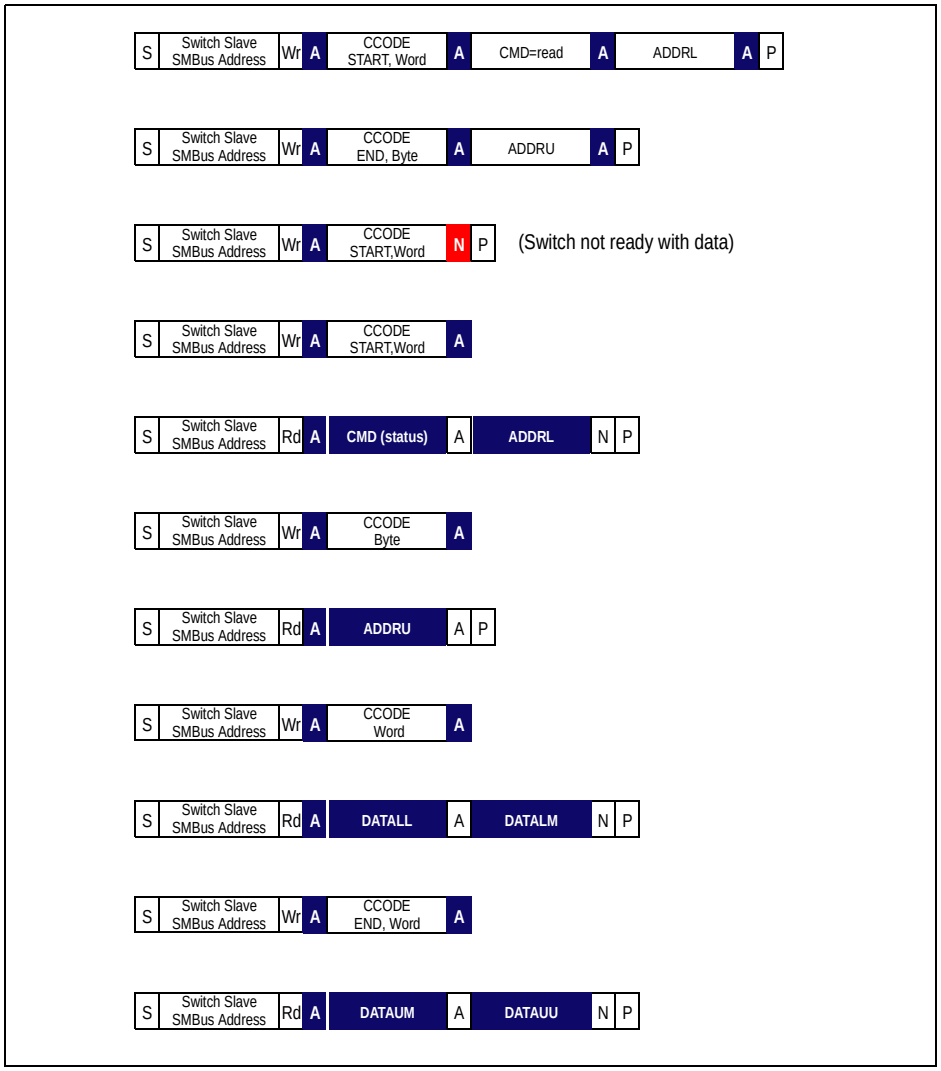

Figure 12.17 CSR Register Read Using SMBus Read and Write Transactions with PEC Disabled

Setting Up I2C Commands for Block Transactions

This section describes how to perform a CSR Register Read and Write Operation. The information contained here does not describe the Serial EEPROM Read or Write Operation, although it is very similar to the CSR Register operations. The main difference is that the serial EEPROM byte sequence contains an extra byte called EEADDR (Serial EEPROM Address) which specifies the address of the Serial EEPROM on the Master SMBUS.

CSR Register Read or Write Operation

In order to setup a CSR Register Read or Write operation byte sequence, the order of the byte sequence must be realized, and the bit fields for each byte sequence must be understood. Tables 12.19, 12.20, and 12.21 provide a description of the CSR byte sequence, command code fields, and CMD field respectively.

Table 12.19 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

Notes

Byte Position	Field Name	Description
0	CCODE	Command Code. Slave Command Code field
1	BYTECNT	Byte Count. The byte count field is only transmitted for block type SMBus transactions. SMBus word and byte accesses do not contain this field. The byte count field indicates the number of bytes following the byte count field when performing a write or setting up for a read. The byte count field is also used when returning data to indicate the number of following bytes (including status). In the device, the byte count field must not exceed 4 (i.e., one DWord).
2	CMD	Command. This field encodes fields related to the CSR register read or write operation.
3	ADDRL	Address Low. Lower 8-bits of the double-word system address1 of register to access.
4	ADDRU	Address Upper. Upper 8-bits of the double-word system address of register to access.
5	DATALL	Data Lower. Bits [7:0] of data doubleword.
6	DATALM	Data Lower Middle. Bits [15:8] of data doubleword.
7	DATAUM	Data Upper Middle. Bits [23:16] of data doubleword.
8	DATAUU	Data Upper. Bits [31:24] of data doubleword.

Table 12.19 CSR Register Read or Write Operation Byte Sequence

SMBus Transactions

The slave SMBus interface responds to the following SMBus transactions initiated by a SMBus master. See the SMBus Specification Version 2.0, August 3, 2000, SBS Implementers Forum for a detailed description of these transactions.

- Byte and Word Write/Read
- Block Write/Read

Initiation of any SMBus transaction other than those listed above to the slave SMBus interface produces undefined results. Associated with each of the above transactions is a command code. The command code format for operations supported by the slave SMBus interface is shown in Table 12.20.

Notes

Bit Fields	Field Name	Description
0	END	End of transaction indicator. Setting both START and END signifies a single transaction sequence 0 - Current transaction is not the last read or write sequence. 1 - Current transaction is the last read or write sequence.
1	START	Start of transaction indicator. Setting both START and END signifies a single transaction sequence 0 - Current transaction is not the first of a read or write sequence. 1 - Current transaction is the first of a read or write sequence.
4 : 2	FUNCTION	This field encodes the type of SMBus operation. 0 - CSR register read or write operation 1 - Serial EEPROM read or write operation 2 through 7 - Reserved
6 : 5	SIZE	This field encodes the data size of the SMBus transaction. 0 - Byte 1 - Word 2 - Block 3 - Reserved
7	PEC	This bit controls whether packet error checking is enabled for the current SMBus transaction. 0 - Packet error checking disabled for the current SMBus transaction. 1 - Packet error checking enabled for the current SMBus transaction.

Table 12.20 Slave SMBus Command Code Fields

The FUNCTION field in the command code indicates if the SMBus operation is a system address register read/write or a serial EEPROM read/write operation. If a command is issued while one is already in progress or if the slave is unable to supply data associated with a command, then the command is NACKed. This indicates to the master that the transaction should be retried.

The format of the CMD field is shown in Table 12.21.

Bit Field	Field Name	Type	Description
0	BELL	Read/Write	Byte Enable Lower. When set, the byte enable for bits [7:0] of the data word is enabled.
1	BELM	Read/Write	Byte Enable Lower Middle. When set, the byte enable for bits [15:8] of the data word is enabled.
2	BEUM	Read/Write	Byte Enable Upper Middle. When set, the byte enable for bits [23:16] of the data word is enabled.
3	BEUU	Read/Write	Byte Enable Upper. When set, the byte enable for bits [31:24] of the data word is enabled.
4	OP	Read/Write	CSR Operation. This field encodes the CSR operation to be performed. 0 - CSR write 1 - CSR read

Table 12.21 CSR Register Read or Write CMD Field Description (Part 1 of 2)

Notes

Bit Field	Field Name	Type	Description
5	0	0	Reserved. Must be zero
6	RERR	Read-Only and Clear	Read Error. This bit is set if the last CSR read SMBus transaction was not claimed by a device. Success indicates that the transaction was claimed and not that the operation completed without error.
7	WERR	Read-Only and Clear	Write Error. This bit is set if the last CSR write SMBus transaction was not claimed by a device. Success indicates that the transaction was claimed and not that the operation completed without error.

Table 12.21 CSR Register Read or Write CMD Field Description (Part 2 of 2)

Examples of Setting Up the I2C CSR Byte Sequence for a CSR Register Read

The pseudo examples below demonstrate the block transaction settings for a CSR register read. As an example, in many of the IDT utilities the CSR byte sequence array is passed to a TotalPhase Aardvark I2C control function.

Table 12.22 lists the constant variables used in this example of setting up the command byte array.

Constant Name	Value	Description
CCode_Block	0x40	[6:5] = 2 - Block option
CCode_Init	0x03	[0:0] = 1 - Current transaction is the last read or write sequence [1:1] = 1 - Current transaction is the first of a read or write sequence.
TranSize_BkWtHeader	3	Transaction block write header of size 3
TranSize_BkRdHeader	4	Transaction block read header of size 4
TranSize_Block	5	Transaction block size of 5
CMD_Init	0x00	[7:0] Initial CMD byte as zero
CMD_BELL	0x01	[0:0] Byte Enable Lower.
CMD_BELM	0x02	[1:1] Byte Enable Lower Middle.
CMD_BEUM	0x04	[2:2] Byte Enable Upper Middle.
CMD_BEUU	0x08	[3:3] Byte Enable Upper.
CMD_DWORD	0x0F	[3:0] Enable BELL, BELM, BEUM, BEUU
CMD_OPRD	0x10	[4:4] Enable CSR Operation default as CSR Read
Len_Byte	1	Length of 1 (1 byte)
Len_Word	2	Length of 2 (2 bytes)
Len_Dword	4	Length of 4 (4 bytes)

Table 12.22 Constants Used in Examples

Step 1. Initialize a CSR register offset variable and a command code init variable

CSR_Offset = address shifted by 2 bits to the right

CCode_i = CCode_Init

Notes

The CSR_Offset is shifted 2 bits to the right so that DWORD aligned register offsets are only accessible; this step may not be needed for some devices.

Read BYTE Setup

Steps 2 and 3 show how each index in the CSR byte sequence array is set for a BYTE read operation. For step 3, the transaction size is a value that is passed to the I2C control function so that it knows how many bytes are being dealt with in the CSR byte sequence.

Step 2. Prepare the I2C byte array

Table 12.23 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader
2	BKCmd_i = CMD_Init | CMD_BELL | CMD_OPRD
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8

Table 12.23 I2C Command Byte Array Indices

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode_i = 0x03 | 0x40 = 0x43

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader

BKCnt_i = 3

The byte count field indicates the number of bytes following the byte count field when setting up for a write or setting up for a read.

Index 2 - Set the byte option (BELL) and set the CSR READ operation (OPRD)

BKCmd_i = CMD_Init | CMD_BELL | CMD_OPRD

BKCmd_i = 0x00 | 0x01 | 0x10 = 0x11

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL_i = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU_i = (0x00 & 0xFF00) >> 8 = 0

Step 3. Calculate the transaction size and read length

TranSize = TranSize_WtB4Rd + 1

ReadLength = length + TranSize_BkRdHeader

Notes

Read WORD Setup

Steps 2 and 3 in this section (see Step 1 above) shows how each index in the CSR byte sequence array is set for a WORD read operation. For step 3, the transaction size is a value that is passed to the I2C control function so that it knows how many bytes are being dealt with in the CSR byte sequence.

Step 2. Prepare the I2C byte array

Table 12.24 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader
2	BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM | CMD_OPRD
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8

Table 12.24 I2C Command Byte Array Indices

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode_i = 0x03 | 0x40 = 0x43

In the Command Code

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader

BKCnt_i = 3

Index 2 - Set the word option (BELL) and (BELM), and set the CSR READ operation (OPRD)

BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM | CMD_OPRD

BKCmd_i = 0x00 | 0x01 | 0x02 | 0x10 = 0x13

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL_i = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU_i = (0x00 & 0xFF00) >> 8 = 0

Step 3. Calculate the transaction size and read length

TranSize = TranSize_WtB4Rd + 1

ReadLength = length + TranSize_BkRdHeader

Read DWORD Setup

Steps 2 and 3 show how each index in the CSR byte sequence array is set for a DWORD read operation. For step 3, the transaction size is a value that is passed to the I2C control function so that it knows how many bytes are being dealt with in the CSR byte sequence.

Notes

Step 2. Prepare the I2C byte array

Table 12.25 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader
2	BKCmd_i = CMD_Init | CMD_DWORD | CMD_OPRD
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8

Table 12.25 I2C Command Byte Array Indices

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode = 0x03 | 0x40 = 0x43

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader

BKCnt = 3

Index 2 - Set the dword option (BELL, BELM, BEUM, BEUU), and CSR READ operation (OPRD)

BKCmd_i = CMD_Init | CMD_DWORD | CMD_OPRD

BKCmd = 0x00 | 0x0F | 0x10 = 0x1F

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU = (0x00 & 0xFF00) >> 8 = 0

Step 3. Calculate the transaction size and read length

TranSize = TranSize_WtB4Rd + 1

ReadLength = length + TranSize_BkRdHeader

Examples of Setting Up the I2C CSR Byte Sequence for a CSR Register Write

The following examples are for the CSR byte sequence array write operations. There are three examples: BYTE write, WORD write, and DWORD write. Refer to Table 12.22 for the constant variables used in the examples. In the CSR write operation, the data bytes in the CSR byte sequence are used. Please refer to Table 12.19 for more information about the byte index locations.

In the following examples, "data" is a variable that is arbitrarily set to 0xBBAA2211.

Step 1. Initialize a CSR register offset variable and a command code init variable

CSR_Offset = address shifted by 2 bits to the right

CCode_i = CCode_Init

Notes

Write BYTE Setup

Steps 2 and 3 show how each index in the CSR byte sequence array is set for a BYTE write operation.

Step 2. Prepare the I2C byte array

Table 12.26 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader + Len_Byte
2	BKCmd_i = CMD_Init | CMD_BELL
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8
5	BKDtL_i = low byte of data

Table 12.26 I2C Command Byte Array Indices

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode_i = 0x03 | 0x40 = 0x43

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader + Len_Byte

BKCnt_i = 3 + 1 = 4

The byte count field indicates the number of bytes following the byte count field when setting up for a write or setting up for a read.

Index 2 - Set the byte option (BELL) and set the CSR WRITE operation (clear OPRD bit)

BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM

BKCmd_i = 0x00 | 0x01 | 0x10 = 0x11

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL_i = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU_i = (0x00 & 0xFF00) >> 8 = 0

Index 5 - Set the lower data byte

BKDtL_i = low byte of data

BKDtL_i = 0x11 (of 0xBBAA2211)

Step 3. Calculate the transaction size

TranSize = TranSize_Block + byte_length

= 5 + 1

= 6

Notes

Write WORD Setup

Steps 2 and 3 shows how each index in the CSR byte sequence array is set for a WORD write operation.

Step 2. Prepare the I2C byte array

Table 12.27 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader + Len_Word
2	BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8
5	BKDtL_i = low byte of word data
6	BKDtL_i+1 = high byte of word data

Table 12.27 I2C Command Byte Array Indices

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode_i = 0x03 | 0x40 = 0x43

In the Command Code

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader + Len_Word

BKCnt_i = 3 + 2

= 5

Index 2 - Set the word option (BELL) and (BELM), and set the CSR WRITE operation (clear OPRD bit)

BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM

BKCmd_i = 0x00 | 0x01 | 0x02

= 0x03

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL_i = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU_i = (0x00 & 0xFF00) >> 8 = 0

Index 5 - Set the lower data byte

BKDtL_i = low byte of word data

BKDtL_i = 0x11 (of 0xBBAA2211)

Notes

Index 6 - Set the upper data byte

BKDtL_i+1 = high byte of word data

BKDtL_i+1 = 0x22 (of 0xBBAA2211)

Step 3. Calculate the transaction size

TranSize = TranSize_Block + word_length

= 5 + 2

= 7

Write DWORD Setup

Steps 2 and 3 show how each index in the CSR byte sequence array is set for a DWORD write operation.

Step 2. Prepare the I2C byte array

Table 12.28 shows the block byte array assignments (in increasing index order starting from index 0). Address offset 0 is used in the examples.

Index #	Assignment Description
0	CCode_i |= CCode_Block
1	BKCnt_i = TranSize_BkWtHeader + Len_Dword
2	BKCmd_i = CMD_Init | CMD_DWORD
3	BKOfL_i = CSR_Offset & 0xFF
4	BKOfU_i = (CSR_Offset & 0xFF00) >> 8
5	BKDtL_i = low byte of low word data
6	BKDtL_i+1 = high byte of low word data
7	BKDtL_i+2 = high byte of high word data
8	BKDtL_i+3 = high byte of high word data

Table 12.28 I2C Command Byte Array Indice

Index 0 - Initialize the command code byte

CCode_i |= CCode_Block

CCode_i = 0x03 | 0x40 = 0x43

In the Command Code

0x43 = (start bit= 1, end bit= 1, function_bits= CSR, size_bits= BLOCK)

Index 1 - Set the byte count

BKCnt_i = TranSize_BkWtHeader + Len_Dword

BKCnt_i = 3 + 4

= 9

Index 2 - Set the word option (BELL) and (BELM), and set the CSR READ operation (OPRD)

BKCmd_i = CMD_Init | CMD_BELL | CMD_BELM

BKCmd_i = 0x00 | 0x01 | 0x02

= 0x03

Notes

Index 3 - Set the lower CSR register offset

BKOfL_i = CSR_Offset & 0xFF

BKOfL_i = 0x00 & 0xFF = 0

Index 4 - Set the upper CSR register offset

BKOfU_i = (CSR_Offset & 0xFF00) >> 8

BKOfU_i = (0x00 & 0xFF00) >> 8 = 0

Index 5 - Set the lower data byte

BKDtL_i = low byte of low word data

BKDtL_i = 0x11 (of 0xBBAA2211)

Index 6 - Set the upper data byte

BKDtL_i+1 = high byte of low word data

BKDtL_i+1 = 0x22 (of 0xBBAA2211)

Index 7 - Set the lower data byte

BKDtL_i = low byte of high word data

BKDtL_i+2 = 0xAA (of 0xBBAA2211)

Index 8 - Set the upper data byte

BKDtL_i+1 = high byte of high word data

BKDtL_i+3 = 0xBB (of 0xBBAA2211)

Step 3. Calculate the transaction size

TranSize = TranSize_Block + Dword_length

= 5 + 4

= 9

General Purpose I/O

Notes

Overview

The switch has 9 General Purpose I/O (GPIO) pins that may be individually configured as general purpose inputs, general purpose outputs, or alternate functions. GPIO pins are controlled by the General Purpose I/O Function (GPIOFUNC), General Purpose I/O Configuration (GPIOCFG), General Purpose I/O Data (GPIOD), and General Purpose I/O Alternate Function Select (GPIOAFSEL) register.

After a switch fundamental reset, all GPIO pins default to a GPIO input function. GPIO pins configured as GPIO inputs are double-clocked and sampled no more frequently than once every 128 ns. Thus, they may be treated as asynchronous inputs. Associated with each GPIO pin are alternate functions.

Note: Care should be exercised when configuring GPIO pins as outputs since an incorrect configuration could cause damage to the switch and external components.

GPIO Configuration

Associated with each GPIO pin is a bit in the GPIOFUNC, GPIOCFG and GPIOD registers. Table 13.1 summarizes the configuration of GPIO pins.

GPIOFUNC	GPIOCFG	Pin Functionality
0	0	GPIO input
0	1	GPIO output
1	don't care	Alternate function

Table 13.1 GPIO Pin Configuration

Input

When configured as an input in the GPIOCFG register and as a GPIO function in the GPIOFUNC register, the GPIO pin is sampled and registered in the GPIOD register. The value of the input pin can be determined at any time by reading the GPIOD register. The value in this register corresponds to the value of the pin irrespective of whether the pin is configured as a GPIO input, GPIO output, or alternate function.

Output

When configured as an output in the GPIOCFG register and as a GPIO function in the GPIOFUNC register, the value in the corresponding bit position of the GPIOD register is driven on the pin.

– System designers should treat the GPIO outputs as asynchronous outputs.

The actual value of the output pin may be determined by reading the GPIOD register.

Alternate Function

Each GPIO pin has two alternate functions. An alternate function associated with a specific GPIO pin is selected by the corresponding field in the GPIOAFSEL register. The alternate functions associated with each GPIO pin are listed in Table 13.2.

Notes

GPIO Pin	Alternate Function 0	Alternate Function 1
0	PART0PERSTN	P16LINKUPN
1	PART1PERSTN	P16ACTIVEN
2	PART2PERSTN	P4LINKUPN
3	PART3PERSTN	P4ACTIVEN
4	FAILOVER0	P0LINKUPN
5	GPEN	P0ACTIVEN
6	FAILOVER1	FAILOVER3
7	FAILOVER2	P8LINKUPN
8	IOEXPINTN	P8ACTIVEN

Table 13.2 GPIO Alternate Function Pin Assignment

Alternate function signals are described in Table 13.3.

Signal	Type	Name/Description
FAILOVERx	I	**Failover Trigger Input x.** When this signal changes state and the corresponding failover capability is enabled, a failover event is signaled.
GPE	O	**General Purpose Event.** Hot-plug general purpose event output
IOEXPINTN	I	**I/O Expander x Interrupt Input.** I/O expander interrupt
PARTxPERSTN	I	**Partition x Fundamental Reset.** Assertion of this signal initiates a partition fundamental reset in the corresponding partition.
PxACTIVEN	O	**Port x Link active status output.** See section Link Status on page 7-16.
PxLINKUPN	O	**Port x link up status output.** See section Link Status on page 7-16.

Table 13.3 GPIO Alternate Function Pins

When configured as an alternate function in the GPIOFUNC register, a pin behaves as the alternate function signal selected by the GPIOAFSEL register. If the alternate function signal is an input, the GPIO pin behaves as an input. If the alternate function signal is an output, the GPIO pin behaves as an output. The value of the alternate function pin may be determined at any time by reading the corresponding GPIOD register.

When an alternate function input signal is not enabled on any GPIO (or I/O expander for hot-plug signals), the alternate function signal is internally held in an inactive state. For example, not configuring PART0PERSTN as an alternate function on GPIO0 causes PART0PERSTN to be internally held high.

Non-Transparent Switch Operation

Notes

Overview

The term *non-transparent operation* is used in this document to describe the operation of the NT function. This chapter describes the PES32NT8xG2's non-transparent operation.

The PCI Express architectural model is one in which a root, typically the main CPU, is responsible for configuring a tree of endpoints (i.e., a hierarchy of virtual PCI buses). Once configured, any endpoint or root may initiate transactions. The root and endpoints share a common address space with routing configured in PCI-PCI bridges.

A limitation of the PCI Express architectural model is that it allows only a single root and that the root and all of the endpoints must share a common address space. This limitation may be overcome through the use of a non-transparent bridge (NTB). A non-transparent bridge allows two or more PCI Express hierarchies to be interconnected with one or more shared address windows between them. The connection is done via the NT Interconnect. Refer to section Non-Transparent Operation on page 1-8 for an introduction to this feature.

In the PES32NT8xG2, each PCI Express hierarchy is associated with a switch partition (see Chapter 5, Switch Partition and Port Configuration). Therefore, an NTB interconnects two or more switch partitions via the NT Interconnect. When a TLP is transferred across partitions, the source partition is the partition on which the TLP was received by the NTB, and the destination partition is the partition to which the TLP is destined.

The PES32NT8xG2 supports eight non-transparent functions (a.k.a., NT functions or NT endpoints). Each NT function appears as a PCI Express endpoint in the PCI Express hierarchy. The NT function is located in a partition's upstream port. A port configured to operate in one of the following modes contains an NT function:

- NT function
- NT with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions

Refer to section Switch Port Mode on page 5-5 for details on the port operating modes.

Base Address Registers (BARs)

Each NT-endpoint implements six Base Address Registers (BARs) labeled BAR 0 through BAR 5. Table 14.1 summarizes supported BAR configurations.

- All BARs may be configured to create 32-bit memory[1] windows between the PCI Express domain and the non-transparent interconnect[2].
- All BARs support direct address translation
- BAR 2 (or BAR 2/3 in 64-bit mode) supports direct address translation or lookup address translation
- BAR 4 (or BAR 4/5 in 64-bit mode) supports direct address translation or lookup address translation

Even and odd BARs may be paired to form 64-bit prefetchable memory space. The 4 KB configuration space associated with the NT endpoint may be mapped into 32-bit memory using BAR 0. BAR 0 and BAR 1 may be paired to map the 4 KB configuration space associated with the NT endpoint into 64-bit memory. See section Mapping NT Configuration Space to BAR 0 on page 14-4 for further details.

1. The NT function's BARs do not support I/O space.

2. Refer to section Non-Transparent Operation on page 1-7 for a description of the non-transparent interconnect.

Notes

BAR	Description
BAR 0	32-bit BAR that maps 4 KB NT-endpoint configuration registers Lower half of 64-bit BAR that maps 4 KB NT-endpoint configuration registers 32-bit BAR with direct address translation Lower half of 64-bit BAR with direct address translation
BAR 1	Upper half of 64-bit BAR that maps 4 KB NT-endpoint configuration registers 32-bit BAR with direct address translation Upper half of 64-bit BAR with direct address translation
BAR 2	32-bit BAR with direct or lookup table address translation Lower half of 64-bit BAR with direct or lookup address translation
BAR 3	32-bit BAR with direct address translation Upper half of 64-bit BAR with direct or lookup table address translation
BAR 4	32-bit BAR with direct or lookup table address translation Lower half of 64-bit BAR with direct or lookup table address translation
BAR 5	32-bit BAR with direct address translation Upper half of 64-bit BAR with direct or lookup table address translation

Table 14.1 NT Endpoint BARs

Each BAR has a corresponding setup register. For example, BAR 0 (BAR0) has an associated BAR Setup 0 (BARSETUP0) register. BAR setup registers allow a BAR to be enabled or disabled, control the operating mode of the BAR as well as advertised capabilities (e.g., size of the BAR window), and if applicable, control the address translation mechanism.

When an even BAR is configured for 64-bit operation, the odd BAR takes on the function of the upper 32-bits of the BADDR field of the even BAR. When an even and odd BAR are merged for 64-bit operation, the fields of the odd BAR Setup register remain read-write but have no functional effect on the operation of the device.

BAR Limit

Base Address Registers are used by a function to specify the amount of memory space required by the function. Software configures read-write BAR register bits with the base address of the allocated memory range. Since BARs specify the size of an aperture with read-only bits in the BADDR field, the PCI architecture only allows apertures to be requested that are a power of two in size. In many applications, it is desirable to allocate smaller apertures.

Notes

Associated with each BAR is a BAR Limit Address (BARLIMITx) register. The limit address specified by this register allows arbitrary control of the aperture size associated with a BAR. Using this capability, the *effective aperture size* may be set arbitrarily to any value, in 1 KB multiples, up to the power of two aperture size requested by the BAR.

- The lower 10-bits of the BARLIMITx register are reserved and assumed to be all ones by the hardware. Thus, the BAR limit address may be anywhere in the range from 0x3FF (i.e., 1KB) to the highest possible memory address.
- Setting the address limit of a BAR to a value less than the BAR base address effectively disables the BAR. The effective aperture size in this case is zero.
- Setting the address limit of a BAR to a value between the BAR base address and the BAR base address plus the power of two aperture size requested by the BAR sets the effective aperture to be from the BAR base address up to and including the BAR limit address.
- Setting the address limit of a BAR to a value greater than the BAR base address plus the power of two aperture size requested by the BAR disables the limit capability. The aperture and effective aperture in this case are both equal to the power of two size requested by the BAR.
- The default value of the BARLIMITx register causes the BAR limit address to point to the highest possible address, in effect disabling the effect of the BARLIMIT register
- The BARLIMIT0 register is ignored when BAR 0 is mapped to the NT endpoint's configuration space.

When an even BAR is configured for 64-bit operation, the odd BAR takes on the function of the upper 32-bits of the BADDR field of the even BAR. In this mode, the odd BARLIMITx register acts as the upper 32-bits of the LADDR field associated with the even BAR.

Figure 14.1 graphically illustrates the operation of the BAR limit address.

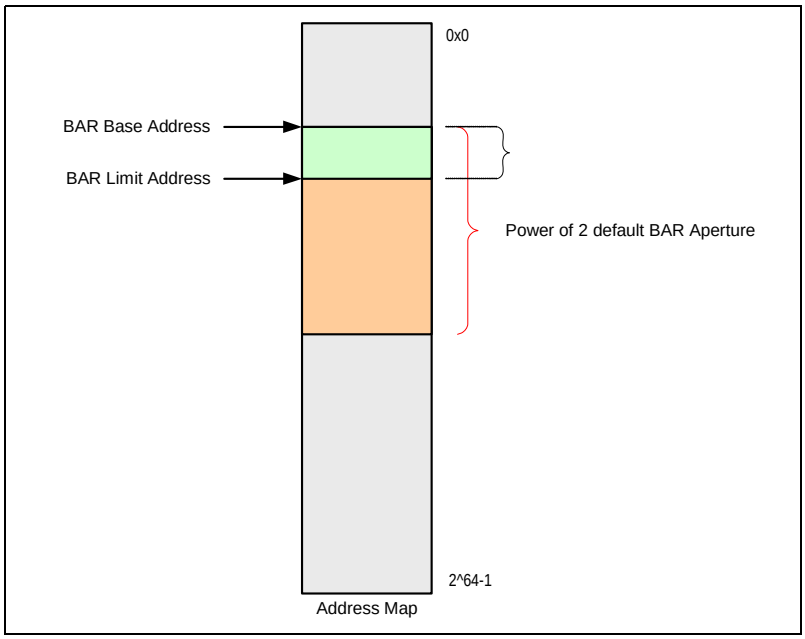

Figure 14.1 BAR Limit Operation

A received TLP whose address does not fall within the effective BAR aperture of a BAR is not processed by the BAR. A received TLP whose address falls within a BAR aperture, but outside the *effective BAR aperture*, is treated as an unsupported request by the NT function. The behavior for TLPs whose address falls within the effective address of multiple BARs is undefined.

Notes

Mapping NT Configuration Space to BAR 0

As mentioned above, the 4 KB configuration space associated with the NT endpoint may be mapped into 32-bit memory using BAR 0. BAR 0 and BAR 1 may be paired to map the 4 KB configuration space associated with the NT endpoint into 64-bit memory.

Mapping NT configuration space to BAR 0 allows these configuration space registers to be accessed via memory read or writes. Mapping NT configuration space to BAR 0 requires that the MODE field be set appropriately in the BARSETUP0 register. When NT configuration space is mapped to BAR 0, the size of the BAR aperture is automatically set to 4 KB and the BARLIMIT0 register is ignored.

When the NT function's configuration space is mapped to BAR 0, it is recommended that this configuration space be placed in non-prefetchable memory space, as some registers may generate side-effect actions when accessed. In addition, memory read or write requests to BAR 0 must specify a length of 1 DWord. Violating this last requirement produces undefined results.

Note: The NT function's configuration space layout follows little-endian convention. Software executing on a big-endian system should take this into account when accessing the NT function's configuration space memory-mapped to BAR 0.

TLP Translation

Direct Address Translation

All BARs may be configured to support direct address translation. Figure 14.2 illustrates the address translation process for a BAR configured as a memory address window with direct address translation.

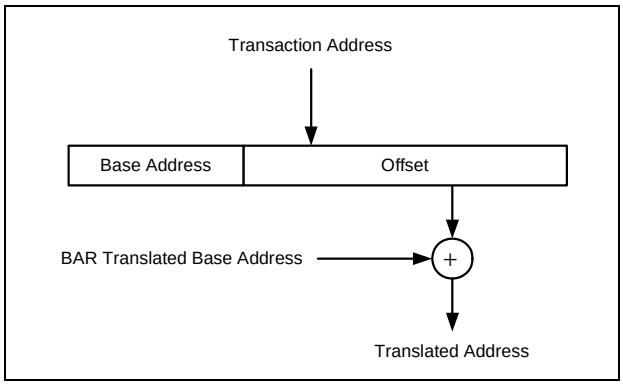

Figure 14.2 Direct Address Translation

The address of a TLP that falls within the effective BAR aperture of a BAR may be divided into a base address and an offset. The base address is equal to the value programmed in BAR BADDR field bits that are read/write. The offset address corresponds to address bits that are not part of the base address.

Associated with each BAR is a translated base address. The translated base address is a 64-bit quantity. The upper 32-bits are set to zero when the translated base address is less than or equal to 0xFFFF_FFFF (i.e., lower 4 GB). The translated base address is always DWord aligned. Therefore, the bottom two bits are always zero.

The translated address for the TLP is equal to the sum of the TLP offset address with the corresponding translated base address field. Following formation of the translated address, the TLP header size is adjusted accordingly:

- If the upper 32-bits of the 64-bit address are all zero, then a 3 DWord header is used.
- If the upper 32-bits of the 64-bit address are non-zero, then a 4 DWord header is used.

Notes

The destination partition of the translated TLP is specified by the Translated Partition (TPART) field in the corresponding BARSETUPx register. If the destination partition associated with the translated TLP is invalid (e.g., there is no NT endpoint associated with the destination partition, the destination partition is not in the active state, or the destination partition is the same as the partition on which the TLP was received), then the TLP is treated as an unsupported request by the NT endpoint that received the request.

Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on the programming of the translated base address.

Lookup Table Address Translation

BARs two and four may be configured to support lookup table address translation.

- BAR two may be configured to support either a 12-entry or 24-entry lookup table.
- BAR four only supports 12-entry lookup table address translation. Configuring BAR four for 24-entry lookup table address translation produces undefined results.
- If both BARs two and four are configured for lookup table address translation, then BAR two only supports a 12-entry lookup table. Configuring BAR two for 24-entry lookup table address translation while BAR four is configured for 12-entry lookup table address translation produces undefined results.

Figure 14.3 illustrates the address translation process for a BAR configured as a memory address window with lookup table address translation.

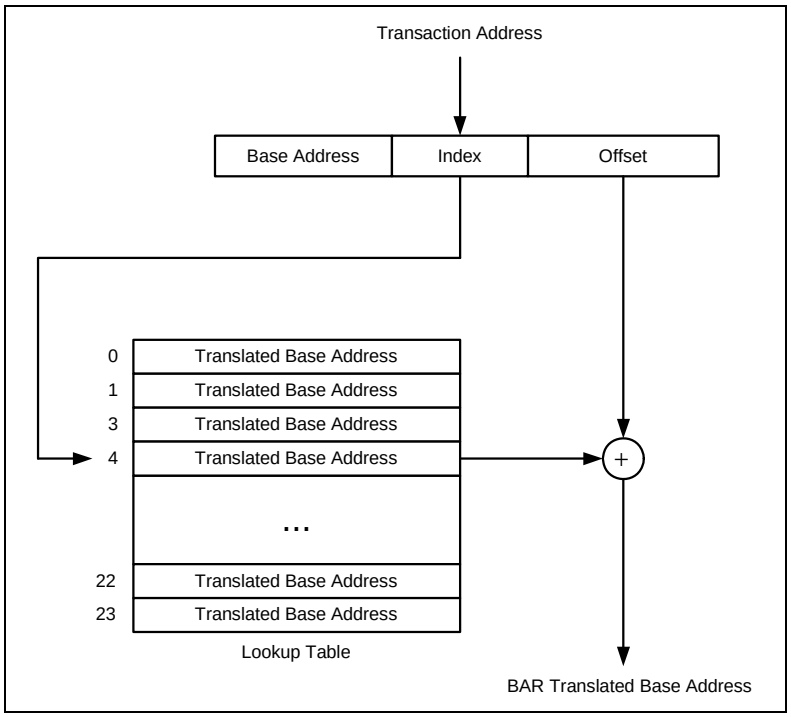

Figure 14.3 Lookup Table Translation

The address of a TLP that falls within the effective BAR aperture of a BAR may be partitioned into a base address, index, and offset.

- The base address is equal to the value programmed in the BAR BADDR field bits that are read/write.
- If the BAR is configured for a 12-entry lookup table, then the index corresponds to the next 4-bits of the address.
- If the BAR is configured for a 24-entry lookup table, then the index corresponds to the next 5-address bits.
- The offset address corresponds to address bits that are not part of the base address or index.

Notes

When the BAR is configured to operate as an address window with lookup table address translation, valid values for the SIZE field in the corresponding BARSETUPx register are 14 through 37 (values greater than 16 require a 64-bit BAR). Setting the SIZE field outside this range produces undefined results.

Associated with a BAR configured to use lookup table address translation is a 12 or 24-entry lookup table. The format of a table entry is shown in Figure 14.4.

- The translated base address field plays the same role as the translated base address in direct address translation.
- The partition field specifies the destination partition associated with the translated TLP.
- The valid field indicates if the table entry is valid.

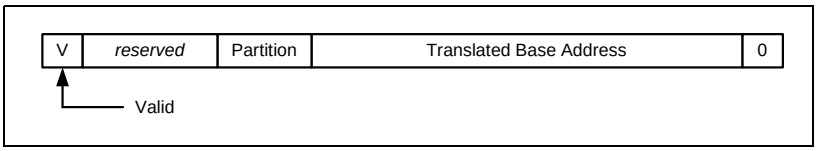

Figure 14.4 Lookup Table Entry Format

When a TLP is processed by a BAR that is configured for lookup table address translation, the index portion of the TLP address is used as an index into the lookup table.

- When BAR two is configured for a 24-entry lookup table translation, the index portion of the TLP address is 5-bits. These 5-bits allow indexing of up to 32 segments in the BAR, numbered 0 to 31. Segments 0 to 23 are each associated with a corresponding lookup table entry (e.g., segment 0 is associated with lookup table entry 0, segment 1 is associated with lookup table entry 1, etc.) Segments 24 to 31 are not associated with any lookup table entry and are considered out of bounds. TLPs whose index maps to an out-of-bound segment are treated as unsupported requests.
- When a BAR two is configured for 12-entry lookup table translation, the index portion of the TLP address is 4-bits. These 4-bits allow indexing of up to 16 segments in the BAR, numbered 0 to 15. Segments 0 to 11 are each associated with a lookup table entry in the lower half of the table (e.g., segment 0 is associated with lookup table entry 0, segment 1 is associated with lookup table entry 1, etc.) Segments 12 to 15 are not associated with any lookup table entry and are considered out of bounds. TLPs whose index maps to an out-of-bound segment are treated as unsupported requests.
- When a BAR four is configured for 12-entry lookup table translation, the index portion of the TLP address is 4-bits. These 4-bits allow indexing of up to 16 segments in the BAR, numbered 0 to 15. Segments 0 to 11 are each associated with a lookup table entry in the upper half of the lookup table (e.g., segment 0 is associated with lookup table entry 12, segment 1 is associated with lookup table entry 13, etc.) Segments 12 to 15 are not associated with any lookup table entry and are considered out of bounds. TLPs whose index maps to an out-of-bound segment are treated as unsupported requests.

In addition, if the table entry pointed to by the index is not valid (i.e., the valid bit in the entry is not set), then the TLP is treated as an unsupported request. If the table entry pointed to by the index is valid, then the translated address for the TLP is formed by adding the offset field of the TLP address to the translated base address associated with the table entry.

The translated base address is a 64-bit quantity allowing the translated window to be located anywhere within the address space of the destination PCI Express domain. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on the programming of the translated base address.

The size of a PCI Express TLP header varies depending on the size of the address. A 32-bit address has a three DWord header while a 64-bit address has a four DWord header. Following formation of the translated address, the TLP header size is adjusted accordingly.

Notes

The destination partition associated with the translated TLP is specified by the partition field in the lookup table entry. If the partition associated with the translated TLP is invalid (e.g., there is no NT endpoint associated with the destination partition, the destination partition is not in the active state, or the destination partition is the same as the original partition), then the TLP is treated as an unsupported request by the NT endpoint that received the request.

The behavior of a TLP whose translated address falls within the BAR aperture(s) of the NT function in the destination partition is undefined. Tables 14.2 and 14.3 summarize the parameters associated with a 12-entry and 24-entry lookup table. Page size refers to the size of the address space translated by each table entry.

BARSETUPx SIZE Field	Aperture Size	Page Size	Base Address (bits)[1]	Index (bits)	Offset (bits)
14	16 KB	1 KB	[63:14]	[13:10]	[9:0]
15	32 KB	2 KB	[63:15]	[14:11]	[10:0]
16	64 KB	4 KB	[63:16]	[15:12]	[11:0]
17	128 KB	8 KB	[63:17]	[16:13]	[12:0]
18	256 KB	16 KB	[63:18]	[17:14]	[13:0]
19	512 KB	32 KB	[63:19]	[18:15]	[14:0]
20	1 MB	64 KB	[63:20]	[19:16]	[15:0]
21	2 MB	128 KB	[63:21]	[20:17]	[16:0]
22	4 MB	256 KB	[63:22]	[21:18]	[17:0]
23	8 MB	512 KB	[63:23]	[22:19]	[18:0]
24	16 MB	1 MB	[63:24]	[23:20]	[19:0]
25	32 MB	2 MB	[63:25]	[24:21]	[20:0]
26	64 MB	4 MB	[63:26]	[25:22]	[21:0]
27	128 MB	8 MB	[63:27]	[26:23]	[22:0]
28	256 MB	16 MB	[63:28]	[27:24]	[23:0]
29	512 MB	32 MB	[63:29]	[28:25]	[24:0]
30	1 GB	64 MB	[63:30]	[29:26]	[25:0]
31	2 GB	128 MB	[63:31]	[30:27]	[26:0]
32	4 GB	256 MB	[63:32]	[31:28]	[27:0]
33	8 GB	512 MB	[63:33]	[32:29]	[28:0]
34	16 GB	1 GB	[63:34]	[33:30]	[29:0]
35	32 GB	2 GB	[63:35]	[34:31]	[30:0]
36	64 GB	4 GB	[63:36]	[35:32]	[31:0]
37	128 GB	8 GB	[63:37]	[36:33]	[32:0]

Table 14.2 12-Entry Lookup Table Parameters

1. Assumes 64-bit TLP address. If only 32-bits are used then bits [31:x] are used.

Notes

BARSETUPx SIZE Field	Aperture Size	Page Size	Base Address (bits)[1]	Index (bits)	Offset (bits)
14	16 KB	512 B	[63:14]	[13:9]	[8:0]
15	32 KB	1 KB	[63:15]	[14:10]	[9:0]
16	64 KB	2 KB	[63:16]	[15:11]	[10:0]
17	128 KB	4 KB	[63:17]	[16:12]	[11:0]
18	256 KB	8 KB	[63:18]	[17:13]	[12:0]
19	512 KB	16 KB	[63:19]	[18:14]	[13:0]
20	1 MB	32 KB	[63:20]	[19:15]	[14:0]
21	2 MB	64 KB	[63:21]	[20:16]	[15:0]
22	4 MB	128 KB	[63:22]	[21:17]	[16:0]
23	8 MB	256 KB	[63:23]	[22:18]	[17:0]
24	16 MB	512 KB	[63:24]	[23:19]	[18:0]
25	32 MB	1 MB	[63:25]	[24:20]	[19:0]
26	64 MB	2 MB	[63:26]	[25:21]	[20:0]
27	128 MB	4 MB	[63:27]	[26:22]	[21:0]
28	256 MB	8 MB	[63:28]	[27:23]	[22:0]
29	512 MB	16 MB	[63:29]	[28:24]	[23:0]
30	1 GB	32 MB	[63:30]	[29:25]	[24:0]
31	2 GB	64 MB	[63:31]	[30:26]	[25:0]
32	4 GB	128 MB	[63:32]	[31:27]	[26:0]
33	8 GB	256 MB	[63:33]	[32:28]	[27:0]
34	16 GB	512 MB	[63:34]	[33:29]	[28:0]
35	32 GB	1 GB	[63:35]	[34:30]	[29:0]
36	64 GB	2 GB	[63:36]	[35:31]	[30:0]
37	128 GB	4 GB	[63:37]	[36:32]	[31:0]

Table 14.3 24-Entry Lookup Table Parameters

1. Assumes 64-bit TLP address. If only 32-bits are used then bits [31:x] are used.

The lookup table for both BARs is configured using the Lookup Table Offset (LUTOFFSET), Lookup Table Lower Data (LUTLDATA), Lookup Table Middle Data (LUTMDATA), Lookup Table Upper Data (LUTUDATA) registers.

- Fields associated with lookup entries are modified by accessing the LUTLDATA, LUTLMDATA and LUTUDATA registers. A read from one of these registers returns the field values of the lookup table entry pointed to by the LUTOFFSET register. Similarly, a write updates the fields of the lookup entry pointed to by the LUTOFFSET register.
- The BAR field in the LUTOFFSET register selects the lookup table associated with the corresponding BAR while the INDEX field in the LUTOFFSET field selects the lookup table entry.

The state of the lookup table is preserved across all resets except a switch fundamental reset. Following a switch fundamental reset, the state of all lookup table fields except the Valid (V) field is undefined. Following a switch fundamental reset, the Valid field is cleared in all entries.

Notes

ID Translation

PCI Express TLPs may be categorized into request TLPs and completion TLPs.

- A request TLP is a packet used to initiate a transaction.
- A completion TLP is a packet used to terminate, or partially terminate a transaction sequence.

Request TLPs contain a requester ID field that defines the unique PCI Express identifier associated with the requester that generated the request TLP.

Completion TLPs contain both a requester ID field and a completer ID field. The completer ID field defines the unique PCI Express identifier associated with the completer that generated the completion TLP.

A PCI Express identifier consists of a 16-bit quantity that is unique for each function in a PCI Express hierarchy. The 16-bit quantity may be interpreted as an 8-bit bus number, 5 bit device number, and 3 bit function number.

This section describes the ID matching, processing, and translation performed by an NT endpoint.

NT Mapping Table

Associated with the switch is a 64-entry Non-Transparent (NT) Mapping table. The NT Mapping table is used to perform ID translation and ID based protection.

The NT Mapping table is a global table shared by all ports configured for NT operation.

The format of the NT Mapping table is shown in Figure 14.5 and the fields are described in Table 14.4.

Mapping Table Entry									
0	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
1	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
2	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
3	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
4	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
5	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V
⋮									
63	RNS	CNS	ATP	Reserved	PART	BUS	DEV	FUNC	V

Figure 14.5 NT Mapping Table

Bit Field	Field Name	Description
0	V	**Valid**. This bit is set if the mapping table is valid.
3:1	FUNC	**Function**. This field contains the mapping table entry PCI Express function number.
8:4	DEV	**Device**. This field contains the mapping table entry PCI Express device number.
16:9	BUS	**Bus**. This field contains the mapping table entry PCI Express bus number.
19:17	PART	**Partition**. This field contains the mapping table entry partition number.

Table 14.4 NT Mapping Table Field Description (Part 1 of 2)

Notes

Bit Field	Field Name	Description
29	ATP	**Address Type Processing.** This field specifies the processing of the address type (AT) field on request TLPs. Refer to section Address Type Processing on page 14-15.
30	CNS	**Completion No Snoop Processing**. This field specifies the no snoop processing on completion TLPs. Refer to section No Snoop Processing on page 14-14.
31	RNS	**Request No Snoop Processing**. This field specifies the no snoop processing on request TLPs. Refer to section No Snoop Processing on page 14-14.

Table 14.4 NT Mapping Table Field Description (Part 2 of 2)

The state of the NT Mapping table is preserved across all resets except a switch fundamental reset. Following a switch fundamental reset, the state of all NT Mapping table fields except the Valid (V) field is undefined. Following a switch fundamental reset, the Valid field is cleared in all entries. The mapping tables may be initialized by using the NT Mapping Table Address (NTMTBLADDR) and NT Mapping Table Data (NTMTBLDATA) registers.

- To access a mapping table entry, the NT Mapping Table Address (ADDR) field in the port's NTMTBLADDR register is initialized with the partition NT Mapping table entry to be accessed.
- Reading from the NT Mapping Table Data (NTMTBLDATA) register returns the value of the fields of the corresponding partition NT Mapping table entry pointed to by the ADDR field in the NTMTBLADDR register.
- Writing to the NTMTBLDATA register causes the fields in the corresponding partition NT Mapping table entry pointed to by the ADDR field in the NTMTBLADDR register to be updated with the value written.
- The NTMTBLDATA register must be accessed using DWORD operations. The behavior for all other access sizes is undefined.
- The NTMTBLADDR and NTMTBLDATA registers can't be accessed via the Global Address Space Access registers (i.e., GASAADDR and GASADATA), or via the Extended Configuration Space Access registers (i.e., ECFGADDR and ECFGDATA).
  - All other methods to access these registers are allowed (i.e., PCI Express configuration requests that target the registers directly, memory read/write requests that map to the NT configuration space, EEPROM, and SMBus).
- Causing an NT Mapping table protection violation results in the NT Mapping Table Access Error (ERR) bit to be set in the NT Mapping Table Status (NTMTBLSTS) register.

Since the NT Mapping table is global and shared by all partitions, the PES32NT8xG2 supports NT Mapping table protection and virtualization.

- Located in the Switch Control and Status register space is an NT Mapping Table Protection (NTMTBLPROTx) register for each partition.
- NT Mapping Table Base (TBLBASE) and NT Mapping Table Limit (TBLLIMIT) fields in the NTMTBLPROTx register control how partition NT Mapping table accesses are translated into physical NT Mapping table accesses.
- The translation process is shown in Figure 14.6.

Notes

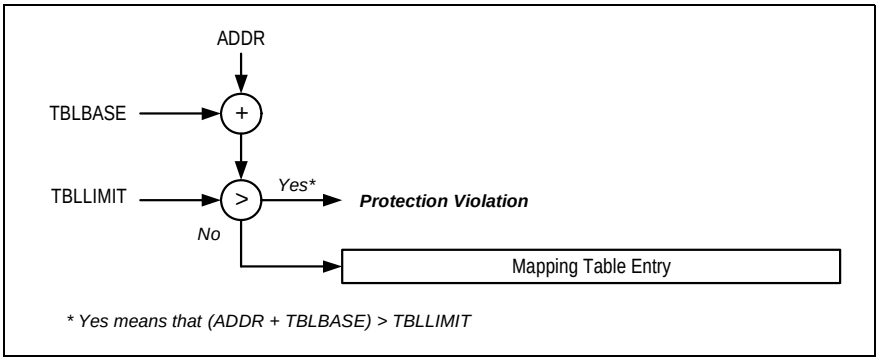

Figure 14.6 NT Table Partitioning

- The physical NT Mapping table entry accessed is equal to the sum of the partition NT Mapping table entry, specified by the ADDR field in the NTMTBLADDR register, with the TBLBASE field in the NTMTBLPROT register associated with the partition. If the resulting physical NT Mapping table entry address is less than or equal to the value in the TBLLIMIT field in the NTMTBLPROT field associated with the partition, then the read or write access is performed on the physical NT Mapping table entry. If the value of the resulting NT Mapping table entry address is greater than the NTBLLIMIT field, then a protection violation is signaled.
- Located in each NTMTBLPROT register is a Partition Blocking Vector (PARTBLOCK). Associated with each partition in the switch is a corresponding bit in the PARTBLOCK vector. When a write to the NTMTBLDATA register is performed with a PART field value whose corresponding bit is set in the PARTBLOCK vector, then a protection violation is signaled.
- A protection violation during an NTMTBLDATA read operation results in a value of zero being returned in all fields and setting of the NT Mapping Table Access Error (ERR) bit in the NT Mapping Table Status (NTMTBLSTS) register.
- A protection violation during an NTMTBLDATA write operation causes the write operation to be ignored (i.e., no table entry or register field is actually updated) and setting of the NT Mapping Table Access Error (ERR) bit in the NT Mapping Table Status (NTMTBLSTS) register.

Following a switch fundamental reset, NT Mapping table protection and virtualization is disabled.

- All partitions may access all 64 physical NT Mapping table entries.
- The virtual NT Mapping table address of all partitions is equal to the physical NT Mapping table address.
- A partition may update any NT Mapping table entry with any PART field value.

Note: NT Mapping table programming (i.e., programming of table entries or table protection settings) must always be done via the same interface (e.g., PCI Express, Slave SMBus, or EEPROM). For example, programming NT Mapping table protection via the EEPROM followed by programming the NT Mapping table entries via PCI Express is not allowed.

Request ID Translation

Request TLPs contain a requester ID field that defines the unique PCI Express identifier associated with the requester that generated the request TLP. When a request TLP is received by an NT endpoint that is to be routed on the NT interconnect (i.e., the TLP hits an NT endpoint BAR aperture and has a valid translation), a requester ID lookup and translation are performed.

Notes

The lookup is performed by matching the 16-bit requester ID in the request TLP along with the partition associated with the NT endpoint to entries in the NT Mapping table. If a lookup match is not found, then the TLP is treated as an unsupported request. Otherwise, the TLP is processed normally as described in this section.

- 16-bit requester ID is compared to the 16-bit value in each NT Mapping table consisting of the BUS, DEV, and FUNC fields regardless of the requester ID interpretation. A requester ID match occurs when the 16-bit value in the requester ID field of the TLP matches an NT Mapping table entry.
- The partition associated with a request TLP is the partition ID associated with the NT endpoint which received the request TLP. A partition match occurs when the partition associated with a request TLP matches the PART field of an NT Mapping table entry.
- A lookup match occurs for a request TLP when a NT Mapping table entry exists that is valid (i.e., the V bit is set) and has both a requester ID match as well as a partition match.
- The behavior of a request TLP with multiple lookup matches is undefined. Multiple lookup matches are the result of an invalid configuration.

PCI Express allows a function to expand the number of supported outstanding requests requiring completions beyond 256 through the use of phantom function numbers. When phantom function numbers are enabled, the Tag field in the TLP header may be logically expanded by using unimplemented function numbers. These unimplemented function numbers are referred to as phantom function numbers. A requester that uses phantom function numbers when communicating with the NT endpoint requires a unique NT Mapping table entry for each phantom function number.

The requester ID field associated with a request TLP that has a lookup match is translated as shown in Figure 14.7.

- The bus field is replaced by the captured bus number of the NT endpoint associated with the partition of the translated TLP.
- Bit 4 of the device field is set to one.
- The lower four bits of the device field and the function field are replaced with the mapping table match entry number.

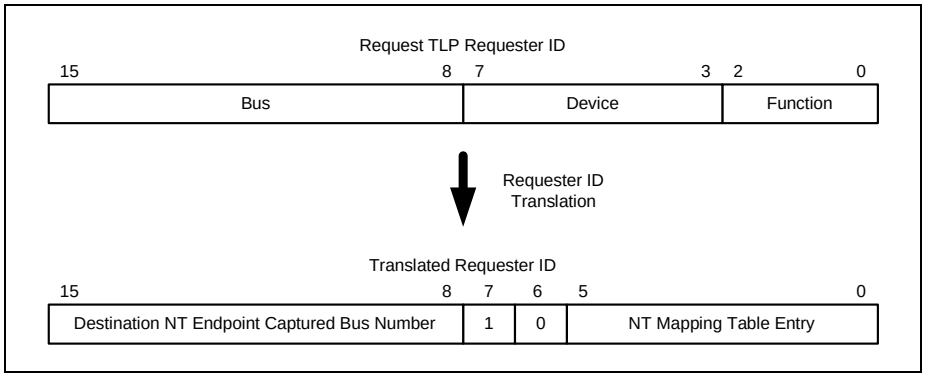

Figure 14.7 Request TLP Requester ID Translation

One function of request TLP ID translation process is that it allows a corresponding reverse translation to occur for completions. This reverse translation is described in section Completion ID Translation on page 14-13. Posted requests (e.g., memory writes) have no corresponding completions. Therefore, the primary role of the NT Mapping table lookup for these TLPs is to provide a form of protection (i.e., only authorized requesters are allowed to issue TLPs that map onto the NT interconnect). When the ID Protection Check Disable (IDPROTDIS) bit in the Endpoint Control (NTCTL) register is set, the NT table lookup for posted requests is skipped and all posted requests are allowed to map onto the NT interconnect regardless of requester ID.

The requester ID field associated with a posted request TLP is translated as shown in Figure 14.8 when the IDPROTDIS bit is set in the NTCTL register.

Notes

- The bus field is replaced by the captured bus number of the NT endpoint associated with the partition of the translated TLP.
- The device and function fields are replaced by the value 0x3. This corresponds to device 0, function 3.

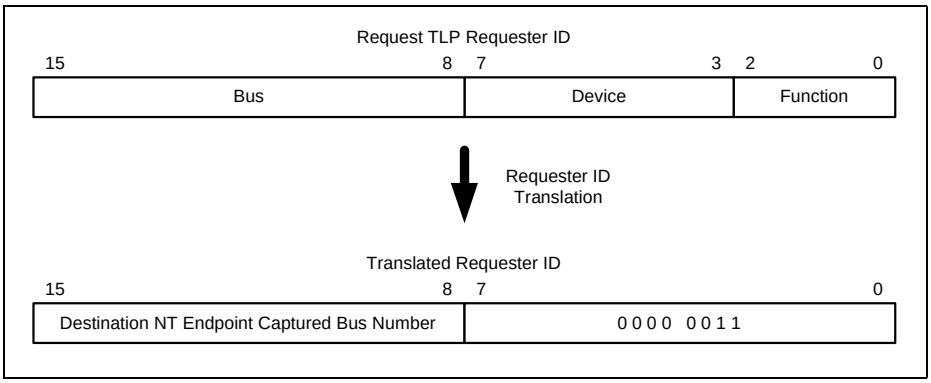

Figure 14.8 Request TLP Requester ID Translation

If the Bus Master Enable (BME) bit is cleared in the PCI Command (PCICMD) register of the NT endpoint associated with the translated TLP (i.e., in the destination partition), then the request is treated as an unsupported request by the NT endpoint that received the request (i.e., in the source partition).

Completion ID Translation

Completion TLPs contain both a requester ID field and a completer ID field. The completer ID field defines the unique PCI Express identifier associated with the completer that generated the completion TLP.

When a completion TLP is received and claimed[1] by an NT endpoint the following processing is performed.

- If the TLP's requester ID function field matches the requester ID used by the NT function during punch-through requests, and a punch-through configuration is in progress, the TLP is accepted by the NT function and processed as described in section Punch-Through Configuration Requests on page 14-18. Otherwise, if a punch-through configuration is not in progress, the TLP is handled as an unexpected completion.
- If the TLP's requester ID function field matches the NT function's bus/device/function assignment within the PCI Express hierarchy, then the TLP is handled as an unexpected completion by the NT function. Refer to section Error Detection and Handling by the NT Function on page 14-25.
- Otherwise, the 8-bit value consisting of the device and function fields is extracted from the requester ID field. The upper two bits are separated and the lower six bits form an NT Mapping table entry index. If the upper two bits are 0b10 and the NT Mapping table entry index points to a valid entry in the NT Mapping table, the processing described is performed. Otherwise, the TLP is handled as an unexpected completion by the NT function.

The requester ID of the translated completion TLP is formed as follows.

- The requester ID bus field is replaced by the NT Mapping table bus field.
- The requester ID device field is replaced by the NT Mapping table device field.
- The requester ID function field is replaced by the NT Mapping table function field.

The completer ID of the translated completion TLP is equal to the bus, device, and function of the NT endpoint associated with the partition of the translated completion TLP (i.e., the NT function that emits the TLP).

1. A completion TLP is claimed by the NT function when the TLP's requester ID matches the NT function's bus/device/function assignment within the PCI Express hierarchy, when it matches a valid entry in the NT function's mapping table (as described below), or when it matches the NT function's requester ID used for punch-through requests (refer to section Punch-Through Configuration Requests on page 14-18).

Notes

If the Completion Enable (CPEN) bit is cleared in the NTCTL register of the NT endpoint associated with the translated TLP (i.e., in the destination partition), then the completion is silently dropped by the NT endpoint that received the request (i.e., in the source partition). Note that this only applies to translated completion TLPs and not to completion TLPs generated by the NT function itself (e.g., in response to a configuration request).

Note that this bit must be set in the NT function of a source partition prior to sending non-posted requests across the NTB, to allow the completions generated in the destination partition to be emitted back into the source partition.

Also, note that a reset or hot reset of the NT function causes the CPEN bit to be cleared, in effect preventing translated completions (which are possibly associated with requests received before the reset or hot reset event) from being emitted by the NT function.

Refer to section Unexpected Completions on page 14-29 for further details on error conditions associated with completion ID translation.

Requester ID Capture Register

In order to program the NT Mapping Table, it is necessary that the requester IDs of agents in the PCI Express hierarchy be known. In most systems, the assignment of requester IDs is done dynamically during enumeration, and is therefore dependent on the organization of the PCI Express hierarchy.

To facilitate programming of the NT Mapping Table, the NT function in the PES32NT8xG2 contains a proprietary register that may be used by a PCI Express agent to determine its requester ID in the PCI Express hierarchy.

This register is the Requester ID Capture register (REQIDCAP), located in the configuration space of the NT function. When an agent issues a configuration read request to the REQIDCAP register, a completion is generated. The completion's data payload is 1 DW, with the upper 16 bits set to zero and the lower 16 bits reflecting the requester ID of the agent that issued the configuration read request. The same operation is supported when the REQIDCAP register is accessed via the NT BAR 0/1, when the BAR is mapped to the NT configuration space. Writes to the REQIDCAP register are ignored.

TLP Attribute Processing

The NT function supports processing of the No Snoop attribute for request or completions TLPs that cross the NTB. It also supports processing of the Address Type field for requests that cross the NTB. The NT function does not support processing of the Relaxed Ordering attribute (i.e., this attribute is not modified in TLPs that cross the NTB). Therefore, TLPs that cross the NTB and have the Relaxed Ordering attribute set are understood to be relaxed ordered TLPs in both the source and destination partitions.

The Enable Relaxed Ordering (ERO) bit in the NT function's PCI Express Device Control register (PCIEDCTL) may be set or cleared by software, but it has no effect on the hardware. When this bit is cleared by software, the user must ensure that no translated TLPs emitted by the NT function have the Relaxed Ordering attribute set (i.e., TLPs received by an NT function in another partition and emitted by the NT function whose ERO bit is cleared).

No Snoop Processing

The No Snoop attribute in the header of request TLPs indicates whether hardware enforced cache coherence is expected. Some platforms lack the ability to control the no snoop attribute for generated requests. Therefore, the PES32NT8xG2 provides the ability to modify the No Snoop attribute for TLPs flowing through the NT interconnect.

When an NT table lookup is performed for request TLPs (described in section Request ID Translation on page 14-11), the Request No Snoop Processing (RNS) field in the matching NT Mapping table entry is examined. If the RNS bit is set, then the No Snoop attribute in the translated TLP is inverted. If the RNS bit is cleared, then the No Snoop attribute in the translated TLP is equal to that of the received request TLP (i.e., the No Snoop attribute is not modified).

Notes

If the Completion No Snoop Processing (CNS) field in the NT Mapping entry corresponding to the extracted NT Mapping table index (see section Completion ID Translation on page 14-13) is set, then the No Snoop attribute in the translated TLP is inverted. If the CNS bit is cleared, then the No Snoop attribute in the translated TLP is equal to that of the received completion TLP (i.e., the No Snoop attribute is not modified).

The NT function supports processing of the No Snoop attribute for request or completions TLPs that cross the NTB. It also supports processing of the Address Type field for requests that cross the NTB.

The NT function does not support processing of the Relaxed Ordering attribute (i.e., this attribute is not modified in TLPs that cross the NTB). Therefore, TLPs that cross the NTB and have the Relaxed Ordering attribute set are understood to be relaxed ordered TLPs in both the source and destination partitions.

The Enable Relaxed Ordering (ERO) bit in the NT function's PCI Express Device Control register (PCIEDCTL) may be set or cleared by software, but it has no effect on the hardware. When this bit is cleared by software, the user must ensure that no translated TLPs emitted by the NT function have the Relaxed Ordering attribute set (i.e., TLPs received by an NT function in another partition and emitted by the NT function whose ERO bit is cleared).

Address Type Processing

As described in the Address Translation Services Specification, March 8, 2007, PCI-SIG and the PCI Express Base Specification Revision 2.1, the Address Type (AT) field in the header of a memory read or memory write TLP indicates the type of address in the TLP (i.e., untranslated, translation request, translated).

The NT endpoint does not support Address Translation Services (ATS) as defined by the PCI-SIG, but it has the ability to modify the AT field for TLPs that cross the NTB. This allows the NTB to receive TLPs with translated addresses (i.e., AT field set to 'translated') in a source partition and emit them as TLPs with translated or untranslated addresses in the destination partition, or vice-versa.

Address type processing is only applied to memory read or write TLPs whose AT field is set to 'translated' or 'untranslated'. Address type processing is not applied to TLPs whose AT field is set to 'translation request'. Address type processing is performed based on the Address Type Processing (ATP) field in the matching NT Mapping Table entry, or based on the ATP field in the NTCTL register when the ID Protection Check Disable (IDPROTDIS) bit in the NTCTL register is set.

When an NT table lookup is performed for a request TLP (described in section Request ID Translation on page 14-11), the Address Type Processing (ATP) field in the matching NT Mapping table entry is examined. If the ATP field is set to 0x1, then the AT field is set to 'translated' in the TLP emitted by the NT endpoint in the destination partition. Otherwise, the AT field is set to 'untranslated' in the TLP emitted by the NT endpoint in the destination partition.

Note that completion TLPs always have the AT attribute set to zero and are not subject to address type field modification.

NT Multicast

The NT function supports non-transparent (NT) multicast, which allows a TLP received by a port in a switch partition that contains an NT function to be multicasted to other ports of the switch, across partitions. NT multicast is described in detail in Chapter 17, Multicast.

Inter-Domain Communications

The NT inter-domain communications capability structure provides facilities for supporting communications between processors in different PCI Express domains.

The NT inter-domain communications capability provides the following facilities:

- Doorbell registers
- Message registers

Notes

Doorbell Registers

Doorbells facilitate event signaling between partitions. Associated with each NT endpoint are one 32-bit outbound doorbell register and one 32-bit inbound doorbell register. An outbound doorbell request from an NT endpoint is initiated by writing a one to the corresponding bit in the Outbound Doorbell Set (OUTDBELLSET) register.

- Outbound doorbell requests are edge-triggered, meaning that the action of writing a one to a bit in the OUTDBELLSET register causes the corresponding outbound doorbell to be signaled. Writing a zero to any bit in the OUTDBELLSET register has no effect on the doorbell request. Reading from the OUTDBELLSET register always returns 0x0.

An inbound doorbell request to the NT endpoint results in the setting of the corresponding bit in the Inbound Doorbell Status (INDBELLSTS) register.

- The setting of a bit in the INDBELLSTS register may be used to generate an NT endpoint interrupt.
- Individual bits in the INDBELLSTS register may be masked from generating an interrupt by setting the corresponding bit in the Inbound Doorbell Mask (INDBELLMSK) register.

The logical operation of doorbells is illustrated in Figure 14.9.

For each of the 32 outbound doorbell request, the requests from all partitions are logically OR-ed together to form a global doorbell request. This global doorbell request is then used to initiate inbound doorbell requests to each of the partitions.

An outbound doorbell may initiate inbound doorbell requests in one or more partitions. All inbound doorbell requests share the same index. In other words, writing a one to bit position 8 in the OUTDBELLSET register may initiate an inbound doorbell request in multiple partitions, but each inbound doorbell request will be associated with the same bit position (e.g., position 8) as that of the outbound request.

Associated with each outbound doorbell is a Global Outbound Doorbell Mask (GODBELLMSKx) register that contains a bit corresponding to each partition. When a bit in this register is set, outbound doorbell requests from the corresponding partition are masked. For example, setting bit 7 in the GODBELLMSK4 register masks doorbell 4 requests from partition 7.

- When a doorbell request from a partition is masked, the state of the doorbell in the corresponding partition plays no role in determining the state of the global doorbell status.

A global doorbell request results in the initiation of corresponding inbound doorbell requests to all unmasked partitions.

- A global doorbell request may be masked to a partition by setting the corresponding partition bit in the Global Inbound Doorbell Mask (GIDBELLMSKx) register.
- When an inbound doorbell request is masked to a partition, the state of the global doorbell status plays no role in determining the state of the corresponding inbound doorbell request in that partition.

Notes

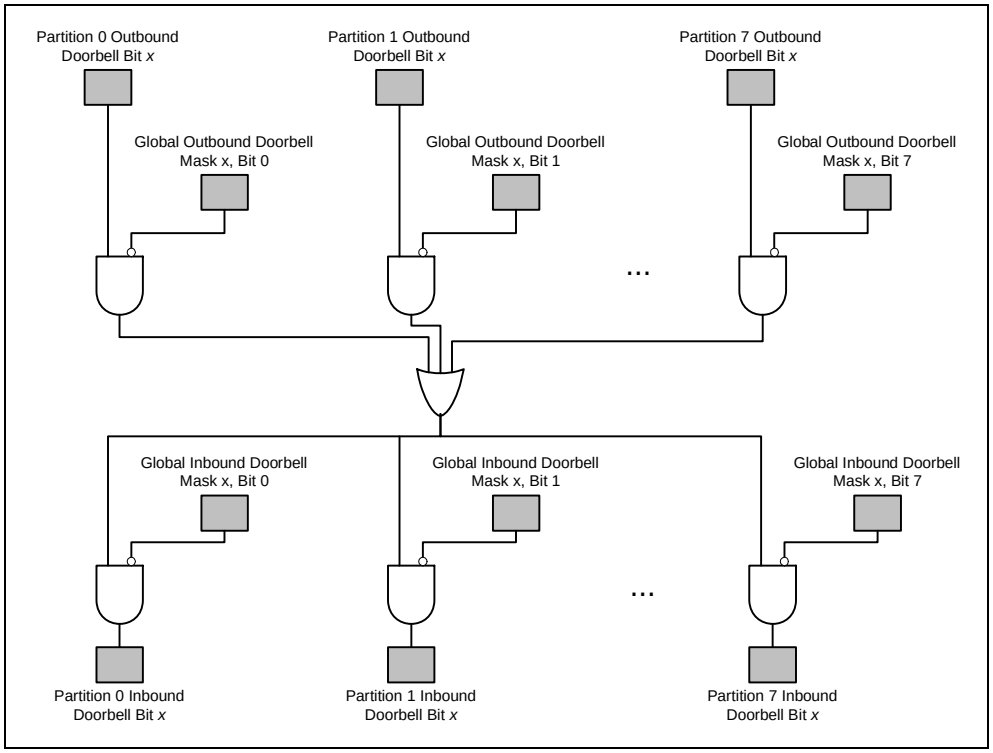

Figure 14.9 Logical Representation of Doorbell Operation

Bits in the Inbound Doorbell Status (INDBELLSTS) register may be used to generate NT endpoint interrupts. A bit in the INDBELLSTS register may be masked from generating an interrupt by setting the corresponding bit in the Inbound Doorbell Mask (INDBELLMSK) register. The global doorbell request status may be determined by reading the Global Doorbell Status (GDBELLSTS) register (not shown in Figure 14.9). A bit in the GDBELLSTS register is set if there exists a pending unmasked inbound doorbell request in any partition that corresponds to the bit.

Message Registers

Message registers enable 32-bit values to be passed between partitions with interrupt notification. Each NT endpoint supports four Inbound Message (INMSG[3:0]) registers and four Outbound Message (OUTMSG[3:0]) registers. The logical operation of message registers is illustrated in Figure 14.10.

- Associated with each outbound message register in a partition is a Switch Partition Message Control (SWPxMSGCTL[3:0]) register.
  - The register SWPxMSGCTLy corresponds to outbound message register y in switch partition x.
- When an outbound message register is written, the value written to the register is transferred to the inbound message register specified by the Register Select (REG) field of the SWPxMSGCTLy register in the partition specified by the Partition (PART) field of the SWPxMSGCTLy register. Thus, fields in the SWPxMSGCTLy register specify the routing of an outbound message register in one partition to an inbound message register in typically a different partition.

Notes

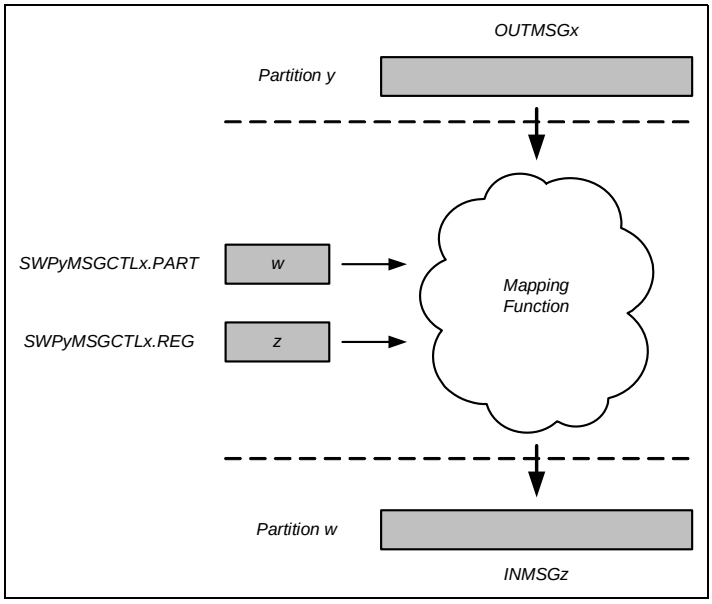

Figure 14.10 Logical Representation of Message Register Operation

Since the mapping of outbound message registers to inbound message registers need not be one-to-one, it is possible to map multiple outbound message registers, from typically different partitions, to a single inbound message register. In such a configuration, it is necessary to deal with possible contention to the inbound message register.

- When an outbound message register is written, the written value is transferred to the inbound message register specified by the corresponding SWPxMSGCTLy register. The transferred value may be accepted or rejected by the inbound message register.
- A transferred value is accepted if the message register is empty. When a transferred value is accepted, the Inbound Message (INMSG) field in the corresponding Inbound Message (INMSGx) register is updated with the transferred value, the Inbound Message Source Partition (SRC) field in the Inbound Message Source (INMSGSRC) register is updated with the partition number from which the message arrived, and the corresponding Inbound Message Status (INMSGSTSx) bit is set in the Message Status (MSGSTS) register. Once a transferred value is accepted, the inbound message register becomes full and remains full until the corresponding INTMSGSTSx bit is cleared.
- A transferred value is rejected if the message register is full. When a transfer value is rejected, the Outbound Message Status (OUTMSGSTSx) bit is set in the Message Status (MSGSTS) register that corresponds to the outbound message register that was written. This bit may be used to determine a transfer failed and needs to be retried.

Bits in the Message Status (MSGSTS) register may be used to generate NT interrupts. A bit in the MSGSTS register may be masked from generating an interrupt by setting the corresponding bit in the Message Status Mask (MSGSTSMSK) register.

Punch-Through Configuration Requests

The NT endpoint has the capability to generate PCI Express configuration transactions on the upstream link. This mechanism, referred to as *punch-through*, is provided to facilitate configuration of systems in which a root complex is not present in the PCI Express hierarchy associated with the NT endpoint. In essence, the NT endpoint may be crosslinked to another endpoint or to a switch device, and issue configuration requests to configure these devices.

Notes

Punch-through requests are always emitted on the NT function's link. In port operating modes with multiple functions (e.g., upstream switch port with NT function), it is not allowed for punch through requests issued by the NT function to hit the primary/secondary/subordinate window of the PCI-to-PCI bridge function or the bus/device/function ID associated with other functions in the port. Breaking this rule produces undefined results.

To generate a punch-through configuration transaction on the NT endpoint's link, the following sequence should be executed. Note that the registers that control punch-through requests are located in the configuration space of the NT function. These registers may be programmed via another port, i.e., using the global address space indirection registers (see Chapter 19, Register Organization) or via the SMBus interface.

1. Check if the punch-through configuration interface is busy by examining the Busy (BUSY) bit in the Punch-Through Configuration Status (PTCSTS) register (located in the configuration space of the NT function) and wait until the interface is not busy.
2. Configure the operation (e.g., read or write) in the Punch-Through Configuration Control registers (PTCCTL0 and PTCCTL1).
3. Write to the Punch-Through Configuration Data (PTCDATA) register to initiate the configuration read or write operation as selected by the OP field in the PTCCTL1 register.
   - This step causes the NT endpoint to emit a PCI Express configuration request TLP. The requester ID in the configuration request TLP is as follows.
     - The bus field is replaced by the captured bus number of the NT endpoint associated in the target partition.
     - The device and function fields are replaced by the value 0x4. This corresponds to device 0, function 4.
     - The tag field is set to 0x0.
   - In addition, the BUSY bit in the PTCSTS register is set to indicate a punch-through configuration transaction is in progress.
4. Wait for the operation to complete by polling the status of the Done (DONE) bit in the PTCSTS register.
   - The Done bit is set when the NT function receives a completion[1] whose destination ID matches the NT function's requester ID (see the requester ID description above).
5. Check the transaction completion status in the Status (STATUS) field of the PTCSTS register. If the initiated transaction was a read and it successfully completed, then the read result may be read from the PTCDATA register.
   - The STATUS field in the PTCSTS register reflects the status of the received completion (e.g., successful completion, unsupported request, completer abort, etc.).

It is possible for a completion to not be received in response to a punch-through configuration transaction. A punch-through operation may be aborted by writing a one to the DONE bit in the PTCSTS register. This will cause subsequent completions to be discarded until a new punch-through configuration transaction is generated. This mechanism should only be used when it is certain that a completion is lost and will never arrive. It is up to the user to make this determination.

Re-programming the Bus Number of the NT Function

In some systems, it may be desirable to use a PCI Express switch to interconnect several intelligent devices without the presence of a PCI Express root (i.e., the switch can be configured via SMBus or EEPROM). One of the challenges in building this type of system is the assignment of PCI Express requester IDs (i.e., bus, device, function) to each of the intelligent devices. Such assignment is a pre-requisite in order for ID-routed TLPs (i.e., completions) to be correctly routed by the PCI Express switch.

[1] The NT function assumes that the received completion is a completion with data (CplD) TLP and does not check for any violations in the format of the TLP.

Notes

Normally, devices with a PCI Express port capture the bus number associated with the port on reception of type 0 configuration write requests that target the port. In system scenarios where there is no root complex in the PCI Express hierarchy, the devices will not receive type 0 configuration write requests. As a result, the default bus number (i.e., bus number 0) will be used by the devices, and ID-routing across the hierarchy won't be possible.

The switch contains a feature that allows software to explicitly configure the bus number associated with a switch port that has an NT function. The programming is done by writing to the Bus (BUS) field in the TLCNTCFG register located in the port's configuration space. Programming of the port's bus number is only allowed when the port operates in NT function mode or NT with DMA function mode.

Figure 14.11 shows a system scenario where a PCI Express switch is used to connect several intelligent devices. This system does not have a root complex, and communication among the intelligent devices is desired. Each intelligent device uses a PES32NT8xG2 NT port to connect to the PCI Express switch. Prior to initiating communication, the CPU located in the intelligent device programs the PES32NT8xG2 NT port that faces the rootless PCI Express hierarchy with an appropriate requester ID (i.e., by writing to the BUS field in the NT port's TLCNTCFG register; this register can be accessed by the processor on the intelligent device using switch's global address space access registers).

Once the requester IDs for each intelligent device are programmed with unique values, traffic across the rootless PCI Express switch routes normally.

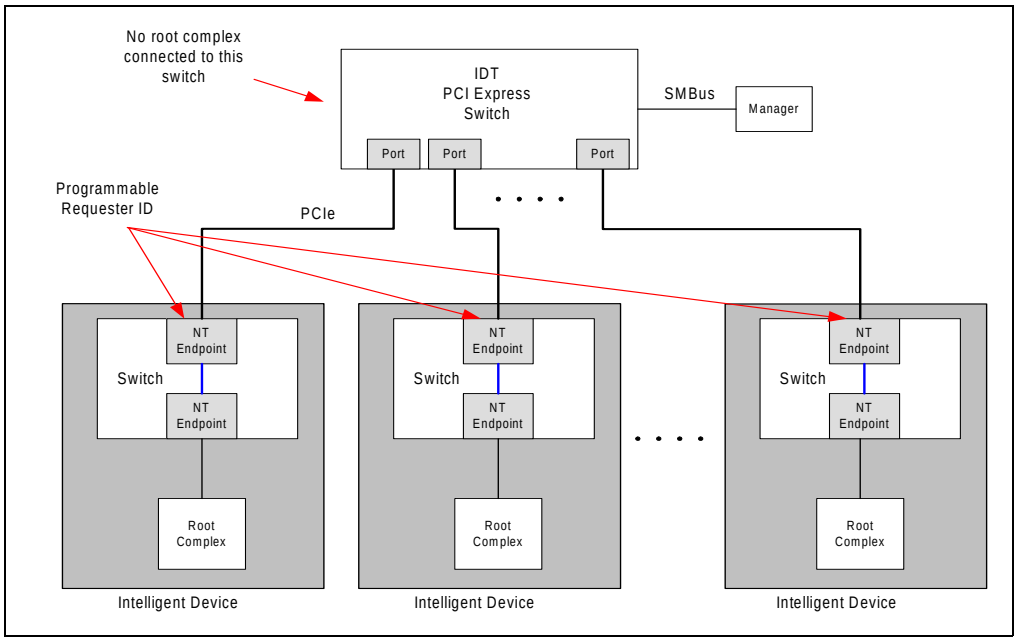

Figure 14.11 Example of a Rootless PCI Express Hierarchy with Bus Number Reprogramming

Interrupts

The NT endpoint has six the following sources of interrupts.

- Message status
- Doorbell status
- Switch events
- Failover change initiated by failover capability associated with partition
- Failover change completed by failover capability associated with partition
- A temperature sensor alarm (see Chapter 18, Temperature Sensor).

Notes

The interrupt sources each have a corresponding status bin in the NT Endpoint Interrupt Status (NTINTSTS) register.

- When an interrupt source requests service, the corresponding bit in the NTINTSTS register is set.
- An interrupt source may be masked from generating an interrupt by setting the corresponding mask bit in the NT Endpoint Interrupt Mask (NTINTMSK) register. By default, all interrupt sources are masked.
- Once a bit corresponding to an interrupt source is set in the NTINTSTS register, interrupts associated with that source are inhibited until the bit is cleared in the NTINTSTS register.

When an unmasked interrupt condition occurs, an MSI or interrupt message is generated by the NT endpoint as described in Table 14.1. The removal of the interrupt condition occurs when unmasked status bits causing the interrupt are masked or cleared.

- When an NT endpoint is configured to generate INTx messages, the INTx used (i.e., INTA, INTB, etc.) depends on the programming of the Interrupt Pin (INTRPIN) register.

An MSI generated by the NT endpoint is always routed to the partition's upstream port link.

- An MSI generated by the NT endpoint must not must not target memory ranges associated with the upstream port's PCI-to-PCI bridge function (e.g., memory base/limit registers) and/or DMA function (e.g., BAR 0 aperture).
- An MSI generated by the NT endpoint never multicasted. Software must never configure the address of an MSI generated by the NT function to fall within an enabled multicast BAR aperture in the partition. Violating this requirement produces undefined results.

Unmasked Interrupt	EN Bit in MSICAP Register	INTXD Bit in PCICMD Register	Action
Asserted	1	X	MSI message generated
	0	0	Assert_INTx message request generated
	0	1	None
Negated	1	X	None
	0	0	Deassert_INTx message request generated
	0	1	None

Table 14.1 NT Endpoint Interrupts

Virtual Channel Support

Virtual channel support for ports in the PES32NT8xG2 is described in section Virtual Channel Support on page 14-21. The NT function contains a VC Capability Structure that provides architected port arbitration and TC/VC mapping for VC0.

For port operating modes in which the NT function is function 0 of the port, the VC Capability Structure in this function provides architected port arbitration and TC/VC mapping for all functions of the port. For port operating modes in which the NT function is not function 0 of the port, the registers in the NT function's VC Capability Structure are 'reserved'[1] and must not be programmed. In these modes, the VC Capability Structure in function 0 of the port provides architected port arbitration and TC/VC mapping for all functions of the port, including the NT function.

1. Reading from a reserved address returns and undefined value. Writes to a reserved address complete successfully but produce undefined behavior on the register.

Notes

Maximum Payload Size

The PES32NT8xG2 requires that the Maximum Payload Size (MPS) field in the PCI Express Device Control (PCIEDCTL) register be set identically in all functions (i.e., PCI-to-PCI bridge, NT, and DMA) of a partition. In addition, when inter-partition transfers are possible between two or more partitions (i.e., across the NT interconnect), all switch functions in these partitions must have the same MPS setting. Violating this rule produces undefined results.

Note that a port with a maximum link width of x1 supports a Maximum Payload Size (MPS) of up to 1 KB. Ports with a maximum link width of x2, x4, or x8 support an MPS of up to 2 KB. The MPAYLOAD field in the PCI Express Device Capabilities (PCIEDCAP) register is automatically set by the hardware based on the port's maximum link width to reflect this.

Power Management

Refer to Chapter 9, Power Management.

Bus Locking

The NT function does not support bus locking. Memory read request-locked TLPs received by an NT function are treated as unsupported requests and an unsupported request completion with no data (CplLk) is returned. The operation of a switch partition is undefined when bus locking is performed in a partition that contains an NT function in its upstream port.

ECRC Support

End-to-End CRC (ECRC) is supported for transactions that are forwarded through the NT interconnect. Since the TLP contents (i.e., header) are modified for TLPs flowing between NT endpoints, a new ECRC must be computed. When a TLP is forwarded on to the NT interconnect by an NT endpoint, the NT endpoint computes the ECRC for the new translated TLP in parallel with checking the ECRC, if it exists, of the received TLP. The existence of an ECRC in the received TLP is indicated by the TD bit in the TLP header.

The NT function only checks and logs ECRC errors when the ECRC Check Enable (ECRCCE) bit is set in the function's AER Control (AERCTL) register, and the TLP with ECRC is received from the upstream port's link.

- ECRC error checking and logging is not performed by the NT function when it does not receive the TLP from the link.
  - In this case, the ECRC error checking and logging is done by the port that received the TLP from the link (e.g., downstream port).
- If the port is operating in a multi-function mode, then ECRC errors are only logged in functions in which ECRC checking is enabled.

If ECRC checking applicable as described above and an ECRC error is detected, then an ECRC error is reported by the NT endpoint that received the TLP. See section Error Detection and Handling by the NT Function on page 14-25 for details.

- If ECRC checking is enabled in an NT endpoint, then ECRC is checked in all TLPs received by the NT endpoint that contain an ECRC. The reception of a TLP without ECRC is not considered an error (i.e., the TLP is processed normally).

ECRC generation is enabled in the NT endpoint when the ECRC Generation Enable (ECRCGE) bit is set in the function's AER Control (AERCTL) register.

If ECRC generation is enabled in the NT endpoint associated with the destination partition of the translated TLP, then the translated TLP contains an ECRC and the TD bit in the translated TLP header is set.

- If ECRC checking is not enabled in the NT endpoint that received the TLP, or if the received TLP does not contain an ECRC, or if ECRC checking is enabled and the ECRC computed by the NT endpoint is correct, then the ECRC associated with the translated TLP is that computed by the NT endpoint associated with the destination partition.

Notes

- When ECRC checking is not enabled in the NT endpoint, there is a possibility of silent data corruption on packets that cross the NTB (i.e., when a TLP with ECRC error is received by the NT endpoint, the NT endpoint does not check ECRC, and a new ECRC is re-computed by the NT endpoint in the destination partition, thereby "hiding" the existing error in the packet). To prevent silent data corruption, it is strongly recommended that ECRC checking be enabled at the NT endpoints.

– If ECRC checking is enabled in the NT endpoint that received the TLP (i.e., the ECRCCE bit is set in the AERCTL register) and the TLP contains an ECRC error, then the ECRC associated with the translated TLP is generated by recomputing the ECRC of the translated TLP and inverting all bits. This ensures that the translated TLP carries a corrupt ECRC, so that the ECRC error may be detected at the TLP's final destination.

Also, if ECRC generation is enabled in an NT endpoint, then all TLPs originated by that endpoint contain an ECRC. If ECRC generation is disabled in an NT endpoint, all TLPs emitted by the function do not have ECRC.

Access Control Services (ACS)

The NT function supports the following ACS checks[1]:

– ACS Peer-to-Peer[2] Request Redirect

– ACS Peer-to-Peer Completion Redirect

– ACS Direct Translated Peer-to-Peer

ACS is programmed via the ACS Capability Structure in the NT function's configuration space.

– The NT function supports ACS checks when the port operates in the following port operating modes:

- Upstream switch port with NT function
- Upstream switch port with NT and DMA function

– In these modes, the ACS Capability Structure is linked into the NT function's configuration space (refer to section NT Function Capability Structures on page 19-22).

– The NT function does not support ACS checks when the port operates in any other mode.

The NT function applies the above ACS checks for TLPs it emits (i.e., TLPs received on another partition that have undergone NT address translation).

– ACS checks are not applied to completion TLPs generated by the NT function in response to a received requests that target BAR 0 of the NT function, when this BAR is configured to map into the NT function's configuration space.

– ACS checks are not applied to punch-through configuration requests issued by the NT function.

Table 14.2 lists ACS checking and handling performed by the NT function. Note that none of the ACS checks result in an ACS violation error.

1. The PES32NT8xG2 does not support ACS Peer-to-Peer Egress Control among the functions of a multi-function upstream port.

2. In a multi-function upstream port, 'peer-to-peer' implies traffic exchanged among the port functions (e.g., from the port's NT function to the port's PCI-to-PCI bridge function).

Notes

ACS Check	PCI Express Base Specification[1] Section	Error Reporting Condition	Action Taken
ACS Peer-to-Peer (P2P) Request Redirect	6.12.1.1	N/A (not an ACS violation)	Offending request is redirected upstream towards root complex.
ACS P2P Completion Redirect			Offending completion is redirected upstream towards root complex.
ACS Direct Translated P2P			Offending TLP is subject to ACS P2P Request Redirect rules.

Table 14.2 ACS Checks Performed by the NT Function in a Port Operating in Multi-function Mode

1. Refer to PCI Express Base Specification Revision 2.1.

When an ACS check causes a TLP to be re-directed, the re-direction is implemented such that TLPs emitted by the NT function that are ACS re-directed follow the ordering rules described in section Packet Ordering on page 4-6. Note that ACS Direct Translated Peer-to-Peer requires that the NT function perform a check on the Address Type (AT) field in request TLPs it emits. Prior to performing this ACS check, the AT field in the emitted TLPs is subject to the processing described in section Address Type Processing on page 14-15.

– If the NT function clears the AT field in a TLP it emits (i.e., the TLP is marked as untranslated), the ACS Direct Translated P2P check is reduced to an ACS P2P Request Redirect check.

ACS checks are only applicable to certain TLP types. Table 14.3 list the ACS checks supported by the NT function and the TLP types on which they are applied.

ACS Check	Applicable to the following TLP type(s)
ACS Peer-to-Peer (P2P) Request Re-direct	Peer-to-Peer Request TLPs
ACS P2P Completion Re-direct	Peer-to-Peer Completion TLPs
ACS Direct Translated P2P	Peer-to-Peer Memory Request TLPs

Table 14.3 TLP Types Affected by ACS Checks

As an example of an ACS check performed by the NT function, consider the case where software enables ACS Peer-to-Peer Request Redirect in the NT function. This commands the NT function to re-direct upstream (i.e., transmit on the upstream link) all requests that it issues which would have otherwise been logically routed via the upstream port's PCI-to-PCI bridge function. Figure 14.12 shows an example of a ACS Peer-to-Peer Request Redirect. The green lines mark the requests intended route, and the orange lines the request's re-directed route due to ACS.

Notes

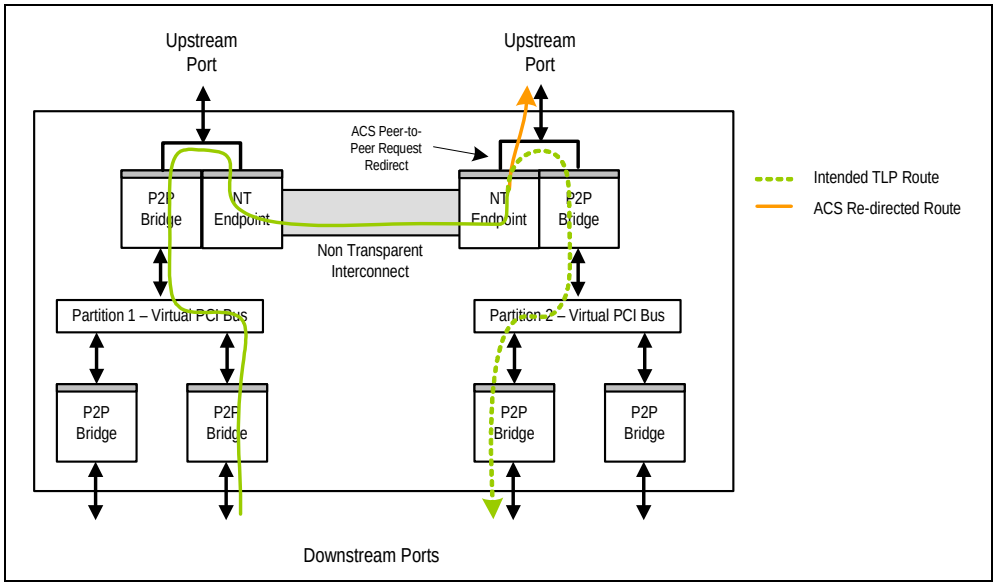

Figure 14.12 Example of ACS Peer-to-Peer Request Re-direct Applied by the NT Function

When multiple ACS checks are enabled, they are prioritized as described in Chapter 10, Tables 10.4 and 10.5. Refer to the PCI Express Base Specification for further information on ACS.

Error Detection and Handling by the NT Function

This section describes error conditions associated with non-transparent switch operation. This includes physical, data-link, and transaction layer errors detected by the switch ports, as well as application layer errors associated with the non-transparent-bridge (NTB) functionality.

– Internal switch errors (i.e., parity errors, switch time-out, and internal memory errors) are associated with the switch core and not with a specific port function. These errors are not described here. Refer to section Internal Errors on page 4-16 for a detailed description of these errors.

The errors described here apply to ports that operate in a mode that includes the NT function (e.g., NT endpoint mode, upstream switch port with NT endpoint mode, etc.) This section focuses specifically on errors related to the NT function[1]. Errors that affect all functions of the port (i.e., non function-specific errors) are noted where appropriate.

Error detection and handling in the PES32NT8xG2 follows the requirements in the PCI Express Base Specification.

The error checking and handling described here is performed by each PES32NT8xG2 NT function. In cases where the error condition propagates among multiple NT functions (e.g., a poisoned TLP received by an NT function, passed across the NT interconnect, and emitted by the NT function in another partition), each NT function performs error checking and handling independently. Refer to section NTB Inter-Partition Error Propagation on page 14-31 for further details and examples on this.

The errors described below are associated with specific actions to log and report the error. The terms ‘uncorrectable error processing’ and ‘correctable error processing’ refer to the processing described in Section 6.2.5 of PCI Express Base Specification.

Errors which are not function-specific are logged in the corresponding status and logging registers of all functions in the port. Errors that are function-specific are logged in the status and logging register of the affected function only. Signaling of non function-specific errors follows the rules in Section 6.2.4 of PCI Express Base Specification.

1. Errors associated with the PCI-to-PCI bridge function (i.e., transparent operation errors) are described in Chapter 10. Errors associated with the DMA function are described in Chapter 15.

Notes

Some of the errors described below are marked as function-specific when the “function claims the TLP”. A function claims a TLP in the following cases:

NT Endpoint function:

- Address Routed TLPs: The TLP address falls within the address space range(s) programmed in the function’s base address registers (BARs).
- ID Routed TLPs: The TLPs requester ID matches any of the following:
  - The NT function’s bus/device/function assignment within the PCI Express hierarchy.
  - The requester ID used by the NT function for punch-through requests (i.e., refer to section Punch-Through Configuration Requests on page 14-18)
  - The upper two bits are 0b10 and the lower six bits point to a valid entry in the NT function’s Mapping table.
- Implicit Route TLPs: Always.

PCI-to-PCI Bridge function

- Refer to section Error Detection and Handling by the PCI-to-PCI Bridge Function on page 10-11.

DMA function:

- Refer to section PCI Express Error Handling by the DMA Function on page 15-28.

For a port in Upstream Switch Port with NT Endpoint mode, the TLP is claimed when either one of its two functions claims the TLP.

Physical Layer Errors

All physical layer errors are non-function specific. These errors are described in the error handling section for the PCI-to-PCI bridge function. Refer to section Physical Layer Errors on page 10-11.

Data Link Layer Errors

All data link layer errors are non-function specific. These errors are described in the error handling section for the PCI-to-PCI bridge function. Refer to section Data Link Layer Errors on page 10-12.

Transaction Layer Errors

Table 14.4 lists non-ACS error checks performed by the transaction layer and the action taken when an error is detected. ACS error checks and handling are discussed in section Transaction Layer Error Pollution on page 14-31.

For some errors, it is necessary to determine if the function that receives a TLP from the link is an “ultimate receiver” or “intermediate receiver”. The NT function is always considered an ultimate receiver when the TLP is received directly from the link. This applies to all TLPs, including NT multicast TLPs.

Per PCI Express Base Specification 2.1, transaction layer errors are ignored in cases where the error is associated with a received packet for which the physical or data-link layers report an error. This prevents error pollution across the stack layers. Within the transaction layer, there are error pollution rules that resolve the cases where two or more errors are detected simultaneously. Refer to section Transaction Layer Error Pollution on page 14-31 for details on transaction layer error pollution.

Notes

Error Condition	PCI Express Base Specification[1] Section	Function-Specific Error	Role Based (Advisory) Error Reporting Condition	Action Taken
Poisoned TLP received	2.7.2.2, 6.2.3.2.4.3	Yes	Advisory when the corresponding error is configured as non-fatal in the AERUESV register	Detected Parity Error bit (PCISTS.DPE) is set. If the poisoned TLP is a completion, the Master Data Parity Error Detected bit (PCISTS.MDPED) is set (if PCICMD.PERRE is set). If the TLP is received from the NT port's link (i.e., the NT function is the ultimate receiver): Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. Affected packet is forwarded across the NTB to the destination partition's NT Endpoint. The following are the actions taken by the NT function in destination partition: Master Data Parity Error Detected (PCISTS.MDPED) is set if the poisoned TLP is a write request.
ECRC check failure[2]	2.7.1	No	N/A (always non-advisory)	Affected packet's ECRC is artificially corrupted as described in section ECRC Support on page 14-22. The TLP is forwarded across the NT bridge (unless the TLP does not hit an NT mapping window, in which case it is dropped). If the TLP is received from the NT port's link (i.e., the NT function is the ultimate receiver): Uncorrectable error processing.
Unsupported request	See Table 14.5	Yes if a function in the port claims the TLP. Else No.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register and the request is non-posted	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. For Non-Posted unsupported requests, the function that claims the TLP generates a completion with UR status. If the request is not claimed, then function 0 generates the completion with UR status.
Completion timeout	2.8	N/A	N/A (always non-advisory)	Not applicable. The NT function does not check for completion timeout as it does not track requests issued across the NT bridge.

Table 14.4 Transaction Layer Errors Associated with the NT Function (Part 1 of 2)

Notes

Error Condition	PCI Express Base Specification[1] Section	Function-Specific Error	Role Based (Advisory) Error Reporting Condition	Action Taken
Completer abort	2.3.1	N/A	N/A (always non-advisory)	Not applicable. The NT function does not issue completions with 'Completer Abort' status except for ACS violations. For this exception case, the error is considered an ACS error and is not logged as a completer abort error.
Unexpected completion received	See Table 14.6	Yes if a function in the port claims the TLP. Else No.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. The unexpected completion is dropped.
Completion with UR status received[3]	6.2.3.2.5	N/A	N/A (always non-advisory)	Reception of a completions with UR status is handled as any other received completion. In addition, the Received Master Abort Status (RMAS) bit in the PCISTS register is set.
Completion with CA status received[4]	6.2.3.2.5	N/A	N/A (always non-advisory)	Reception of a completions with CA status is handled as any other received completion. In addition, the Received Target Abort Status (RTAS) bit in the PCISTS register is set.
Receiver overflow	2.6.1.2	No	N/A (always non-advisory)	Uncorrectable error processing. TLP is nullified.
Flow control protocol error	2.6.1	No	N/A (always non-advisory)	Not applicable. The PES32NT8xG2 does not check for any flow control protocol errors.
Malformed TLP	See section TLP Malformation Checks on page 14-30	No	N/A (always non-advisory)	Uncorrectable error processing. TLP is nullified.
Internal Error	6.2	Yes	N/A (always non-advisory)	Refer to section Internal Errors on page 4-16.

Table 14.4 Transaction Layer Errors Associated with the NT Function (Part 2 of 2)

1. Refer to PCI Express Base Specification Revision 2.1.
2. Refer to section ECRC Support on page 14-22.
3. If the completion is unexpected, then it is handled as an unexpected completion received error.
4. If the completion is unexpected, then it is handled as an unexpected completion received error.

Unsupported Requests

Table 14.5 lists the conditions for which the NT function handles requests as unsupported requests (UR).

Notes

Conditions Handled as UR	Description	PCI Express Base Specification Sect.
Effective BAR Aperture check	Refer to section BAR Limit on page 14-2.	n/a
Lookup Table Address Translation error: Table entry invalid, partition entry invalid, or out-of-bounds lookup table entry selected.	Refer to section Lookup Table Address Translation on page 14-5.	n/a
Destination Partition errors: Destination partition not active Destination partition does not have an NT endpoint Destination partition is same as source partition BME bit cleared in NT function of destination partition when translating a request TLP Link is down in destination partition's upstream port and TLP is destined to this link[1].	Refer to section Lookup Table Address Translation on page 14-5, and section Request ID Translation on page 14-11	n/a
Requester ID miss in NT Mapping Table	Refer to section Request ID Translation on page 14-11	n/a
NT function in D3Hot state	Refer to section Overview on page 9-1.	5.3.1.4.1
NT function in destination partition in D3Hot state	Refer to section Overview on page 9-1.	n/a
Type 1 Configuration Requests	Reception of a Type 1 configuration request at the NT function.	7.3.3
Vendor Defined Type 0 message reception[2]	Vendor Defined Type 0 message which targets the NT function.	2.2.8.6
Messages with invalid message code	Reception of a message TLP with invalid message code that targets the NT function.	2.3.1
Poisoned configuration write request, poisoned memory write request, or poisoned message with data targeting the NT function	Reception of a poisoned IO request, memory write request, or message with data (except Vendor Defined messages) that targets a switch port's NT function.	2.7.2.2
Reception of MRdLk Request	Reception of an MRdLk request by the NT function	6.5.7

Table 14.5 Conditions Handled as Unsupported Requests (UR) by the NT Function

1. This excludes inter-function transfers (e.g., NT function to PCI-to-PCI bridge function) in the destination partition, as these transfers are not destined to the upstream link. Refer to section Link States on page 7-9 for a list of link states in which the link is considered down (e.g., L2/L3 Ready).

2. NOTE: Vendor Defined Type 1 messages which target the NT function are silently discarded.

Unexpected Completions

Table 14.6 lists the conditions for which the NT function handles completions as unexpected completions.

Notes

Conditions Handled as UC	Description	PCI Express Base Specification Section
Non function-specific unexpected completion	Port receives a completion TLP that is not claimed by any function of the port. This is a non function-specific error and is therefore logged in all functions of the port.	6.2.4
NT function unexpected completion (case 1)	NT function receives a completion TLP whose requester ID matches the NT function's bus and device number is zero, and the function number matches the NT function's number (i.e., zero or one depending on the port operating mode). This is a function-specific error associated with the NT function. Note that this does not refer to a completion TLP whose requester ID matches a valid entry in the NT function's mapping table. Such TLPs are handled as described below.	2.3.2
NT function unexpected completion (case 2)	Port receives a completion TLP whose requester ID has the two upper bits set to 0b10 and the lower six bits point to a valid entry in the NT function's mapping table, but the following problems are encountered: The destination partition is not active, the destination partition does not have an NT endpoint, or the destination partition is the same as source partition.	N/A

Table 14.6 Conditions Handled as Unexpected Completion (UC) by the NT Function

TLP Malformation Checks

TLP malformation checks are done by the port and are non function-specific. Refer to section TLP Malformation Checks on page 10-16 for details of malformation checks performed by a port.

TLP Header Logging

TLP header logging is subject to the rules outlined in section 6.2 of the PCI Express Base Specification Revision 2.1.

Note: The switch does not support the recording of multiple headers, nor does it support the recording of headers for uncorrectable internal errors. When an uncorrectable internal error is reported by AER, a header of all ones is recorded.

The following non function-specific errors require that the offending TLP's header be logged in the AER capability structure of all function's in the port.

- Reception of a TLP with ECRC error on the port's link.
- Reception of a request that is unsupported on the port's link, when no function in the port claims the TLP.
- Reception of an unexpected completion on the port's link, when no function in the port claims the TLP.
- Reception of a malformed TLP on the port's link.

Notes

The following function-specific errors require that the offending TLP's header be logged in the NT function's AER capability structure.

- Reception of a request that is unsupported and is claimed by the NT function.
- Reception of an unexpected completion that is claimed by the NT function.
- Reception of a poisoned TLP on the upstream port's link that is claimed by the NT function.
  - When the TLP is not received on the link, header logging is not performed.

Transaction Layer Error Pollution

The error pollution rules for non-transparent operation are the same as for transparent operation. Refer to section Transaction Layer Error Pollution on page 10-20 for further details.

NTB Inter-Partition Error Propagation

This section describes the PES32NT8xG2's handling of error conditions that propagate across the device during inter-partition transfers. This section builds upon the error handling concepts described above for the NT function, as well as the error handling for transparent operation described in Chapter 10.

Figure 14.13 shows a basic configuration of the PES32NT8xG2 with two partitions connected by an non-transparent bridge (NTB). Note that this is only a sample configuration chosen to illustrate the concepts described in this section. These concepts are applicable to all other possible multi-partition configurations allowed by the PES32NT8xG2.

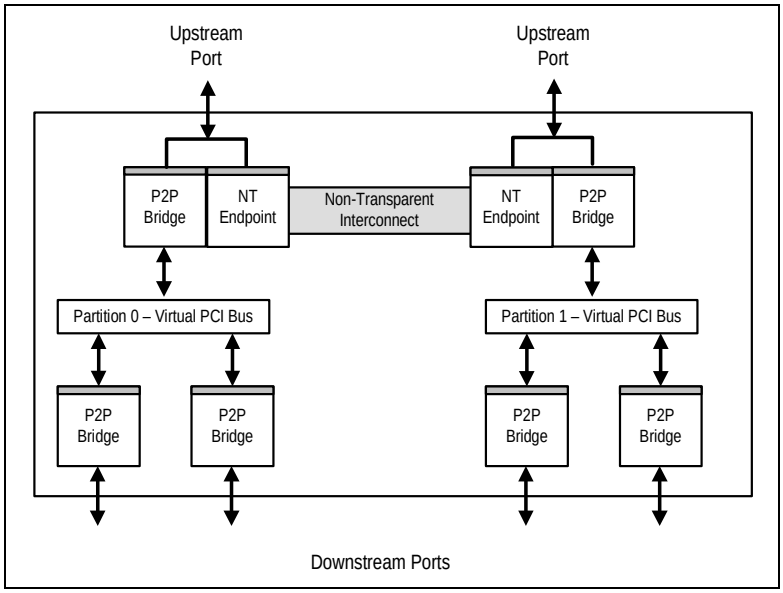

Figure 14.13 Basic Non-Transparent PES32NT8xG2 Configuration

In the sample configuration shown in the figure, each partition is composed of three PCI-to-PCI (P2P) bridge functions (i.e., a 3-port transparent switch). The NTB is associated with an NT function in each partition, and the upstream port of each partition is configured in Upstream Switch Port with NT Endpoint mode.

Note that both functions of the upstream port (i.e., PCI-to-PCI bridge function and NT function) connect to the PCI Express link, as well as to each other. As described in section Non-Transparent Operation on page 1-7, these functions may route TLPs to each other (i.e., an inter-function transfer).[1]

Each of the logical functions shown in the figure has independent error detection and handling capabilities. The PCI-to-PCI bridge functions perform error handling as described in Chapter 10, and the NT functions performs error handling as described in the sections above.

1. Note that inter-function transfers are not transmitted on the upstream port's link.

Notes

When a TLP is routed across the PES32NT8xG2 (within a partition or across partitions via the NTB), each function that receives the TLP checks for errors. Thus, it is possible that more than one function detect and report an error associated with the TLP.

This section focuses on errors related to TLP routes that cross the NTB (i.e., inter-partition routes). Routes that do not cross the NTB (i.e., intra-partition routes) are not discussed here as these are well understood from the error handling description in the PCI Express Base Specification.

The following sections describe error propagation for the different types of errors detected by the PES32NT8xG2 ports. The description is based on the logical paths that TLPs can follow across the functions shown in Figure 14.13.

Physical Layer Errors

Physical layer errors (refer to section Physical Layer Errors on page 10-11) are only detected by the port that receives the packet from the link. No other ports or functions in the logical path of the TLP check for physical layer errors.

Data Link Layer Errors

Data link layer errors (refer to section Data Link Layer Errors on page 10-12) are detected by the ingress port that receives the packet from the link, or by the egress port (if any) that transmits the packet. No other ports or functions in the logical path of the TLP check for data link layer errors.

Transaction Layer Errors

Transaction Layer errors associated with the reception of a TLP are checked by each PES32NT8xG2 function that receives the TLP. In some cases, the first function in the TLP's logical path that detects an error will nullify or drop the TLP. As a result, no other functions in the TLP's logical path will detect the error condition.

In other cases, the first function in the TLP's logical path that detects an error will log the error appropriately and allow the TLP to continue in its path. As a result, other functions in the TLP's logical path may check, detect, and log errors associated with the TLP. In this case, it is possible that multiple functions detect and report errors associated with the TLP.

For inter-partition transfers, functions in both partitions may detect and report errors associated with the TLP. A function that detects an error and is configured to signal the error by issuing an error message (i.e., Fatal, Non-Fatal, or Correctable message), does so by issuing the message towards the upstream port's link associated with the function's partition.

Additionally, for certain types of errors (e.g., unsupported request on a non-posted request), it is possible that the function that detects the error generate a completion TLP destined to the original sender of the TLP (i.e., the requester). In the case where the request TLP was received by the PES32NT8xG2 on one partition, crossed the NTB, and the error was found in a second partition, the completion TLP generated by the function that detects the error in the second partition logically crosses the NTB and is sent towards the requester in the first partition.

The following sections describe error propagation for each type of transaction layer error.

Receiver Overflow Error

Receiver overflow errors are only detected by the port that receives the TLP from the link. The TLP is nullified and no other ports or functions in the logical path of the TLP will detect this type of error.

ECRC Error

ECRC errors are only detected and logged in AER by the function(s) in the port that received the TLP from the link. No other functions in the logical path of the TLP log the ECRC error. An NT function that receives a TLP with ECRC error handles it as described in section ECRC Support on page 14-22.

Notes

Malformed TLP Error

In the PES32NT8xG2, malformed TLP errors are checked at the ingress port that receives the packet from the link, or at the egress port (if any) that transmits the packet. Malformed TLPs are nullified by the function that detects the error and thus no other functions in the logical path of the TLP will detect this type of error.

Note that since TLP malformation checks are only performed by the ingress port that receives the packet from the link and by the egress port that transmits the packet to the link, intermediate functions in the TLPs logical path do not detect formation errors. To prevent device malfunction, the user must guarantee that:

- The TC/VC mappings of all functions in the PES32NT8xG2 are configured such that incoming TLPs are always mapped to VC 0.
- The Maximum Payload Size (MPS) field in the PCIEDCTL register is set identically in all functions that a TLP may logically traverse within the switch.

Unsupported Request (UR) Error

TLPs received by an upstream port that are not claimed by any function in the upstream port are treated as unsupported requests and the error is logged in all functions of the port. Upon claiming a received request TLP (i.e., posted or non-posted), each PES32NT8xG2 function checks for UR errors. If the function determines the request is unsupported, the TLP is consumed by this function and is handled as a UR error. Else, the TLP is consumed and processed normally by the function, or forwarded across the function (e.g., PCI-to-PCI bridge), depending on its final destination. For non-posted requests, a function that treats the request as a UR must generate a completion TLP and send it to the original requester.

In the PES32NT8xG2, it is possible that a non-posted TLP that is routed across partitions via the NTB (i.e., from a source partition to a destination partition) be UR-ed by a PES32NT8xG2 function in the destination partition. In this case, the completion TLP generated by the function that detects the error is logically routed back within the PES32NT8xG2 towards the initiator of the request. Thus, the completion TLP generated by the function in the destination partition logically crosses the NTB and is routed towards the request initiator in the source partition.

Figure 14.14 shows an example of a non-posted request TLP received by an PES32NT8xG2 port on a first partition that is transferred across the NTB to a second partition. The request is URed by the PCI-to-PCI bridge function in the PES32NT8xG2 upstream port of the second partition.

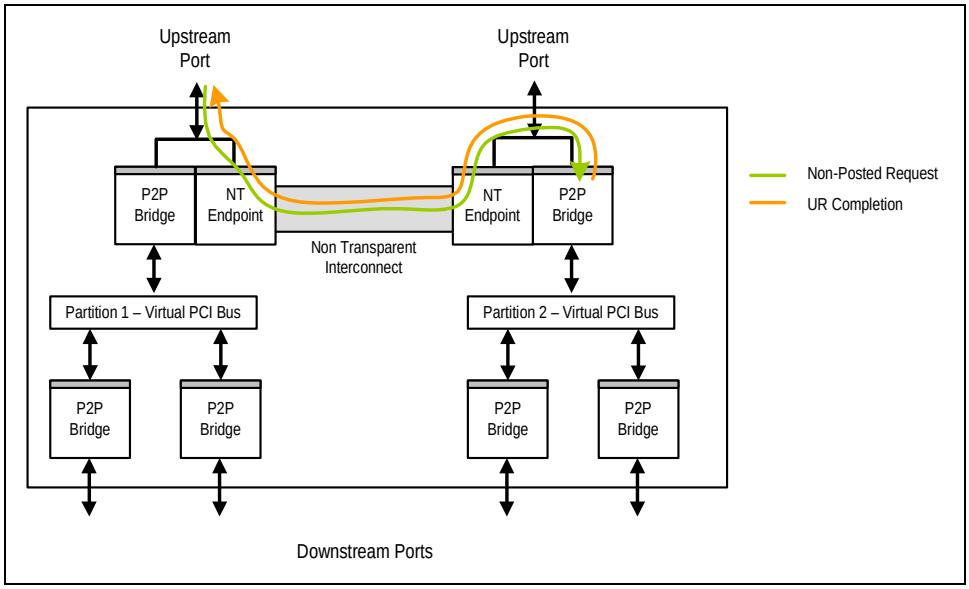

Figure 14.14 Unsupported Request Example # 1

Notes

When the PCI-to-PCI bridge function in Partition 2 detects the UR error, it logs it as such and generates a completion TLP destined towards the NT Endpoint function associated with Partition 2. The completion TLP is then transferred across the NTB and transmitted by the NT Endpoint in Partition 1 towards the initiator of the request.[1]

Note that each function that receives the TLP (i.e., NT function in partition 1 and PCI-to-PCI bridge function in partition 2) checks for UR errors. Also, notice that the request TLP logically stops at the PCI-to-PCI bridge function in partition 2 since this function detects the UR error. For this example, error logging would occur as shown in Table 14.7.

Function	Error Logging
Upstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Unsupported Request' error in Table 10.9
NT Endpoint (Partition 2)	Refer to row corresponding to 'Completion with UR status received' in Table 14.4
NT Endpoint (Partition 1)	No error logged.

Table 14.7 Error Logging at Each Function for UR Example # 1

Figure 14.15 shows another example of an unsupported request condition for an inter-partition transfer. In this case, a non-posted request TLP received by the downstream switch port in a first partition is logically routed towards the NT endpoint in the same partition. The request TLP is then transferred across the NTB to a second partition. The request is then URed by the PCI-to-PCI bridge function in the PES32NT8xG2 downstream switch port in the second partition.

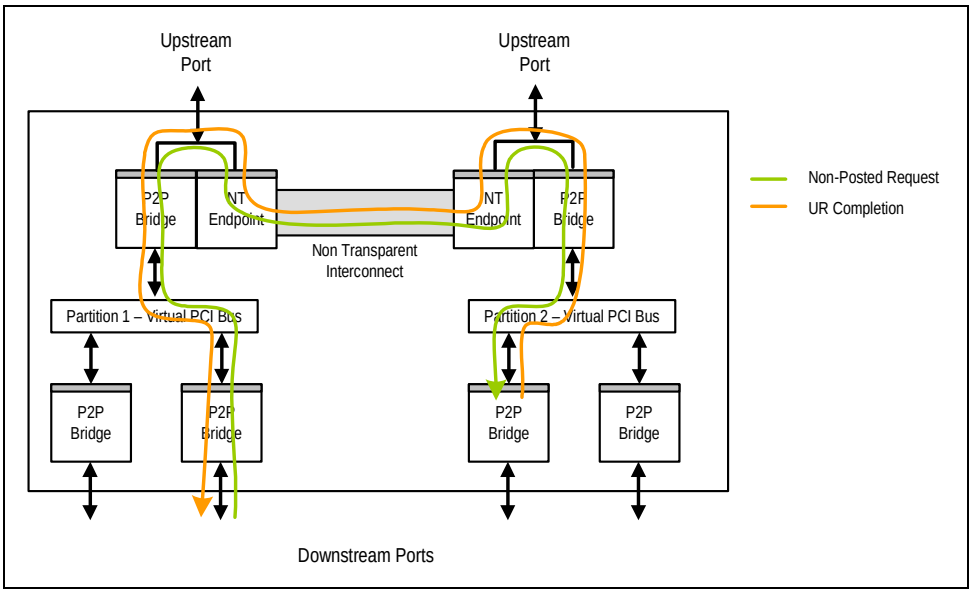

Figure 14.15 Unsupported Request Example # 2

When the downstream switch port's PCI-to-PCI bridge function in Partition 2 detects the UR error, it generates a completion TLP destined towards the NT Endpoint function associated with Partition 2. The completion TLP is then transferred across the NTB and sent by the NT Endpoint in Partition 1 towards the initiator of the request. Again, note that each function that receives the TLP (i.e., NT function in partition 1,

1. Note that the completion is guaranteed to cross the NTB since it corresponds to a request that had previously crossed the NTB. That is, the completion's destination ID will hit a valid entry in the NT Mapping table.

Notes

PCI-to-PCI bridge function of the upstream port in partition 2, and PCI-to-PCI bridge function of the downstream switch port in partition 2) checks for UR errors. Also, notice that the request TLP logically stops at the PCI-to-PCI bridge function of the downstream switch port in partition 2 since this function detects the UR error.

For this example, error logging occurs as shown in Table 14.8.

Function	Error Logging
Downstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Unsupported Request' error in Table 10.9
Upstream PCI-to-PCI Bridge (Partition 2)	No error logged.
NT Endpoint (Partition 2)	Refer to row corresponding to 'Completion with UR status received' in Table 14.4
NT Endpoint (Partition 1)	No error logged.
Upstream PCI-to-PCI Bridge (Partition 1)	No error logged.
Downstream PCI-to-PCI Bridge (Partition 1)	No error logged.

Table 14.8 Error Logging at Each Function for UR Example # 2

Unexpected Completion Received Error

Unexpected completion TLPs are dropped by the function that detects the error. Therefore, no other functions in the logical path of the TLP will detect this type of error.

Poisoned TLP Received Error

Poisoned TLP errors may propagate across multiple PES32NT8xG2 ports. The following rules apply:

- All functions in the logical path of the poisoned TLP log the error in the PCI legacy error registers.
  - The PCI-to-PCI bridge function logs the error in the PCI Status (PCISTS) or Secondary Status (SECSTS) registers as appropriate.
  - The NT and DMA functions log the error in the PCISTS register.
- The function that claims the TLP in the port that receives the poisoned TLP from the link logs the error in its AER Capability Structure and handles it appropriately.
  - A PCI-to-PCI bridge function that receives a poisoned TLP handles it as described in Table 10.9.
  - An NT function that receives a poisoned TLP handles it as described in Table 14.4.
  - A DMA function that receives a poisoned TLP handles it as described in Table 15.12.

Figure 14.16 shows an example of a poisoned TLP propagating across the PES32NT8xG2 during an inter-partition transfer. As the TLP is logically routed from the downstream switch port in a first partition, across the NTB, to the downstream switch port in a second partition, each function in the TLP's path logs the poisoned TLP error per the rules described above.

Notes

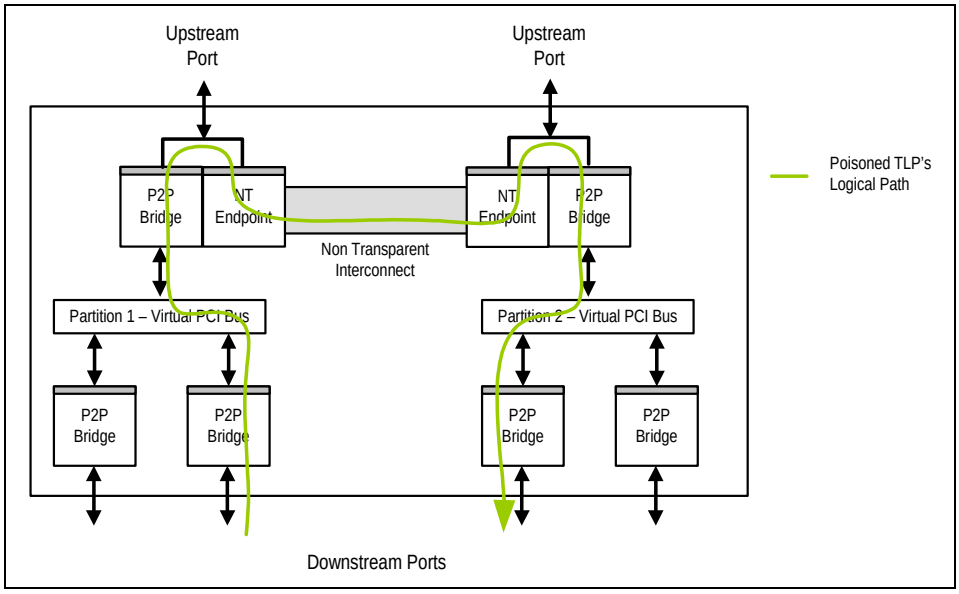

Figure 14.16 Poisoned TLP Error Propagation Example

Table 14.9 shows the error logging for each function in the TLP's logical path.

Function	Error Logging
Downstream PCI-to-PCI Bridge (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9. Note that this port is considered the ultimate receiver in partition 1, as it is the port that receives the TLP on the link and the TLP targets the NT function in partition 1's upstream port (see section Transaction Layer Errors on page 10-13).
Upstream PCI-to-PCI Bridge (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9. Note that the bridge only logs parity error reception in the SECSTS register as it did not receive the TLP directly from the link.
NT Endpoint (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' in Table 14.4. Note that the NT function only logs parity error reception in the PCISTS register as it did not receive the TLP directly from the link.

Table 14.9 Error Logging at Each Function for Poisoned TLP Example (Part 1 of 2)

Notes

Function	Error Logging
NT Endpoint (Partition 2)	Refer to row corresponding to 'Poisoned TLP received' in Table 14.4. In the table, this NT function is considered the "NT endpoint in the destination partition".
Upstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9 Note that the bridge only logs parity error reception in the PCISTS register as it did not receive the TLP directly from the link.
Downstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9 Note that the bridge only logs parity error reception in the PCISTS register as it did not receive the TLP directly from the link.

Table 14.9 Error Logging at Each Function for Poisoned TLP Example (Part 2 of 2)

ACS Errors

ACS checks may cause the offending TLP to be dropped (i.e., ACS violation) or re-directed towards the root-complex by the detecting function. In cases where the TLP is dropped, no other functions in the TLP's logical path detect the ACS error. In cases where the TLP is re-directed towards the root-complex, other functions in the TLP's logical path may perform ACS checks on the TLP.

ACS re-direction may occur at three logical points in a TLPs path:

- Downstream switch port: A TLP received on a downstream switch port that is destined towards another downstream switch port in the same partition is re-directed towards the root-complex (Figure 10.3).
- Upstream port's PCI-to-PCI Bridge function: A TLP received on the upstream PCI-to-PCI bridge function's secondary side that is destined towards the NT function in the same port is re-directed towards the root-complex (Figure 10.5).
- Upstream port's NT function: A TLP transmitted by the upstream NT function's that is destined towards the PCI-to-PCI bridge function in the same port is re-directed towards the root-complex (Figure 14.12).

Refer to section Access Control Services on page 10-6 for details on ACS checks performed by the PCI-to-PCI bridge function, and section Access Control Services (ACS) on page 14-23 for details on ACS checks performed by the NT function.

Combined Transaction Layer Errors

As a TLP is logically routed across the PES32NT8xG2 functions, it is possible for functions to detect different types of transaction layer errors for the TLP. For example, on reception of a TLP, a PCI-to-PCI bridge function may log a poisoned TLP error and forward the TLP to the next function in its logical route. This next function may detect an unsupported request (UR) error on the TLP and handle it accordingly[1].

In general, when a first function receives a TLP, detects an error, and forwards the TLP to the next function on the route, the next function in the TLP's path may not detect an error, may detect the same error, or may detect a higher priority error.

Figure 14.17 shows an example of a poisoned non-posted TLP received on a downstream switch port of a first partition. As the TLP propagates within the first partition, across the NTB, and towards a second partition, the functions in the logical path of the TLP log the poisoned TLP error reception appropriately. As the TLP reaches a downstream switch port in the second partition, the port's PCI-to-PCI bridge function detects

1. Note that UR errors have higher precedence than poisoned TLP errors per the error pollution rules described in section Transaction Layer Error Pollution on page 14-31.

Notes

that the non-posted TLP's request is unsupported (e.g., the downstream switch port's link is down). As a result, the downstream switch port handles the TLP as an supported request error and generates a completion TLP with UR status.

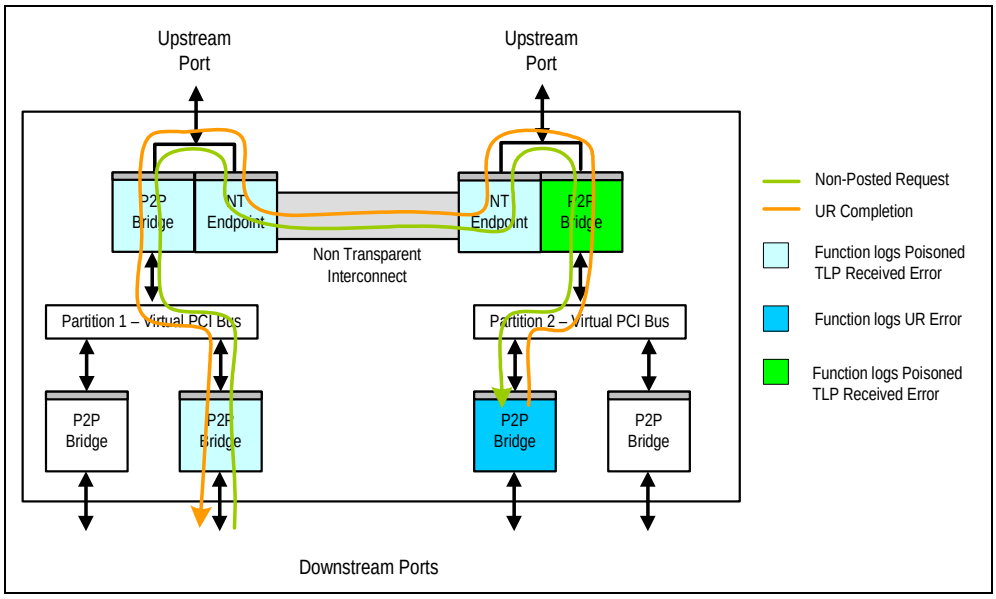

Figure 14.17 Example of Combined Transaction Layer Error Handling

Table 14.10 shows the error logging for each function in the TLP's logical path.

Function	Error Logging
Downstream PCI-to-PCI Bridge (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9. Note that this port is considered the ultimate receiver in partition 1, as it is the port that receives the TLP on the link and the TLP targets the NT function in partition 1's upstream port (see section Transaction Layer Errors on page 10-13).
Upstream PCI-to-PCI Bridge (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9. Note that the bridge only logs parity error reception in the SECSTS register as it did not receive the TLP directly from the link.
NT Endpoint (Partition 1)	Refer to row corresponding to 'Poisoned TLP received' in Table 14.4. Note that the NT function only logs parity error reception in the PCISTS register as it did not receive the TLP directly from the link.

Table 14.10 Error Logging at Each Function for Poisoned TLP Example (Part 1 of 2)

Notes

Function	Error Logging
NT Endpoint (Partition 2)	Refer to row corresponding to 'Poisoned TLP received' in Table 14.4. In the table, this NT function is considered the "NT endpoint in the destination partition". Additionally, refer to row corresponding to 'Completion with UR status received' in Table 14.4.
Upstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Poisoned TLP received' error in Table 10.9. Note that the bridge only logs parity error reception in the PCISTS register as it did not receive the TLP directly from the link.
Downstream PCI-to-PCI Bridge (Partition 2)	Refer to row corresponding to 'Unsupported Request' error in Table 10.9.

Table 14.10 Error Logging at Each Function for Poisoned TLP Example (Part 2 of 2)

Note that the NT function in partition 2 logs the reception of the poisoned TLP reception, as well as the reception of a completion TLP with UR status (i.e., the completion TLP logically generated by the downstream PCI-to-PCI bridge function in partition 2).

Error Emulation Control in the NT Function

The PES32NT8xG2 provides the capability to emulate error occurrence in the AER uncorrectable and correctable error status registers. Associated with the NT function are two error emulation registers. The NT function Uncorrectable Error Emulation (NTUEEM) and NT function Correctable Error Emulation (NTCEEM) registers allow emulation of errors in the NT function.

When a bit in these registers is set, it causes the hardware to emulate the detection of the corresponding error. The detection of the error is handled as shown in Figure 6-2 of the PCI Express 2.1 base specification (i.e., the corresponding error is logged in the AER status registers (i.e., AERUES or AERCES), and reported to the root-complex).

- To allow emulation of advisory errors, the NTUEEM register contains a bit named ADVISORYNF. When this bit is set in conjunction with another bit in the NTUEEM register, the hardware flags the error as an advisory error and handles it according to Figure 6-2 of the PCI Express 2.1 base specification. Refer to the description of this bit for details.

Since the error emulation does not involve an actual TLP, the AER Header Log registers (AERHL[1:4]DW) in the switch have RWL type, such that they may be modified by software to emulate the capturing of the TLP's header.

Error Emulation Usage and Limitations

The following are some usage guidelines and limitations associated with error emulation.

- To emulate the detection of a correctable error:
  - The desired error bit must be set in the NTCEEM register.
- To emulate the detection of an uncorrectable fatal error:
  - The desired error bit must be set in the NTUEEM register.
  - The severity of the error must be set to fatal in the AERUESV register.
- To emulate the detection of an advisory uncorrectable non-fatal error:
  - The desired error bit must be set in the NTUEEM register. The error bit selected must qualify for advisory handling as specified in the PCI Express 2.1 specification. Otherwise, the operation of the emulation logic is undefined.
  - The ADVISORYNF bit must be set in the NTUEEM register.
  - The severity of the error must be set to non-fatal in the AERUESV register.

Notes

Due to a limitation in the hardware, it is not possible to emulate the detection of a non-advisory uncorrectable non-fatal error.

Non Transparent Operation Restrictions

The following lists usage restrictions associated with non-transparent operation.

- TLPs received and translated by the NT function must not map into an NT BAR aperture in the destination partition.
- TLPs received and translated by the NT function must not map into the BAR 0 aperture of the DMA function in the destination partition.[1]
- TLPs received and translated by the NT function must not map into a multicast BAR aperture in the destination partition.
- The PCI Express Base Specification mandates that a requester not issue memory requests whose address/length combination crosses 4 KB boundaries. To honor this requirement, the translated base address(es) in the NT function should be programmed such that the twelve lower bits are set to 0x0.

1. Note that DMA BAR 0 maps the configuration registers of the DMA to PCI memory space. Still, this restriction does not imply that it is not possible to configure the DMA function using requests issued by an agent in another partition. Such an operation is still possible and is accomplished by accessing the DMA configuration registers via the switch's global address space. Refer to Chapter 19, Register Organization, for details.

DMA Controller

Notes

Overview

The PES32NT8xG2 supports two direct memory access controller (DMA) functions. Each DMA function appears as a PCI Express endpoint in the PCI Express hierarchy, located in a partition's upstream port. In each partition, the operating mode of the switch's upstream port determines if this port contains a DMA function. Refer to section Switch Port Mode on page 5-5 for details on the port operating modes.

A DMA function is associated with two DMA channels. A DMA channel is an engine that transfers data between two agents in the PCI Express hierarchy. DMA channels act independently and operate by processing descriptors. DMA channels are programmed via configuration registers in the DMA function's configuration space.

The following sections describe the DMA function and the operation of the DMA channels. The chapter starts by describing PCI Express aspects of the DMA function (e.g., interrupts, base address registers, PCI Express error handling, etc.), followed by a detailed description of DMA channels (e.g., descriptor formats, processing, etc.).

Note: DMA operations are supported only by ports 0 and 8.

Base Address Registers

The DMA function implements one base address register, labeled BAR 0. BAR 0 can be enabled to map the DMA function's 4 KB configuration space in 32-bit or 64-bit system memory. The DMA function does not support mapping of its 4 KB configuration space to PCI I/O Space.

The configuration space of the DMA consists of PCI Express registers as well as proprietary registers associated with the DMA channel operation. Each DMA channel has an independent set of registers to control its operation. BAR 0 has an associated BAR Setup register (BARSETUP0). The BAR setup register allows the BAR to be enabled and configured (e.g., prefetchable memory, 32-bit system memory or 64-bit system memory, etc.). When BAR 0 is enabled, the amount of memory address space requested by the DMA function (i.e., the BAR aperture) is hardwired to 4 KB.

When the DMA function's configuration space is mapped to BAR 0, it is recommended that this configuration space be placed in non-prefetchable space, as some registers may generate side-effect actions when accessed. In addition, memory read or write requests to BAR 0 must specify a length of 1 DWord. Violating this last requirement produces undefined results.

Note that the DMA function's configuration space layout follows little-endian convention. Software executing on a big-endian system should take this into account when accessing the DMA function's configuration space memory-mapped to BAR 0.

DMA Channel Functional Description

Each DMA channel provides a high performance means of moving data. All DMA channels operate independently. This section describes the functional operation of a DMA channel and its programming interface.

A DMA channel operates by reading descriptors from memory, performing the operation outlined by the descriptor, and writing updated descriptor status information back to memory.

– Complex data movement operations, such as scatter/gather, may be implemented by linking DMA descriptors together to form a descriptor list.

Notes

All data transfer operations performed by a DMA channel are memory-to-memory DMA operations.

– Read requests are issued by the DMA channel to read source addresses. As completions corresponding to the request containing data are received by the DMA, the DMA transforms the completion into memory writes to the appropriate destination address.

DMA descriptors, DMA source addresses from where data is read, and DMA destination addresses to where data is written may be located anywhere in the PCI Express memory address space.

– Memory holding DMA descriptors may be located above the upstream port, below a downstream switch port, or in another partition and accessed through an NT endpoint.
  • A DMA descriptor must not be placed at memory address 0x0, as this address is used to indicate the end of a descriptor list (see section DMA Descriptors on page 15-6).
– DMA memory-to-memory transfers may be performed without limitations between any two memory regions accessible from the PCI Express hierarchy associated with which the DMA channel is associated.
  • between memory associated with an upstream port and a downstream switch port
  • between memory associated with two downstream switch ports
  • between two memory regions associated with a single upstream port or single downstream switch port
  • between memory associated with any port and memory in another partition via an NT endpoint
  • between two memory regions associated with another partition via an NT endpoint
  • from any memory region to multiple memory regions, as described above, utilizing multicast
– If the DMA BAR is enabled (section Base Address Registers on page 15-1), the memory regions used for DMA transfers must not overlap with the DMA BAR aperture in memory space. When this occurs, the operation of the DMA is undefined.

Data Transfer and Addressing

Figure 15.1 conceptually illustrates a data transfer operation performed by a DMA channel.

– The DMA channel issues memory read requests TLPs (i.e., MRd) to read data from source memory.
– The source memory responds to a memory read request with one or more completion (Cpl) TLPs that contain the requested data. These completions target the DMA channel.
– When a DMA channel receives a completion, it transforms the completion into a memory write request TLP (i.e., MWr) to destination memory.

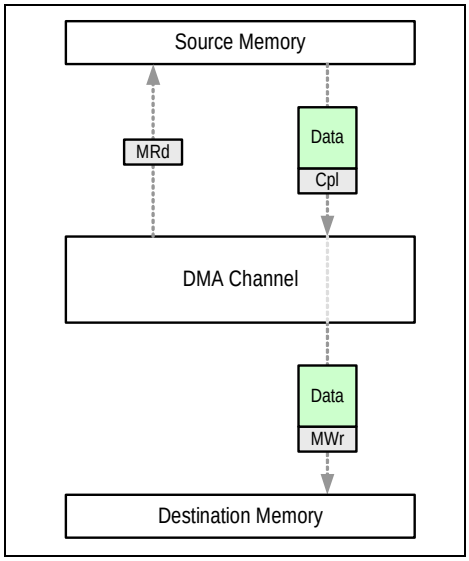

Figure 15.1 DMA Data Transfer

Notes

The determination of what data to read from source memory and where to write this data to destination memory is referred to as the DMA addressing.

The DMA supports two addressing modes: linear addressing and constant addressing. The simplest form of DMA addressing is linear addressing. In linear addressing, a sequential block of data consisting of BCOUNT bytes is transferred from a starting address SADDR to a destination address DADDR. Linear addressing is graphically illustrated in Figure 15.2.

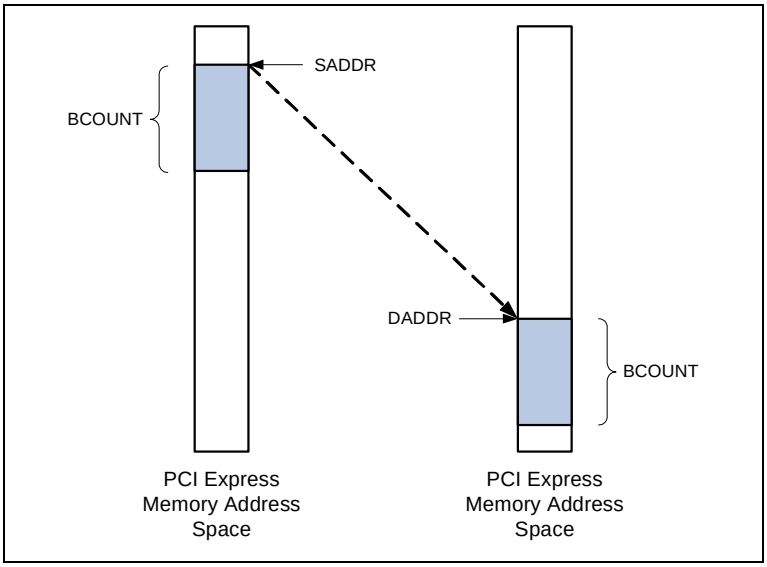

Figure 15.2 Linear Addressing

The operations performed by a DMA channel during linear addressing are shown in Figure 15.3. The starting address of the operation along with the byte count are initialized. Data is then sequentially read or written until the byte count is exhausted. While the operation is shown to proceed a byte at a time, blocks of data are transferred as described in tbd. *Source addressing* refers to the process of reading data from source memory while *destination addressing* refers to the process of writing data to the destination memory. In this example, both source and destination addressing use linear addressing; however, this need not be the case in general.

```
addr = SADDR
for (i=0; i<BCOUNT; i++) {
    data = memRead(addr)
    addr = addr + 1
}
```

(a) Source Addressing

```
addr = DADDR
for (i=0; i<BCOUNT; i++) {
    memWrite(addr,data)
    addr = addr + 1
}
```

(b) Destination Addressing

Figure 15.3 Linear Addressing Operations

The general addressing operations implemented by a DMA channel are shown in Figure 15.4 and the associated parameters are described in Table 15.1. Although both the source and destination addressing algorithms are the same, different parameter initialization may result in different behavior.

Notes

```
count = BCOUNT
addr = SADDR
for (j=0; j<SSCOUNT; ++j) {
    for (i=0; i<SSSIZE; ++i) {
        data = memRead(addr)
        count = count - 1
        if (count == 0) {
            goto done
        }
        addr = addr + 1
    }
    addr = addr + SSDIST
}
```

(a) Source Addressing

```
count = BCOUNT
addr = DADDR
for (j=0; j<DSCOUNT; ++j) {
    for (i=0; i<DSSIZE; ++i) {
        memWrite(addr,data)
        count = count - 1
        if (count == 0) {
            goto done
        }
        addr = addr + 1
    }
    addr = addr + DSDIST
}
```

(b) Destination Addressing

Figure 15.4 DMA Channel Addressing[1]

Global Parameters		
BCOUNT	Byte Count	Total number of bytes to transfer
Source Addressing Parameters		
SADDR	Source Address	Starting source memory address
SSSIZE	Source Stride Size	Amount of data bytes to transfer in each source stride (a value of zero indicates an infinite stride size)
SSCOUNT	Source Stride Count	Number of times to execute the stride loop
SSDIST	Source Stride Distance	Stride distance in bytes (two's complement representation of a positive or negative number)
Destination Addressing Parameters		
DADDR	Destination Address	Starting destination memory address
DSSIZE	Destination Stride Size	Amount of data bytes to transfer in each destination stride (a value of zero indicates an infinite stride size)
DSCOUNT	Destination Stride Count	Number of times to execute the stride loop
DSDIST	Destination Stride Distance	Stride distance in bytes (two's complement representation of a positive or negative number)

Table 15.1 DMA Channel Addressing Parameters

The addressing operation consists of two loops. The inner loop performs linear addressing, while the outer loop implements stride addressing (i.e., used for constant addressing as described later).

1. The pseudo code in this figure does not support the SSSIZE or DSSIZE equal to zero case. See the text for behavior.

Notes

The programming of the addressing parameters must meet the following rules.

1. BCOUNT <= (SSCOUNT * SSSIZE)
2. (SSCOUNT * SSSIZE) = (DSCOUNT * DSSIZE)

An addressing operation completes execution when the byte count is exhausted or, in case the above rules are violated, when stride addressing completes.

- When (SSCOUNT * SSSIZE) < BCOUNT, source addressing expires before the byte count expires and the transfer terminates. The DMA may be configured to report this condition as an error.
- When (DSCOUNT * DSSIZE) < BCOUNT, destination addressing expires before the byte count expires and the transfer terminates. The DMA may be configured to report this condition as an error.

Linear Addressing Example

The simplest form of DMA addressing is a linear addressing from a source address to a destination address as shown in Figure 15.2.

Table 15.2 illustrates the parameters necessary to transfer 1 KB of data from a source address of 0x0100_0000 to a destination address of 0x0200_0001.

- BCOUNT is set to 1024 indicating the total number of bytes to transfer.
- SADDR is set to the source address 0x0100_0000 and DADDR is set to the destination address 0x0200_0001.
- Since strides are not used in this example, SSSIZE and DSSIZE are set to zero. A value of zero indicates an infinite stride. This results in the inner loop executing in Figure 15.4 until the byte count is exhausted.
- All stride counts (i.e., SSCOUNT and DSCOUNT) are set to one to indicate one iteration through the outer loop.
- Since strides are not used, all stride distances (SSDIST and DSDIST) are set to zero.

BCOUNT	1024
SADDR	0x0100_0000
SSSIZE	0
SSCOUNT	1
SSDIST	0

DADDR	0x0200_0001
DSSIZE	0
DSCOUNT	1
DSDIST	0

Table 15.2 Linear Addressing DMA Example

A zero length linear addressing operation (i.e., one with the BCOUNT field set to zero) may be used as a synchronization barrier. Such an operation results in a memory read request to the source address of one DWord with no byte enables set. The operation completes when the corresponding completion is received. No write is performed to the destination address.

- A zero length linear addressing operation may be used to ensure that data associated with memory writes from a previous DMA descriptor operation have reached their ultimate destination before the DMA channel is allowed to proceed.

Constant Addressing Example

By setting the stride distance to a negative value, the DMA can perform constant addressing. Constant addressing is used to repeatedly read/write the same address(es) and is often used when moving data to/from a memory mapped FIFO.

The parameters in Table 15.3 illustrate an example where constant addressing is used to repeatedly read DWord data from a memory mapped FIFO and transfer (DMA) the data to a destination buffer. In this example, there are 1024 bytes are read a DWORD at a time from the address 0x0100_0000. This operation is graphically depicted in Figure 15.5.

Notes

BCOUNT	1024
SADDR	0x0100_0000
SSSIZE	4
SSCOUNT	256
SSDIST	-4

DADDR	0x0200_0001
DSSIZE	0
DSCOUNT	1
DSDIST	0

Table 15.3 Constant Addressing DMA Example

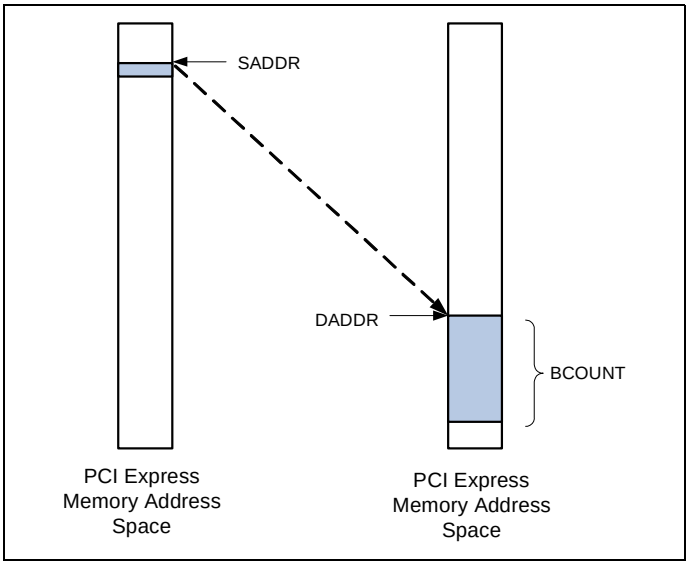

Figure 15.5 Constant Addressing Example

Note that the Source Stride Size (SSSIZE) determines the size of the region from which data is read. In the example above, SSSIZE is set to 4 and SSCOUNT is set to 256 to perform 256 read operations from the same DWord, for a total of 1024 bytes transferred (i.e., BCOUNT is set to 1024). Similarly, the Destination Stride Size (DSSIZE) controls the size of the region to which data is written. In the example above, DSSIZE is set to 0 to indicate an infinite destination stride and DSCOUNT is set to 1 as the destination addressing is linear.

Finally, note the Maximum Read Request Size (MRRS) field in the Data Transfer DMA Descriptor may cause the DMA to issue a single or multiple read operations to transfer the desired data. For the example above, if MRRS is set to 1 byte, then the DMA would issue four requests for each DWord of data transferred.

DMA Descriptors

A DMA channel operates by reading descriptors from memory and performing the specified processing.

Complex data movement operations, such as scatter gather, may be implemented by linking multiple descriptors together into a descriptor list. Figure 15.6 illustrates a descriptor list.

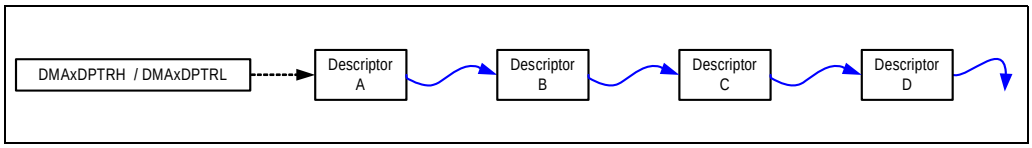

Figure 15.6 DMA Descriptor List

Notes

All DMA descriptors share the same common format shown in Figure 15.7.[1]

- DMA descriptors are eight DWords in size.
- DMA descriptors must be DWord aligned. Processing by a channel of a DMA descriptor with an unaligned DWord address results in an error.
- A DMA descriptor must not be placed at memory address 0x0, as this address is used to indicate the end of a descriptor list.
- DMA descriptors must not cross a 4-KB address boundary. Processing by a channel of a DMA descriptor that violates this rule produces undefined behavior.
- The Descriptor Type (DTYPE) field specifies the type of the descriptor that indicates to the DMA channel how descriptor specific fields should be interpreted and the processing that should be performed by the DMA channel.
- The Descriptor Status (DSTS) field specifies the status of the descriptor. A value of zero indicates that the descriptor has not been processed. A value other than zero indicates that the descriptor has been processed by a DMA channel and indicates the stopping condition (e.g., successful completion or error).
- The Next Lower (NEXTL) and Next Upper (NEXTU) fields together form a 64-bit address that indicates the address of a next descriptor in a descriptor list. A value of zero indicates that this is the last descriptor in a descriptor list.

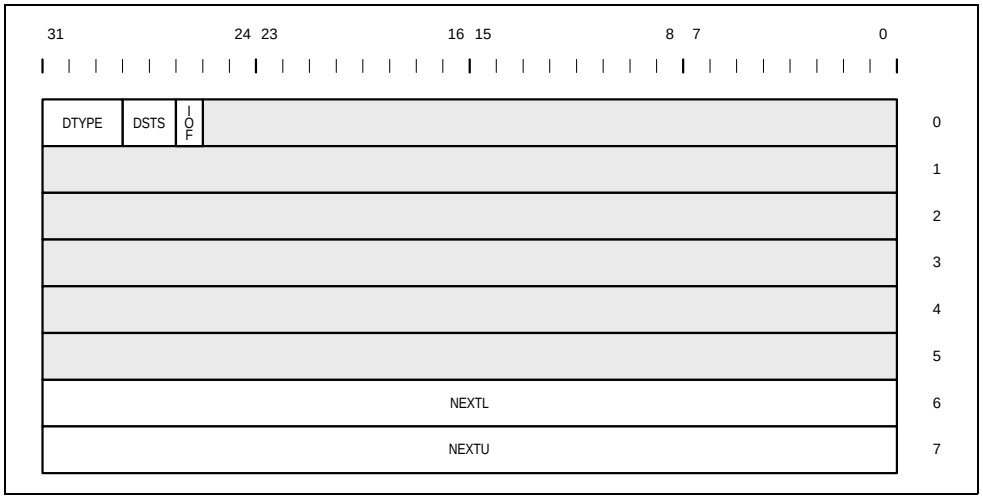

Figure 15.7 General DMA Descriptor Format

When a DMA channel completes processing of a descriptor, it writes back DWord zero with updated status information.

There are two descriptor types supported by the DMA channel. Stride control descriptors are used to modify stride control parameters while data transfer descriptors are used to perform data movement operations.

Stride Control Descriptor

A stride control descriptor is used to modify DMA channel stride parameters, no data is transferred in processing of a stride control descriptor.

- The values loaded by a stride control descriptor are used by all subsequent data transfer descriptors until the processing of the next stride control descriptor.

The format of a stride control descriptor is shown in Figure 15.8 and the fields are described in Table 15.4. Following the PCI convention, the format is shown in little-endian.

[1] Following the PCI convention, the descriptor layout is shown in little-endian.

Notes

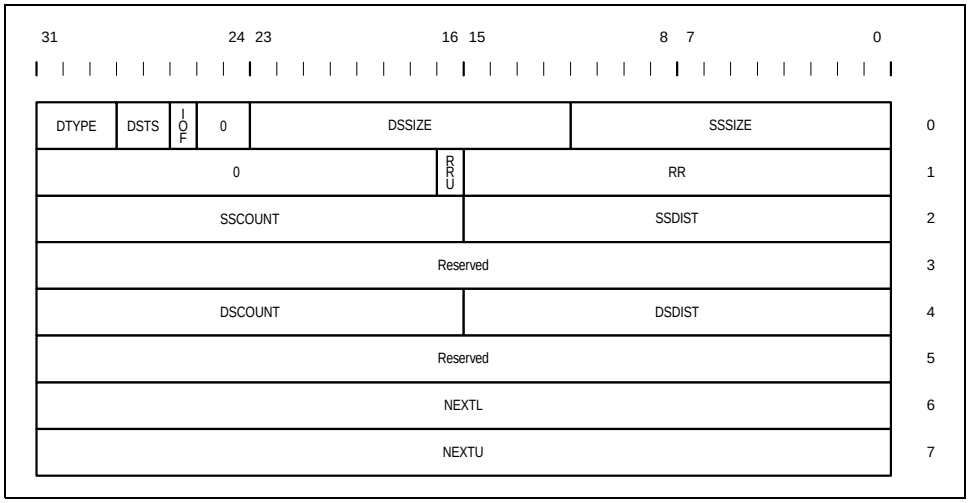

Figure 15.8 Stride Control DMA Descriptor Format

Field	DWord	Bit Position	Description
SSSIZE	0	11:0	**Source Stride Size.** This field specifies the source stride in bytes. A value of zero indicates an infinite stride size.
DSSIZE	0	23:12	**Destination Stride Size.** This field specifies the destination stride in bytes. A value of zero indicates an infinite stride size.
IOF	0	26	**Interrupt on Finished.** When this bit is set and the DMA controller normally finishes processing of the descriptor, then the F bit is set in the corresponding channel DMA Status (DMAxS) register.
DSTS	0	28:27	**Descriptor Status.** A non-zero value in this field indicates that the DMA controller has finished processing the operation associated with this descriptor. 0x0 - Unprocessed descriptor 0x1 - Descriptor processing completed normally (i.e., finished) 0x2 - Reserved 0x3 - Descriptor processing completed due to an error Other - Reserved
DTYPE	0	31:29	**Descriptor Type.** This field encodes the type of descriptor and must be set to 0x3 in stride control DMA descriptors. 0x0 - Reserved 0x1 - Data transfer DMA descriptor 0x2 - Immediate data transfer DMA descriptor 0x3 - Stride control DMA descriptor Others - Reserved
RR	1	15:0	Request Rate. Controls the request rate at which the DMA channel issues data read requests (see section DMA Request Rate Control on page 15-22). The value in this field is used when the Request Rate Update (RRU) field in this descriptor is set

Table 15.4 Stride Control DMA Descriptor Fields (Part 1 of 2)

Notes

Field	DWord	Bit Position	Description
RRU	1	16	**Request Rate Update**. When this bit is set, the DMA channel uses the value in the Request Rate (RR) field of this descriptor to update the DMA Channel Request Rate Control (DMACxRRCTL) register.
SSDIST	2	15:0	**Source Stride Distance.** This field contains the source stride distance in bytes. The value in this field is a signed number in two's complement notation.
SSCOUNT	2	31:16	**Source Stride Count.** This field contains the source stride count. The value in this field is an unsigned number. This field must be non-zero.
Reserved	3	31:0	Reserved.
DSDIST	4	15:0	**Destination Stride Distance.** This field contains the destination stride distance in bytes. The value in this field is a signed number in two's complement notation.
DSCOUNT	4	31:16	**Destination Stride Count.** This field contains the destination count. The value in this field is an unsigned number. This field must be non-zero.
Reserved	5	31:0	Reserved.
NEXTL	6	31:0	**Next Descriptor Address Lower.** Lower 32-bits of 64-bit next descriptor DMA address. Descriptors must be word aligned thus the lower two bits must be zero.
NEXTU	7	31:0	**Next Descriptor Address Upper.** Upper 32-bits of 64-bit next descriptor DMA address.
SSSIZE	0	11:0	**Source Stride Size.** This field specifies the source stride in bytes. A value of zero indicates an infinite stride size.

Table 15.4 Stride Control DMA Descriptor Fields (Part 2 of 2)

The DMA channel performs the following actions as part of stride control DMA descriptor specific processing.

- The value in the Source Stride Size (SSSIZE) field in the descriptor is loaded into the Source Stride Size (SSSIZE) field in the DMA Channel Stride Size (DMACxSSIZE) register.
- The value in the Destination Stride Size (DSSIZE) field in the descriptor is loaded into the Destination Stride Size (DSSIZE) field in the DMACxSSIZE register.
- The value of the Source Stride Distance (SSDIST) field in the descriptor is loaded into the Stride Distance (SDIST) field in the DMA Channel Source Stride Control (DMACxSSCTL) register.
- The value of the Source Stride Count (SSCOUNT) field is in the descriptor is loaded into the Stride Count (SCOUNT) field in the DMACxSSCTL register.
- The value of the Destination Stride Distance (DSDIST) field in the descriptor is loaded into the Stride Distance (SDIST) field in the DMA Channel Destination Stride Control (DMACxDSCTL) register.
- The value of the Destination Stride Count (DSCOUNT) field is in the descriptor is loaded into the Stride Count (SCOUNT) field in the DMACxDSCTL register.

Data Transfer Descriptor

A data transfer descriptor is used to instruct the DMA channel to transfer data from a source device to a destination device. The format of a data transfer descriptor is shown in Figure 15.9 and the fields are described in Table 15.5. Following the PCI convention, the format is shown in little-endian.

Notes

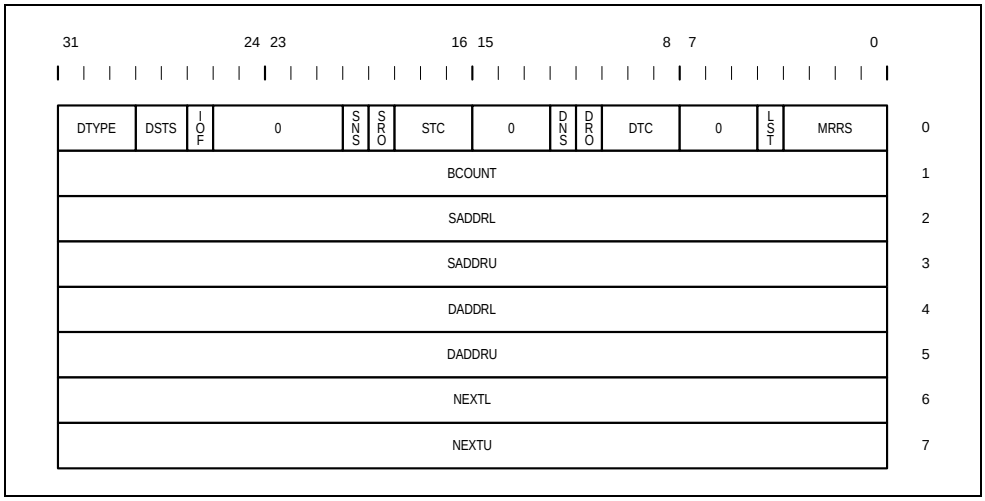

Figure 15.9 Data Transfer DMA Descriptor Format

Field	DWord	Bit Position	Description
MRRS	0	3:0	**Maximum Read Request Size.** This field specifies the maximum DMA source read request size. 0000b - 1B 0001b - 2B 0010b - 4B 0011b - 8B 0100b - 16B 0101b - 32B 0110b - 64B 0111b - 128B 1000b - 256B 1001b - 512B 1010b - 1024B 1011b - 2048B 1100b - 4096B 1101b - Reserved 1110b - Reserved 1111b - Reserved
LST	0	4	**Last.** When this bit is set, it indicates that this descriptor is the last descriptor in the list. Setting this bit is equivalent to setting the next descriptor address fields (NEXTL and NEXTU) to zero. Refer to section DMA Descriptor Processing on page 15-15 for details.
DTC	0	10:8	**Destination Traffic Class.** This field specifies the traffic class used by destination TLPs.
DRO	0	11	**Destination Relaxed Ordering.** This field specifies the state of the relaxed ordering attribute in destination TLPs.
DNS	0	12	Destination No Snoop. This field specifies the state of the no snoop attribute in destination TLPs.
STC	0	18:16	**Source Traffic Class.** This field specifies the traffic class used by source TLPs.

Table 15.5 Data Transfer DMA Descriptor Fields (Part 1 of 2)

Notes

Field	DWord	Bit Position	Description
SRO	0	19	**Source Relaxed Ordering.** This field specifies the state of the relaxed ordering attribute in source TLPs.
SNS	0	20	**Source No Snoop.** This field specifies the state of the no snoop attribute in source TLPs.
IOF	0	26	**Interrupt on Finished.** When this bit is set and the DMA controller normally finishes processing of the descriptor, then the F bit is set in the corresponding channel DMA Status (DMAxS) register.
DSTS	0	28:27	**Descriptor Status.** A non-zero value in this field indicates that the DMA controller has finished processing the operation associated with this descriptor. 0x0 - Unprocessed descriptor 0x1 - Descriptor processing completed normally (i.e., finished) 0x2 - Reserved 0x3 - Descriptor processing completed due to an error Other - Reserved
DTYPE	0	31:29	**Descriptor Type.** This field encodes the type of descriptor and must be set to 0x1 in data transfer DMA descriptors. 0x0 - Reserved 0x1 - Data transfer DMA descriptor 0x2 - Immediate data transfer DMA descriptor 0x3 - Stride control DMA descriptor Others - Reserved
BCOUNT	1	31:0	**Byte Count.** Total number of bytes to transfer.
SADDRL	2	31:0	**Source Address Lower.** Lower 32-bits of 64-bit source DMA address.
SADDRU	3	31:0	**Source Address Upper.** Upper 32-bits of 64-bit source DMA address.
DADDRL	4	31:0	**Destination Address Lower.** Lower 32-bits of 64-bit destination DMA address.
DADDRU	5	31:0	**Destination Address Upper.** Upper 32-bits of 64-bit destination DMA address.
NEXTL	6	31:0	**Next Descriptor Address Lower.** Lower 32-bits of 64-bit next descriptor DMA address. Descriptors must be word aligned thus the lower two bits must be zero.
NEXTU	7	31:0	**Next Descriptor Address Upper.** Upper 32-bits of 64-bit next descriptor DMA address.

Table 15.5 Data Transfer DMA Descriptor Fields (Part 2 of 2)

A data transfer DMA descriptor instructs the DMA channel to perform a data transfer operation. Processing of the descriptor completes when either the data transfer operation completes or an when an error is detected.

- The Source Address Lower (SADDRL) and Source Address Upper (SADDRU) together form a 64-bit source address (SADDR) from which source data is read.
  - The source address may have any byte alignment.
- The Destination Address Lower (DADDRL) and Destination Address Upper (DADDRU) together form a 64-bit destination address (DADDR) to which destination data is written.

Notes

- The destination address may have any byte alignment.

– The Byte Count (BCOUNT) field specifies the number of bytes to transfer.

– The data transfer operation performed in processing the descriptor is controlled by DMA parameters as described in section Data Transfer and Addressing on page 15-2.
  - The SADDR, BADDR and BCOUNT parameters are contained in the data transfer DMA descriptor.
  - The SSSIZE, SSCOUNT, SSDIST, DSSIZE, DSCOUNT, and DSDIST parameters are contained in the DMAxSSIZE, DMAxSSCTL, and DMAxDSCTL registers.

    The default value of these fields enables source and destination linear addressing.

    More complex forms of addressing may be enabled through the use of stride control DMA descriptors.

A data transfer operation consists of performing memory read operations from the source address and when completions for these reads arrive at the DMA, transforming them into memory write operations to the destination address.

– The DMA controller initiates memory read operations from the source address by generating a memory read request (MRd) TLP to the PCI Express port associated with the source address.
  - If the source address is below 4 GB, then a MRd TLP with a 32-bit address is generated. If the address is above 4 GB, then a MRd TLP with a 64-bit address is generated.

– The DMA channel always attempts to issue the maximum sized memory read request possible that is less than or equal to the Maximum Memory Read Request (MRRS) field, does not transfer more sequential data than that required by the source addressing mode, and that does not cause the read request to cross a 4-KB boundary.
  - The memory read request size determines the value of the LENGTH, 1st DW BE, and Last DW BE field in the MRd TLP header
  - The MRRS field in the descriptor must be initialized to a value that is less than or equal to the value of the MRRS field in the PCI Express Device Control (PCIEDCTL) register of the PCI function with which the DMA channel is associated. No error check is performed by the DMA channel and failure to follow this requirement produces undefined results.

– In response to a memory read request, one or more completions are returned. These completions are transformed by the DMA channel into memory write requests (MWr) to the PCI Express port associated with the destination address.
  - The address of the memory write is the next destination address to be written.

    If the address is below 4 GB, then a MWr TLP with a 32-bit address is generated. If the address is above 4 GB, then a MWr TLP with a 64-bit address is generated.
  - The DMA channel transforms completions into memory write requests in a fly-by manner by replacing the completion TLP header with a memory write request TLP header. However, as outlined below the DMA controller may sometimes split completions into multiple memory write requests.

    Since the source and destination addresses may have any byte alignment, the DMA controller may need to “shift” data when transforming a completion into a memory write request. This shifting may cause data from the last DWord of a completion to be moved into the next TLP when a completion with a payload of maximum size is received. When this occurs, this additional data is transferred in a subsequent memory write request.

    In PCI Express, requests must not specify an address/length combination that causes a memory space access to cross a 4-KB boundary. Since there is no relationship between source and destination addresses, the DMA controller may split a completion if the corresponding memory write were to otherwise cross a 4-KB boundary.

    The destination addressing mode may require that completion data be written to two or more non-sequential addresses. When this occurs, the DMA controller splits the completion into memory write requests as required.

    Note that per PCI Express rules, completions associated with different requests have no ordering relationship. Thus, such completions may not be received by a requester (e.g., the DMA) in the same order that the requests were issued. The DMA converts these com-

Notes

pletions to memory write requests on the fly. As a result, the memory writes issued by the DMA may not arrive at the destination location in the order in which the read requests were issued. A user that wishes to keep a strict order between the order in which the bytes are read and written, may do so by programming the descriptor such that the address/length combination does not cross 4KB boundaries, and ensuring that the Byte Count (BCOUNT) field is less than or equal to the Maximum Read Request Size (MRRS) field. Alternatively, the DMA channel may configured for one outstanding request, as described in section DMA Outstanding Requests on page 15-21.

Immediate Data Transfer Descriptor

An immediate data transfer descriptor is used to instruct the DMA channel to transfer data that is embedded in the descriptor to a destination device (i.e., there is no source device associated with the DMA transfer).

The format of an immediate data transfer descriptor is shown in Figure 15.10 and the fields are described in Table 15.6. Following the PCI convention, the format is shown in little-endian.

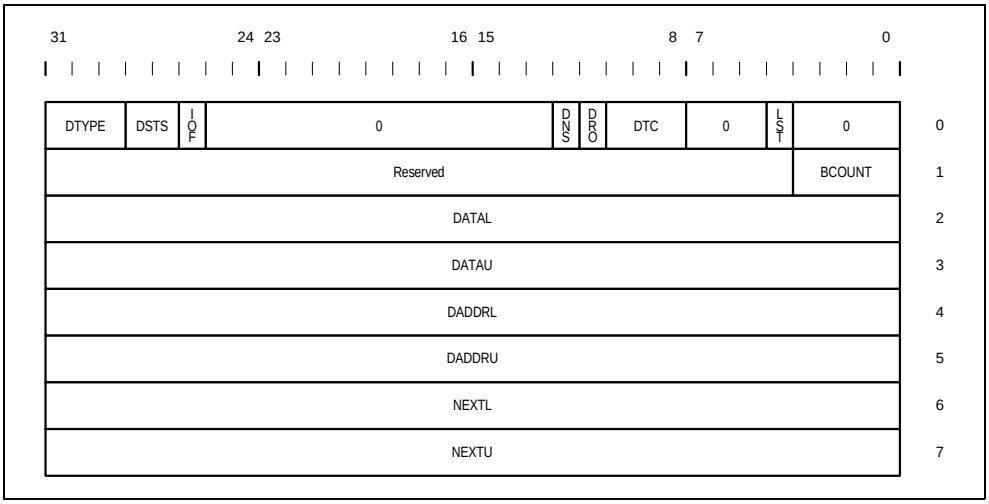

Figure 15.10 Immediate Data Transfer DMA Descriptor Format

Field	DWord	Bit Position	Description
LST	0	4	**Last.** When this bit is set, it indicates that this descriptor is the last descriptor in the list. Setting this bit is equivalent to setting the next descriptor address fields (NEXTL and NEXTU) to zero. Refer to section DMA Descriptor Processing on page 15-15 for details.
DTC	0	10:8	**Destination Traffic Class.** This field specifies the traffic class used by destination TLPs.
DRO	0	11	**Destination Relaxed Ordering.** This field specifies the state of the relaxed ordering attribute in destination TLPs.
DNS	0	12	Destination No Snoop. This field specifies the state of the no snoop attribute in destination TLPs.

Table 15.6 Immediate Data Transfer DMA Descriptor Fields (Part 1 of 2)

Notes

Field	DWord	Bit Position	Description
IOF	0	26	**Interrupt on Finished.** When this bit is set and the DMA controller normally finishes processing of the descriptor, then the F bit is set in the corresponding channel DMA Status (DMAxS) register.
DSTS	0	28:27	**Descriptor Status.** A non-zero value in this field indicates that the DMA controller has finished processing the operation associated with this descriptor. 0x0 - Unprocessed descriptor 0x1 - Descriptor processing completed normally (i.e., finished) 0x2 - Reserved 0x3 - Descriptor processing completed due to an error Other - Reserved
DTYPE	0	31:29	**Descriptor Type.** This field encodes the type of descriptor and must be set to 0x2 in immediate data transfer DMA descriptors. 0x0 - Reserved 0x1 - Data transfer DMA descriptor 0x2 - Immediate data transfer DMA descriptor 0x3 - Stride control DMA descriptor Others - Reserved
BCOUNT	1	3:0	**Byte Count.** Total number of bytes to transfer. Immediate data transfer descriptors can transfer a maximum of 8 bytes. 0x1 - Transfer 1 byte 0x2 - Transfer 2 bytes 0x3 - Transfer 3 bytes 0x4 - Transfer 4 bytes 0x5 - Transfer 5 bytes 0x6 - Transfer 6 bytes 0x7 - Transfer 7 bytes 0x8 - Transfer 8 bytes Others - Reserved
DATAL	2	31:0	Data Lower. Data to be written into the destination device (first 4 bytes).
DATAH	3	31:0	Data Upper. Data to be written into the destination device (last 4 bytes).
DADDRL	4	31:0	**Destination Address Lower.** Lower 32-bits of 64-bit destination DMA address.
DADDRU	5	31:0	**Destination Address Upper.** Upper 32-bits of 64-bit destination DMA address.
NEXTL	6	31:0	**Next Descriptor Address Lower.** Lower 32-bits of 64-bit next descriptor DMA address. Descriptors must be word aligned thus the lower two bits must be zero.
NEXTU	7	31:0	**Next Descriptor Address Upper.** Upper 32-bits of 64-bit next descriptor DMA address.

Table 15.6 Immediate Data Transfer DMA Descriptor Fields (Part 2 of 2)

An immediate data transfer DMA descriptor instructs the DMA channel to perform a memory write operation of the data embedded in the descriptor to the destination address. Processing of the descriptor completes when the memory write operation is issued (i.e., posted TLP(s)) or an when an error is detected.

– The Destination Address Lower (DADDRL) and Destination Address Upper (DADDRU) together form a 64-bit destination address (DADDR) to which destination data is written.
  • The destination address may have any byte alignment.

Notes

- If the address is below 4 GB, then a MWr TLP with a 32-bit address is generated. If the address is above 4 GB, then a MWr TLP with a 64-bit address is generated.

– The Byte Count (BCOUNT) field specifies the number of bytes to transfer.
  - Transferred bytes are written contiguously to the locations starting at the destination address.
  - Setting this field to a reserved value results in no data being transferred.
– The Data Lower field contains the first 4 bytes of data to be transferred.
  - DATAL[7:0] corresponds to the first byte, DATAL[15:8] corresponds to the second byte, and so on.
– The Data Upper field contains the next 4 bytes of data to be transferred.
  - DATAL[7:0] corresponds to the fifth byte, DATAL[15:8] corresponds to the sixth byte, and so on.
  - The Data Upper field is valid when the Byte Count indicates that 5 or more bytes. Otherwise, the data in this field is not transferred.
– Immediate data transfers do not support constant addressing. Data bytes are written contiguously to the target location(s), starting at the specified destination address.
  - In PCI Express, requests must not specify an address/length combination that causes a memory space access to cross a 4-KB boundary. The DMA controller may generate multiple memory write TLPs if the transfer were to otherwise cross a 4-KB boundary.

DMA Descriptor Processing

DMA descriptor processing consists of reading a descriptor from memory, executing the operation outlined by the descriptor, writing back updated descriptor completion status and then proceeding to the next descriptor.

Descriptor List Processing

DMA descriptor processing is initiated as a result of the following events when the Error (E) bit in the DMAxSTS register is cleared.

– Setting of the Run (RUN) bit in the DMA Channel Control (DMACxCTL) register when the DMA channel is idle.
– As a side effect of writing a non-zero value to the DMA Channel Descriptor Pointer Low (DMACxDPTRL) register.
  - Initiation of a DMA descriptor processing as a side effect of writing to the DMACxDPTRL register may be disabled by setting the Disable DMACxDPTRL Descriptor Processing Initiation (DISADPTRL) bit in the DMA Channel Configuration (DMAxCFG) register.
– As a side effect of writing a non-zero value to the DMA Channel Descriptor Pointer High (DMACxDPTRH) register.
  - Initiation of a DMA descriptor processing as a side effect of writing to the DMACxDPTRH register may be disabled by setting the Disable DMACxDPTRH Descriptor Processing Initiation (DISADPTRH) bit in the DMA Channel Configuration (DMAxCFG) register.
– As a side effect of writing a non-zero value to the DMA Channel Next Descriptor Pointer Low (DMACxNDPTRL) or DMA Channel Next Descriptor Pointer High (DMACxNDPTRH) registers, as explained in section Descriptor Chaining on page 15-16.

The DMACxDPTRL and DMACxDPTRH registers together form a 64-bit descriptor address. When DMA descriptor processing is initiated, the DMA controller reads and begins processing the descriptor at the address pointed to by the 64-bit descriptor address.

– When DMA descriptor processing is initiated, the RUN bit in the DMAxCTL register is set.

When the DMA channel finishes processing of a descriptor, updated descriptor status is written back to memory. When writing back the updated descriptor status, the DMA uses PCI Express byte enables to only update the fourth byte in the first DWord of the descriptor (i.e., the memory byte where the DSTS field is located).

If the next descriptor address is non-zero (i.e., NEXTL/H are non-zero), then the DMA channel proceeds to process the descriptor located at that address. If the next descriptor address is zero or if the LST bit in the descriptor is set to 0x1, and chaining is not enabled as described in section Descriptor Chaining on page 15-16, then the DMA channel halts descriptor processing.

Notes

When the DMA channel halts descriptor processing it sets the Halt (H) bit in the DMA Channel Status (DMACxSTS) register and clears the Run (RUN) bit in the DMACxCTL register.

- The DMACxDPTRL and DMACxDPTRH registers continue to hold the value of the last descriptor that was fetched.
- If the RUN bit is set again by software, the DMA channel re-starts descriptor processing by fetching the descriptor pointed to by the DMACxDPTRL and DMACxDPTRH registers.
- If a descriptor to be processed by a DMA channel is read from memory and contains a non-zero Descriptor Status (DSTS) field (i.e., the descriptor status is not “unprocessed descriptor”), the condition is handled as follows.
  - If the Descriptor Status Check Processing (DSCP) field in the DMACxCFG register is set to “process descriptor”, the descriptor is processed normally.
  - If the DSCP field is set to “process next descriptor”, the descriptor is not processed (i.e., no DMA transfer is performed for the descriptor and the descriptor is not written back). Instead, the DMA channel updates the DMACxDPTRL and DMACxDPTRH registers with the descriptor’s NEXTL and NEXTH fields respectively, and starts processing the descriptor pointed to by these fields (i.e., next descriptor in the list). Refer to section Dynamic Appending of Descriptor Lists on page 15-19 for details.
  - If the DSCP field is set to “abort processing”, descriptor processing is aborted and the condition is handled as an error.

When a DMA finishes processing of a DMA descriptor normally without error and the Interrupt on Finished (IOF) bit set in the descriptor, then the Finished (F) bit is set in the DMA Channel Status (DMACxSTS) register.

Descriptor Chaining

Without descriptor chaining, a DMA channel halts descriptor processing when it reaches the last descriptor in a descriptor list (i.e., one with the NEXTL and NEXTH fields set to zero or when the LST bit in the descriptor is set to 0x1).

DMA chaining is enabled by initializing the DMA Channel Next Descriptor Pointer Low (DMACxNDPTRL) and DMA Channel Next Descriptor Pointer High (DMACxNDPTRL) with the starting address of a descriptor list. When the DMA channel completes processing the last descriptor in a descriptor list (i.e., one with a NEXTL/H field value of zero or the LST bit set to 0x1) and the DMACxNDPTRL/H are non-zero, then the DMA controller performs the following actions.

- The DMACxDPTRL register is loaded with the value in the DMACxNDPTRL register
- The DMACxDPTRH register is loaded with the value in the DMACxNDPTRH register
- The contents of the DMACxNDPTRL and DMACxNDPTRH registers are set to zero.
- The Chain (C) bit is set in the DMACxSTS register.
- The DMA controller continues processing descriptors starting with the descriptor pointed to by the DMACxDPTRL/DMACxDPTRH registers. When the last descriptor of the new list is reached, the process repeats.

An example of DMA chaining is shown in Figure 15.11. In this example the DMACxDPTRL/H registers are initialized with the starting address of descriptor list ABCD, and DMACxNDPTRL/H registers are initialized with the starting address of the descriptor list WXYZ. When the DMA channel completes processing descriptor D, the value of DMACxNDPTRL/H is transferred into DMACxDPTRL/H, DMACxNDPTRL/H is set to zero, the C bit is set in the DMACxSTS register, and the DMA continues processing DMA descriptor W. If the DMACxNDPTR register is not updated, then when the DMA channel completes processing descriptor Z, it sets the H bit in the DMACxSTS register, clears the RUN bit in the DMACxCTL register and halts.

Notes

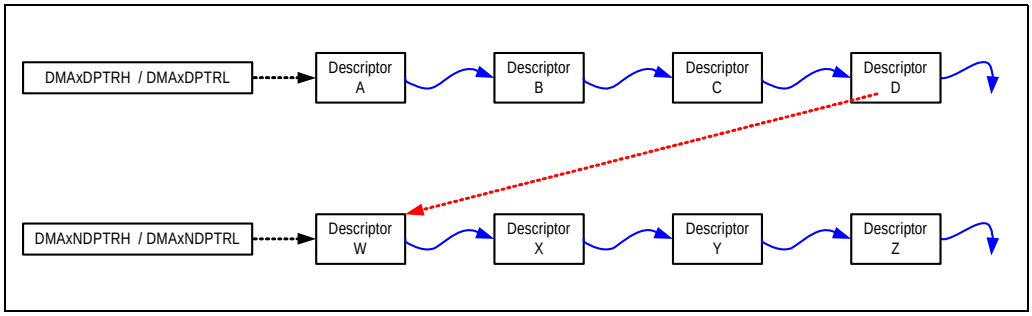

Figure 15.11 DMA Chaining Example

Writing to the DMACxNDPTRL/H registers while the DMA is running (i.e., the RUN bit is set in the DMACxCTL register) simply modifies the register value. Writing a non-zero value to the DMACxNDPTRL/H registers while the DMA is not running (i.e., the RUN bit is cleared) and the Error (E) bit is cleared in the DMACxSTS register not only modifies the register value, but also causes a descriptor chaining operation to take place.

- The DMACxDPTRL/H register is loaded with the value in the DMACxNDPTRL/H register
- The contents of the DMACxNDPTRL and DMACxNDPTRH registers are set to zero.
- The Chain (C) bit is set in the DMACxSTS register.
- The DMA controller starts processing descriptors starting with the descriptor pointed to by the DMACxDPTRL/H registers.

Automatic execution of a descriptor chaining operation may be disabled. This provides a race-free mechanism for software to update the DMA Channel Next Descriptor Pointer registers (DMACxNDPTRH/L) with a 64-bit value while the DMA is running.Initiation of a descriptor chaining operation as a side effect of writing to the DMACxNDPTRL register may be disabled by setting the Disable DMACxNDPTRL Descriptor Processing Initiation (DISANDPTRL) bit in the DMA Channel Configuration (DMAxCFG) register.

Execution of a descriptor chaining operation as a side effect of writing to the DMACxNDPTRL register may be disabled by setting the Disable DMACxNDPTRL Descriptor Processing Initiation (DISNDPTRL) bit in the DMA Channel Configuration (DMAxCFG) register.

Execution of a descriptor chaining operation as a side effect of writing to the DMACxNDPTRH register may be disabled by setting the Disable DMACxNDPTRH Descriptor Processing Initiation (DISNDPTRH) bit in the DMA Channel Configuration (DMAxCFG) register.

Table 15.7 shows the behavior of the DMA with respect to the setting of the DISNDPTRH and DISNDPTRL bits in the DMACxCFG register.

DMAxCFG. DISNDPTRH	DMAxCFG. DISNDPTRL	DMA Chaining Behavior
0	0	Execute DMA chaining if (DMACxNDPTRH != 0) or (DMACxNDP-TRL != 0)
0	1	Execute DMA chaining if (DMACxNDPTRH != 0)
1	0	Execute DMA chaining if (DMACxNDPTRL != 0)[1]
1	1	DMA Chaining is disabled

Table 15.7 DMA Chaining Disabling

1. The DMACxNDPTRL and DMACxNDPTRH registers must never be written with the value of 0x0. Doings so produces an undefined operation.

Notes

Aborting a DMA Operation

The processing of DMA descriptors by a DMA channel may be aborted by writing a one to the Abort (ABORT) bit in the DMACxCTL register. When a DMA operation is aborted due to this condition, the following actions take place:

- The DMA channel ceases to issue read requests. This includes DMA descriptor read requests and data read requests.
- The DMA channel waits for completions, or completion time-outs, for all outstanding memory read requests and processes these normally (e.g., completions for outstanding data read requests are converted to memory write requests, completions for outstanding descriptor read requests are processed). During this time, if the DMA finishes processing of a descriptor, the DMA may write-back the descriptor prior to aborting the operation.
- The DMACxDPTRL/H registers point to the descriptor whose processing was aborted.
- All prefetched descriptors (see section Descriptor Prefetching on page 15-22) are discarded.
- When all of the above complete, the Abort (A) bit is set in the DMACxSTS register.
  - If the DMA channel is idle when the Abort bit is written, then the A bit in the DMACxSTS register is immediately set.

The processing of a DMA descriptors by a DMA channel may also be aborted as a side effect of error detection during descriptor processing (refer to section Error Handling on page 15-27). When a DMA operation is aborted due to an error, the following actions take place:

- The DMA channel ceases to issue requests. This includes DMA descriptor read requests, source address read requests, and memory write requests.
- The DMA channel waits for completions, or completion time-outs, for all outstanding memory read requests. Received completions are discarded (i.e., the completions are not converted to memory write requests).
- The DMACxDPTRL/H registers point to the descriptor whose processing was aborted.
- All prefetched descriptors are discarded.
- The bit corresponding to the error is set in the DMA Channel Error Status (DMACxERRSTS) register.
- When all of the above complete, the Abort (A) bit and the Error (E) bit are set in the DMACxSTS register.

Suspending and Resuming a DMA Operation

In some applications it is desirable to append descriptors to an active descriptor list (i.e., one that is being processed by a DMA channel). The DMA channel suspend and resume capability provides a race-free mechanism to perform this operation.

- For DMA descriptors located below 4 GB, it is possible to resolve the descriptor update/read race condition without using the suspend/resume mechanism described in this section. Instead, the more efficient mechanism described in section Dynamic Appending of Descriptor Lists on page 15-19 may be used.
- DMA channel suspend and resume requires that the DSCP field in the DMACxCFG register be set to “process next descriptor” prior to enabling a DMA channel (see section Descriptor List Processing on page 15-15), and descriptor chaining must not be enabled (see section Descriptor Chaining on page 15-16).

The operation of a DMA channel may be suspended by writing a one to the Suspend (SUSPEND) bit in the DMAxCTL register. When a DMA operation is suspended, the following actions take place:

- If the DMA is processing a descriptor, the DMA channel suspends processing after writing back the updated descriptor status to memory, but without loading the DMACxDPTRL/H registers to point to the next descriptor. Once the suspend operation takes place, the Suspend (S) bit is set in the DMACxSTS register.

Notes

- All prefetched descriptors are discarded.

– If the DMA channel is halted when suspended (i.e., the DMA has completed processing descriptors in a list), then the Suspend (S) bit in the DMACxSTS register is immediately set. Software should wait for the Suspend bit in the DMACxSTS register to be set prior to resuming the DMA channel as described next. Violating this rule produces undefined behavior.

The RUN bit remains set while a DMA is suspended. A suspended DMA channel may be resumed by simultaneously writing a zero to the S bit and a one to the RUN bit in the DMACxCTL register while the Error (E) bit in the DMAxSTS register is cleared. The DMA channel re-reads the descriptor pointed to by the DMACxDPTRL/H registers. This register points to the last descriptor processed before the suspend. The re-fetched descriptor is not processed due to the setting of the DSCP field in the DMACxCFG register. Instead, the DMA follows the re-fetched descriptor's NEXTL/H fields and starts processing the next descriptor.

A summary of DMACxCTL register writes and their effect on the DMA channel is provided in Table 15.8.

Run	Abort	Suspend	Action
0	0	X	No effect on operation of DMA channel
X	0	1	Suspend DMA
X	1	X	Abort DMA channel operation
1	0	0	No action if E bit in the DMACxSTS register is set. Initiate DMA channel operation if RUN bit in the DMACxCTL register and the E bit in the DMACxSTS are cleared. Resume DMA channel operation if RUN bit in the DMACxCTL register was set and the E bit in the DMACxSTS register is cleared.
X	1	1	Undefined behavior

Table 15.8 DMA Channel Control (DMACxCTL) Register Action Summary

Dynamic Appending of Descriptor Lists

Refer to section Suspending and Resuming a DMA Operation on page 15-18 for a description of a race-free mechanism to append or modify descriptors in an active descriptor list by suspending and resuming descriptor processing in a DMA channel. For scenarios where a suspend/resume operation is not desired (e.g., to improve DMA performance), this section presents an alternative approach to performing dynamic appending of descriptors to a descriptor list.

The mechanism described in this section is only applicable for dynamic appending of descriptor lists located below 4 GB. For descriptor lists above the 4 GB address, the mechanism described in section Suspending and Resuming a DMA Operation on page 15-18 should be used.

To enable this usage model, the Descriptor Status Check Processing (DSCP) field in the DMACxCFG register must be set to "process next descriptor" prior to enabling a DMA channel (see section Descriptor List Processing on page 15-15), and descriptor chaining must not be enabled (see section Descriptor Chaining on page 15-16).

Descriptors may be appended to an active descriptor list (i.e., a descriptor list which a DMA channel is currently processing), by modifying the NEXTL field of the last descriptor in the current descriptor list or by modifying the LST bit of the last descriptor in the current descriptor list.

Note that the DMA engine never modifies the NEXTL field or the LST bit in a descriptor during descriptor write-back. Therefore, there is no conflict between software that modifies the NEXTL field or the LST bit and the DMA engine write-back operation. Also note that the LST bit is not applicable to Stride Control descriptors.

Notes

After descriptors are appended to an active descriptor list, software must set the RUN bit in the DMACxCTL register[1]. If the DMA channel had not initiated processing of the last descriptor in the original list at the time the RUN bit is set by software, the DMA channel continues processing descriptors normally, including the newly appended descriptors.

– If descriptor pre-fetching is enabled, the DMA channel discards all prefetched descriptors and re-fetches them.

If the DMA channel had initiated or completed descriptor processing of the last descriptor in the original list at the time the RUN bit is set (i.e., the DMA channel halted after processing the last descriptor in the list), the DMA channel re-starts descriptor processing as described in section Descriptor List Processing on page 15-15.

– The DMACxDPTRL and DMACxDPTRH hold the address of the last descriptor that was processed by the DMA channel. This descriptor will be re-fetched by the DMA, but will not be re-processed (i.e., due to the setting of the DSCP field in the DMACxCFG register). Instead, processing will re-start at the next descriptor in the list (i.e., the newly appended descriptor(s)).

TLP Attribute and Traffic Class Control

Each DMA channel provides fine grain control over request TLP attributes.

– The relaxed ordering attribute in a TLP header allows TLP ordering rules to be relaxed. Setting the relaxed ordering attribute, when safe, can improve performance by reducing head-of-line blocking.
  • The relaxed ordering controls described below are only applicable when the Enable Relaxed Ordering (ENO) bit in the DMA function's PCI Express Device Control (PCIEDCTL) register is set. When this bit is cleared, the DMA function does not set the relaxed ordering attribute of TLPs it generates.

– The no snoop attribute in a TLP header provides a hints regarding coherence requirements. Setting the no-snoop attribute, when safe, can improve performance by eliminating snoop operations.
  • The No Snoop controls described below are only applicable when the Enable No Snoop (ENS) bit in the DMA function's PCIEDCTL register is set. When this bit is cleared, the DMA function does not set the No Snoop attribute of TLPs it generates.

The traffic class (TC) of a TLP may be used to map a TLP to a specific virtual channel.

Descriptor Attribute and Traffic Class Control

The state of attributes in DMA descriptor memory read and write operations may be independently controlled.

– The Descriptor Read Relaxed Ordering (DRRO) bit in the DMA Channel Configuration (DMAxCFG) register controls the relaxed ordering attribute in DMA descriptor read operations.

– The Descriptor Read No Snoop (DRNS) bit in the DMAxCFG register controls the no snoop attribute in DMA descriptor read operations.

– The Descriptor Write Relaxed Ordering (DWRO) bit in the DMAxCFG register controls the relaxed ordering attribute in DMA descriptor write operations.

– The Descriptor Write No Snoop (DWNS) bit in the DMAxCFG register controls the no snoop attribute in DMA descriptor write operations.

The Descriptor Traffic Class (DTC) field in the DMAxCFG register specifies the traffic class used for both read and write DMA descriptors.

[1] Setting of the RUN bit by software overcomes any attempt by the hardware to clear this bit in the same clock cycle.

Notes

Data Transfer Attribute and Traffic Class Control

The state of memory request TLP attributes and traffic class may be independently controlled for memory read and write operations on a descriptor by descriptor basis by fields in the data transfer DMA descriptor.

- The Source Relaxed Ordering (SRO) field in a descriptor specifies the state of the relaxed ordering attribute in memory read request TLPs used to transfer data. The Destination Relaxed Ordering (DRO) field specifies the state of the relaxed ordering attribute in memory write request TLPs.
- The Source No Snoop (SNS) field in a descriptor specifies the state of the no snoop attribute in memory read request TLPs used to transfer data. The Destination No Snoop (DNS) field specifies the state of the no snoop attribute in memory write request TLPs.
- The Source Traffic Class (STC) field in a descriptor specifies the traffic class of memory read request TLPs used to transfer data. The Destination Traffic Class (DTC) field in a descriptor specifies the traffic class of memory write request TLPs.

Channel Interrupts

The following DMA channel events result in a bit being set in the DMA Channel Status (DMACxSTS) register.

- Normal completion of a descriptor operation with the IOF bit set in the descriptor.
- When descriptor processing is aborted.
- When DMA descriptor chaining occurs.
- When a DMA channel halts descriptor processing.
- When a DMA channel suspends descriptor processing.
- When an unmasked error bit is set in the DMA Channel Error Status (DMACxERRSTS) register.

The assertion of a bit in the DMACxSTS register may be used to generate an interrupt in the DMA function with which the channel is associated.

- Associated with each bit in the DMACxSTS register is a corresponding bit in the DMACxMSK register. When a bit in the DMACxMSK register is set, the corresponding bit in the DMAxSTS register is masked from generating an interrupt.

Refer to section Interrupts on page 15-24 for further details.

DMA Outstanding Requests

When processing a descriptor, a DMA channel transfers data by reading it from a given location and writing it to another location. Per PCI Express rules, the DMA channel must break read requests that cross addresses aligned at 4 KB boundaries. Thus, processing a descriptor may require the DMA channel to issue several memory read requests.

Each read operation has an associated latency (i.e., the amount of time between the read request issued by the DMA and the corresponding completion arriving at the DMA). The latency is dependent on the PCI Express hierarchy structure, link widths, completer latency, etc.

This latency is incurred every time the DMA issues the first read request associated with a descriptor processing. To aid in “hiding” this latency for subsequent read requests while processing the same descriptor, each DMA channel supports two outstanding read requests. That is, a DMA channel is capable of issuing a read request before the completion for a prior read request is received.

- DMA outstanding requests are not issued across descriptor boundaries.

The performance of the DMA engine improves when the DMA issues few read requests when processing a descriptor, and each read request transfers a large amount of data. Note that the DMA addressing specified in the descriptor determines the boundaries at which the DMA channel breaks requests. Linear addressing causes the DMA to break requests at every 4 KB of data transfer (on average).

Notes

Descriptor Prefetching

When the amount of data moved by data transfer descriptors is small (e.g., when moving data associated with 64B packets), the overhead in fetching DMA descriptors from memory between data transfer operations may limit performance.

To overcome this overhead, the DMA channel supports descriptor prefetching. When descriptor prefetching is enabled, the DMA channel issues memory read requests for DMA descriptors before they are required. This allows the memory latency associated with fetching descriptors to be overlapped with data transfer operations.

- When descriptor prefetching is enabled, the DMA controller follows the NEXTL/NEXTH field and issues memory read requests for descriptors before they are required (i.e., before the descriptor processing associated with a previous descriptor has completed).
- The DMA channel queues prefetched descriptors until they are processed. Prefetched descriptors are discarded when DMA channel operation is suspended or aborted.[1]
- Descriptor prefetching stops when the end of a descriptor list is reached. Descriptor prefetching does not initiate descriptor chaining.

By default, descriptor prefetching is disabled in the DMA. Descriptor prefetching may be enabled through the DMA Descriptor Prefetch Level (DPREFETCH) field in the DMA Channel Configuration (DMACxCFG) register.

DMA Request Rate Control

By default, a DMA channel issues data transfer memory read requests when needed. While this provides the highest level of performance, it can also lead to congestion in the PCI Express topology and tax memory bandwidth.

To support background DMA operations (i.e., ones that consume only a fraction of available system bandwidth), each DMA channel supports a request rate control capability. Request rate control is enabled and controlled by the value in the Request Rate (RR) field of the DMA Channel Request Rate Control (DMACxRRCTL) register.

When request rate control is enabled, a counter is loaded when each data transfer memory read request (MRd) TLP is issued with the number of DWords requested by that TLP. While non-zero, the counter is decremented by the rate indicated by the RR field. For example, a value of one in the RR field indicates that the counter is decremented every 4 ns while a value of 1000 in the RR field indicates that the counter is decremented every 4 us. A new memory transfer read request is not issued until the counter is zero.

- Assuming no other overhead, the bandwidth consumed by data transfers is equal to approximately (1000 * DWords)/RR MBps.
- A RR value of 1000 results in the DMA channel consuming approximately 1 MBps of memory read and memory write bandwidth.

Request rate control has no effect on descriptor memory read requests and data transfer memory write requests.

- Since completions associated with memory read requests are transformed into memory write requests, the request rate controls both the completion bandwidth as well as the memory write request bandwidth.

1. This includes descriptors in the process of being prefetched. In this case, the DMA waits for the prefetching to complete and then discards the descriptor.

Notes

It is possible to “in-line” request rate control information within a descriptor, using the Request Rate (RR) and Request Rate Update (RRU) fields in a stride descriptor (see section Stride Control Descriptor on page 15-7). This allows control of the request rate depending on the bandwidth of the source and destination devices associated with the DMA transfer.

- For example, software could build a descriptor list that transfers data from a high-bandwidth source device (i.e., x8 Gen 2 root port) to a low-bandwidth destination device (x1 Gen 1 endpoint). The descriptor list would start with a stride descriptor that controls parameters associated with the transfer, including the request rate. This rate would be programmed to throttle DMA requests to prevent congestion at the destination device and the PCI Express topology.
- In this same example, a second descriptor list is chained to the first one. This second list would be programmed to transfer data between the high-bandwidth root port and a high-bandwidth endpoint (i.e., x8 Gen 2). The second list would start with a stride descriptor that controls parameters associated with the transfer, and re-programs the request rate used by the DMA channel. In this example, given that both the source and destination devices have the same bandwidth, the request rate would be disabled by setting the request rate to a value of zero (i.e., the DMA channel would not throttle its requests).

DMA Multicast

A DMA channel may be configured such that the destination address of a descriptor hits a multicast BAR aperture[1] in the upstream port’s PCI-to PCI bridge (i.e., transparent multicast) and/or in the NT function (non-transparent multicast). When this occurs, the posted TLP emitted by the DMA function is transferred to the upstream port’s link (i.e., the port where the DMA function resides), as well as multicasted to the appropriate ports, as dictated by transparent and non-transparent multicast operation.

- In some systems, it may not be desired that the posted TLPs emitted by the DMA be transferred on the upstream port’s link in addition to being multicasted to the appropriate ports in the switch. The Multicast Receive Interpretation (MCRCVINT) register in the DMA function’s configuration space may be programmed to control this behavior. Refer to the definition of this register for details.

Figure 15.12 shows an example of the path taken by a posted TLP emitted by the DMA that falls in the multicast BAR aperture of the upstream port’s PCI-to-PCI bridge function (assuming that the MCRCVINT register is set to 0x0). As shown in the figure, the TLP is transmitted on the upstream port’s link, as well as multicasted to the appropriate downstream ports.

Figure 15.13 shows a similar example, but this time the TLP emitted by the DMA is NT multicasted to ports in other partitions. As shown in the figure, the TLP is transmitted on the upstream port’s link, as well as NT multicasted to ports in other partitions.

Note that the behavior for TLPs that are multicasted differs from unicast transfers generated by the DMA. Unicast TLPs, when claimed by an upstream port’s function (e.g., PCI-to-PCI bridge or NT), only target this function. If not claimed by any of the upstream port functions, the TLP is sent on the upstream link only.

DMA multicast allows the DMA to be configured to read data from a source location, and multicast this data to several destination locations, thereby improving the performance of the transfer operation.

1. Refer to Chapter 17 for details on multicast operation. Multicast operation only applies to posted TLPs.

Notes

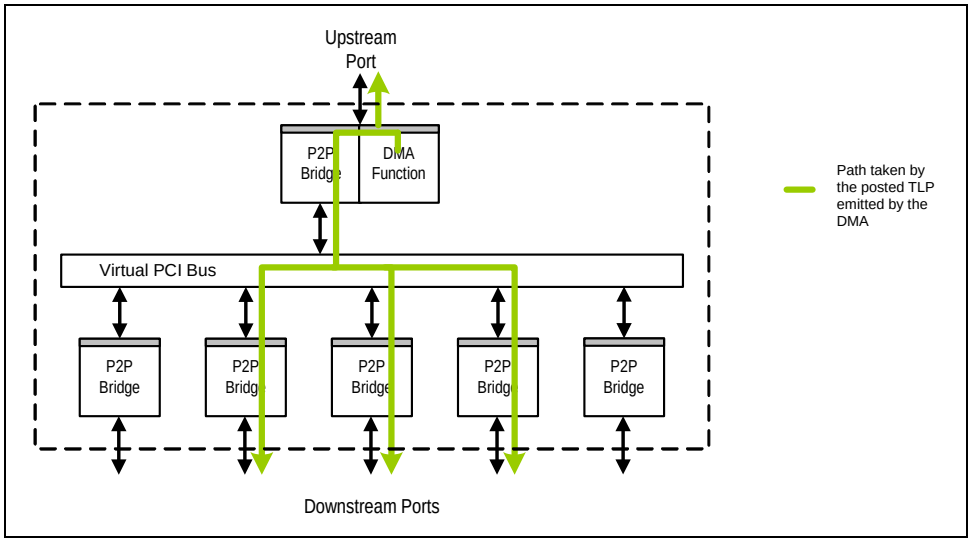

Figure 15.12 Path Taken by a TLP Emitted by the DMA When it is Multicasted[1]

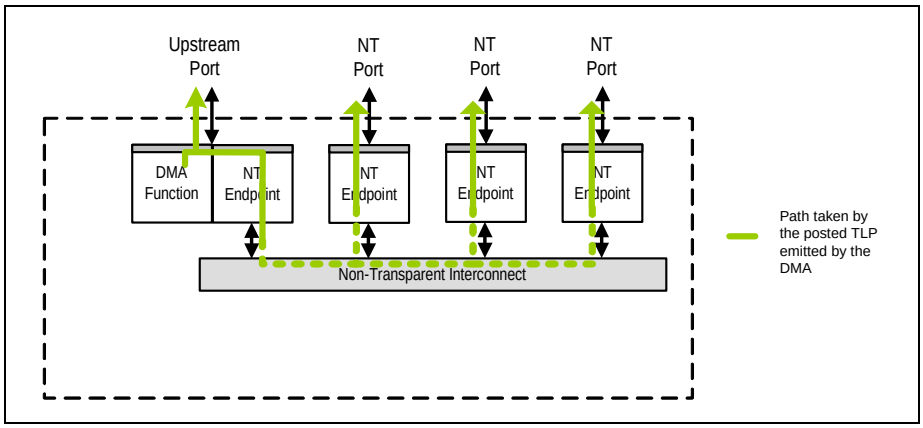

Figure 15.13 Path Taken by a TLP Emitted by the DMA When it is NT Multicasted

Interrupts

The DMA function has the following sources of interrupts.

- DMA channel 0 interrupt
- DMA channel 1 interrupt

DMA channel interrupts are described in section Channel Interrupts on page 15-21.

When the DMA function detects the occurrence of an unmasked interrupt condition, an MSI or interrupt message is generated by the function per the rules in Table 15.9. The removal of the interrupt condition occurs when unmasked status bit(s) causing the interrupt are masked or cleared. An MSI generated by the DMA may be transmitted to a device (i.e., link partner) associated with any port of the switch partition in which the DMA resides.

In addition, an MSI generated by the DMA may be transmitted to a device (i.e., link-partner) associated with any port of another switch partition by configuring the port in which the DMA function resides to include an NT function, and programming the DMA's MSI address to fall into the BAR regions of the NT function.

Furthermore, if the DMA's MSI address falls within an enabled multicast BAR aperture in the partition in which the DMA resides, the MSI may be multicasted or NT multicasted (see Chapter 17).

1. The figure assumes that the Multicast Receive Interpretation (MCRCVINT) register is set to 0x0.

Notes

Unmasked Interrupt	EN bit in MSICAP Register	INTXD bit in PCICMD Register	Action
Asserted	1	X	MSI message generated
	0	0	Assert_INTx message request generated
	0	1	None
Negated	1	X	None
	0	0	Deassert_INTx message request generated
	0	1	None

Table 15.9 Downstream Switch Port Interrupts

When the DMA function is configured to generate INTx messages, the specific INTx used (e.g., INTA, INTB, etc.) depends on the programming of the Interrupt Pin (INTRPIN) register.

Virtual Channel (VC) Support

The DMA function supports per-descriptor traffic-class (TC) control for request TLPs that it transmits, as described in section TLP Attribute and Traffic Class Control on page 15-20. Also, section Virtual Channel Support on page 4-5 describes virtual channel support and TC/VC mapping in the switch ports.

The DMA function does not have a dedicated VC Capability Structure that defines TC/VC mapping. Instead, TC/VC mapping is performed as specified in the VC Capability Structure associated with function 0 of the multi-function port in which the DMA function resides. Depending on the port operating mode, function 0 of the multi-function port may be a PCI-to-PCI bridge or NT function.

- TLPs received by the DMA are checked for TC/VC mapping violations at the ingress port.
- TLPs generated by the DMA are checked for TC/VC mapping at the egress port.

The DMA module does not perform any further TC/VC mapping checks. It is the responsibility of the user to ensure that DMA descriptor traffic class controls map to VC0 within the PES32NT8xG2 switch ports involved in the DMA transfer.

Access Control Services (ACS) Support

The DMA function supports the following ACS checks[1]:

- ACS Peer-to-Peer[2] Request Redirect
- ACS Peer-to-Peer Completion Redirect

ACS is programmed via the ACS Capability Structure in the DMA function's configuration space. The DMA does not support ACS Peer-to-Peer Completion redirect on completions generated by the DMA in response to received requests that target DMA BAR 0. ACS Peer-to-Peer Completion Redirect is advertised as supported in the DMA's ACS Capability Structure, but it does not have any functional effect. Note that completions TLPs generated by the DMA in response to received configuration request TLPs are always routed upstream and therefore ACS has no functional effect on these completions either.

Table 15.10 lists ACS checking and handling performed by the DMA function.

1. Note the DMA does not support Address Translation Services (ATS). As a result, the DMA function does not support ACS Direct Translated P2P.

2. For a port operating in a multi-function upstream port mode (e.g., upstream switch port with DMA function), 'peer-to-peer' implies traffic sent from one of the port functions to another (e.g., from the port's DMA function to the port's PCI-to-PCI bridge function, etc.)

Notes

ACS Check	PCI Express Base Specification[1] Section	Error Reporting Condition	Action Taken
ACS Peer-to-Peer Request Redirect	6.12.1.1	N/A (not an ACS violation)	Offending request is redirected upstream towards root complex.
ACS Peer-to-Peer Completion Redirect			This ACS check has no functional effect in the switch, as described above.

Table 15.10 ACS Checks Performed by the DMA Function

1. Refer to PCI Express Base Specification Revision 2.1.

When an ACS check causes a TLP to be re-directed, the re-direction is implemented such that TLPs emitted by the DMA function that are ACS re-directed follow the ordering rules described in section Packet Ordering on page 4-6. ACS checks are only applicable to certain TLP types. Table 15.11 list the ACS checks supported by the DMA function and the TLP types on which they are applied.

ACS Check	Applicable to the following TLP type(s)
ACS Peer-to-Peer (P2P) Request Re-direct	Peer-to-Peer Request TLPs
ACS P2P Completion Re-direct	N/A (this ACS check has no functional effect in the switch, as described above).

Table 15.11 TLP Types Affected by ACS Checks

As an example of an ACS check performed by the DMA function, consider the case where software enables ACS Peer-to-Peer Request Redirect in the DMA function. This commands the DMA to re-direct upstream (i.e., transmit on the upstream link) all requests that it issues which would have otherwise been logically routed via the upstream port's PCI-to-PCI bridge function or NT function. Figure 15.14 shows an example of a ACS Peer-to-Peer Request Redirect. The green lines mark the requests intended route, and the orange lines the request's re-directed route do to ACS.

Note that all peer-to-peer requests that the DMA issues (e.g.,descriptor read requests, data read requests, descriptor write requests, and data write requests) are all subject to ACS Peer-to-Peer Request Redirect

Notes

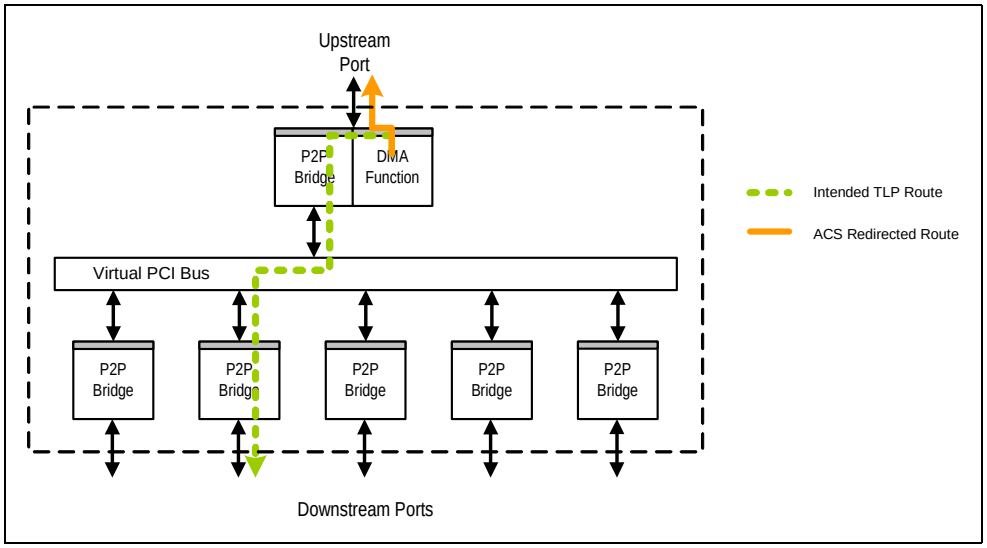

Figure 15.14 Example of ACS Peer-to-Peer Request Redirect Applied by the DMA Function

Refer to PCI Express Base Specification Revision 2.1 for further information on ACS.

Power Management

Refer to Chapter 9, Power Management.

Bus Locking

The DMA function does not support bus locking. Memory read request-locked (MRdLk) TLPs received by the DMA function are treated as unsupported requests and an unsupported request completion with no data (CplLk) is returned. The operation of a switch partition is undefined when bus locking is performed in a partition that contains a DMA function in its upstream port.

ECRC Support

The DMA function supports End-to-End CRC (ECRC) generation and checking. The DMA function performs ECRC error checking and logging when the ECRC Check Enable (ECRCCE) bit is set in the function's AER Control (AERCTL) register and the TLP is received from the upstream port's link.

– ECRC error checking and logging is not performed by the DMA function when it does not receive the TLP from the upstream port's link. In this case, the ECRC error checking and logging is done by the function that received the TLP from the link (e.g., PCI-to-PCI bridge function).

When ECRC checking is enabled, the reception of a TLP without ECRC is not considered an error (i.e., the TLP is processed normally). If the port is operating in a multi-function mode, then ECRC errors are only logged in functions in which ECRC checking is enabled.

ECRC generation is enabled in the DMA function when the ECRC Generation Enable (ECRCGE) bit is set in the function's AER Control (AERCTL) register. If ECRC generation is enabled in the DMA function, then all TLPs originated by the DMA function contain an ECRC. Otherwise, all TLPs originated by the DMA function do not contain ECRC.

Error Handling

This section describes error detection and reporting performed by the DMA function. Errors detected by the DMA function are categorized into:

– PCI Express errors
– DMA channel errors

Notes

PCI Express errors are those specified in the PCI Express Base specification. DMA channel errors are additional proprietary errors associated with the operation of the DMA channels within the DMA function. PCI Express errors are described in section PCI Express Error Handling by the DMA Function on page 15-28.

Internal switch errors (i.e., parity errors, switch time-out, and internal memory errors) are associated with the switch core and not with a specific port function. These errors are not described here. Refer to section Internal Errors on page 4-16 for a detailed description of these errors.

PCI Express Error Handling by the DMA Function

The error handling described in this section corresponds to that outlined in PCI Express Base Specification 2.1. This section describes error conditions detected by the DMA function. This includes physical, data-link, and transaction layer errors detected by the port, as well as application layer errors associated with the DMA function in the port.

The errors described here apply to ports that operate in a mode that includes a DMA function (e.g., upstream switch port with DMA function mode, NT with DMA function mode, etc.) This section focuses specifically on PCI Express errors related to the DMA function[1]. Errors that affect all functions of the port (i.e., non function-specific errors) are noted where appropriate.

PCI Express errors are logged in standard PCI Express registers (e.g., AER capability, PCI Status, Device Status registers, etc.) and signaled via the mechanisms defined in PCI Express Base Specification 2.1 (e.g., correctable or uncorrectable error messages, etc).

– The terms ‘uncorrectable error processing’ and ‘correctable error processing’ refer to the processing described in Section 6.2.5 of PCI Express Base Specification 2.1.

In cases where a PCI Express error can be correlated to the operation of a DMA channel (e.g., a completion timeout when a DMA channel reads a descriptor), or when a PCI Express error causes one or all DMA channels to abort operation, additional error log bits are provided in the corresponding DMA Channel Error Status (DMACxERRSTS) register. This section describes all such cases.

– Errors logged in the DMACxERRSTS register may cause the DMA function to generate an interrupt as described in section Channel Interrupts on page 15-21. Individual errors may be masked from generating an interrupt by programming the appropriate bit(s) in the DMA Channel Error Mask (DMACxERRMSK) register.

For some of the errors listed in this section, detection of the error causes the DMA channel to abort processing of the current descriptor. When this occurs, the actions described in section Aborting a DMA Operation on page 15-18 take place.

Additional Notes Regarding PCI Express Errors

Some of the errors described in this section are non function-specific. Errors that are not function-specific are logged in the corresponding status and logging registers of all functions in the port. Errors that are function-specific are logged in the status & logging register of the affected function. Signaling of non function-specific errors follows the rules in Section 6.2.4 of PCI Express Base Specification 2.1.

For example, depending on the operating mode of the port, the DMA function may share the upstream port of a partition with the PCI-to-PCI bridge function (i.e., the upstream port is a multi-function port). Errors that are non function-specific would be logged and signaled by all functions of the port, per the rules in Section 6.2.4 of PCI Express Base Specification 2.1. Errors that are specific to the DMA function would only be logged in the configuration space registers of the DMA function.

1. Errors associated with the NT function (i.e., non-transparent operation errors) are described in Chapter 14. Errors associated with the PCI-to-PCI bridge function are described in Chapter 10.

Notes

In addition, some of the errors described below are marked as function-specific when the “function claims the TLP”. Some of the errors described below are marked as function-specific when the “function claims the TLP”. A function claims a TLP in the following cases:

- DMA function:
  - Address Routed TLPs: The TLP address falls within the address space range(s) programmed in the DMA’s base address registers (BARs).
  - ID Routed TLPs: The TLP destination ID matches the DMA’s bus/device/function assignment within the PCI Express hierarchy.
  - Implicit Route TLPs: Always.
- PCI-to-PCI Bridge function
  - Refer to section Error Detection and Handling by the PCI-to-PCI Bridge Function on page 10-11.
- NT Endpoint function:
  - Refer to section Error Detection and Handling by the NT Function on page 14-25.

PCI Express errors associated with the DMA function are categorized into physical, data-link, and transaction layer errors.

Physical Layer Errors

All physical layer errors are non-function specific. These errors are described in the error handling section for the PCI-to-PCI bridge function. Refer to section Physical Layer Errors on page 10-11.

Data Link Layer Errors

All data link layer errors are non-function specific. These errors are described in the error handling section for the PCI-to-PCI bridge function. Refer to section Data Link Layer Errors on page 10-12.

Transaction Layer Errors

Table 15.12 lists all the PCI Express error checks performed by the DMA function’s transaction layer as well as the action taken when an error is detected. Per PCI Express Base Specification 2.1, transaction layer errors are ignored in cases where the error is associated with a received packet for which the physical or data-link layers report an error. This prevents error pollution across the layers.

Within the transaction layer, there are error pollution rules that resolve the cases where two or more errors are detected simultaneously. Refer to section Error Pollution on page 15-34 for details on transaction layer error pollution in the DMA function.

Table 15.12 indicates if the detected errors are specific to a DMA channel (i.e., channel-specific errors). Channel-specific errors only affect the DMA channel associated with the error. Non channel-specific errors affect all or none of the DMA channels. The determination is done on an error by error basis as described in the sections below.

For some errors, it is necessary to determine if the function that receives a TLP from the link is an “ultimate receiver” or “intermediate receiver”. The DMA function is always considered an ultimate receiver when the TLP is received directly from the link.

Notes

Error Condition	PCI Express Base Spec[1] Section	Function-Specific Error	Role Based (Advisory) Error Reporting Condition	Channel-Specific Error	Action Taken
Poisoned TLP received: Completion associated with a descriptor read request	2.7.2.2, 6.2.3.2.4.3	Yes	N/A (always non-advisory since the DMA aborts processing of the descriptor when this occurs)	Yes	See section Poisoned TLP Reception on page 15-32. Detected Parity Error bit (PCISTS.DPE) is set. Master Data Parity Error Detected bit (PCISTS.MDPED) is set (if PCICMD.PERRE is set). If the TLP is received from the DMA port's link (i.e., the DMA function is the ultimate receiver): Uncorrectable error processing.
Poisoned TLP received: Completion associated with a data read request			Advisory when the corresponding error is configured as non-fatal in the AERUESV register.	Yes	See section Poisoned TLP Reception on page 15-32. Detected Parity Error bit (PCISTS.DPE) is set. Master Data Parity Error Detected bit (PCISTS.MDPED) is set (if PCICMD.PERRE is set). If the TLP is received from the DMA port's link (i.e., the DMA function is the ultimate receiver): Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing.
Poisoned TLP received: Poisoned memory write request addressing a DMA BAR aperture or Poisoned configuration write request			N/A	No (DMA channels are not affected)	This case is handled as 'reception of a request TLP that is unsupported' (see below).
ECRC check failure[2]	2.7.1	No	N/A (always non-advisory)	No (All DMA channels are affected)	See section ECRC Errors on page 15-32. The received TLP with bad ECRC is discarded. If the TLP is received from the DMA port's link (i.e., the DMA function is the ultimate receiver): Uncorrectable error processing.

Table 15.12 PCI Express Errors Detected by the DMA Function's Transaction Layer (Part 1 of 3)

Notes

Error Condition	PCI Express Base Spec[1] Section	Function-Specific Error	Role Based (Advisory) Error Reporting Condition	Channel-Specific Error	Action Taken
Reception of a request TLP that is unsupported	2.3.1	'Yes' if a function claims the TLP. Else 'No'.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register and the request is non-posted	No (DMA channels are not affected)	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing. For Non-Posted unsupported requests, the function that claims the TLP generates a completion with UR status. If the request is not claimed, then function 0 of the port generates the completion with UR status. See section Reception of a Request TLP That is Unsupported on page 15-33.
Completion time-out (i.e., the DMA detects a completion time-out on an outstanding request)	2.8	Yes	N/A (always non-advisory since the DMA does not re-issue requests after a completion time-out)	Yes	Uncorrectable error processing. See section Completion Timeout on page 15-33
Completer abort issued in response to a received request	2.3.1	Not applicable. The DMA function never issues completions with 'Completer Abort' status.			
Unexpected completion received	2.3.2	'Yes' if a function claims the TLP. Else 'No'.	Advisory when the corresponding error is configured as non-fatal in the AERUESV register	No (DMA channels are not affected)	Non-advisory case: uncorrectable error processing. Advisory case: correctable error processing See section Unexpected Completions on page 15-33.
Completion with UR status received[3]	6.2.3.2.5	Yes	N/A	Yes	The Received Master Abort Status (RMAS) bit in the PCISTS register is set. See section Completion with UR Status Received on page 15-33.
Completion with CA status received[4]	6.2.3.2.5	Yes	N/A	Yes	The Received Target Abort Status (RTAS) bit in the PCISTS register is set. See section Completion with CA Status Received on page 15-34.
Receiver overflow	2.6.1.2	No	N/A (always non-advisory)	No (DMA channels are not affected)	Uncorrectable error processing Offending TLP is nullified.

Table 15.12 PCI Express Errors Detected by the DMA Function's Transaction Layer (Part 2 of 3)

Notes

Error Condition	PCI Express Base Spec[1] Section	Function-Specific Error	Role Based (Advisory) Error Reporting Condition	Channel-Specific Error	Action Taken
Flow control protocol error	2.6.1	Not applicable. The DMA function does not check for any flow control protocol errors.			
Malformed TLP Received	See section Malformed TLP Errors on page 15-34	No	N/A (always non-advisory)	No (DMA channels are not affected)	Uncorrectable error processing Offending TLP is nullified. See section Malformed TLP Errors on page 15-34.
Internal Errors	Section 6.2	No	N/A (always non-advisory)	No (All DMA channels are affected)	Refer to section Internal Errors on page 4-16.

Table 15.12 PCI Express Errors Detected by the DMA Function's Transaction Layer (Part 3 of 3)

1. Refer to PCI Express Base Specification Revision 2.1., March 4, 2009, PCI-SIG.

2. Refer to section ECRC Support on page 15-27.

3. If the completion is unexpected, then it is handled as an unexpected completion received error.

4. If the completion is unexpected, then it is handled as an unexpected completion received error.

The sub-sections below describe in detail the DMA function's error handling for the cases listed in Table 15.12. The error handling described in this section is in addition to the 'action taken' column in Table 15.12.

Poisoned TLP Reception

The DMA handles the reception of a poisoned TLP as follows:

- If the poisoned TLP is a memory write request that falls into one of the DMA's BAR apertures, the TLP is handled as an unsupported request (see section Reception of a Request TLP That is Unsupported on page 15-33).
- If the poisoned TLP is a configuration write request, the TLP is handled as an unsupported request (see section Reception of a Request TLP That is Unsupported on page 15-33).
- If the poisoned TLP is a completion TLP associated with an outstanding descriptor read request by a DMA channel, the Descriptor Poisoned Error (DSCP) bit is set in the corresponding DMA Channel Error Status (DMACxERRSTS) register.
  - The poisoned completion TLP is discarded.
  - The DMA channel aborts descriptor processing.
- If the poisoned TLP is a completion TLP associated with an outstanding data read request by a DMA channel, the Data Poisoned Error (DATP) bit is set in the corresponding DMACxERRSTS register and the behavior is determined by the Poisoned Completion Reception Control (PCRC) field in the DMACxCFG register.

Note: The reception of any other type of poisoned TLP is handled as an unsupported request or unexpected completion.

ECRC Errors

Refer to section ECRC Support on page 15-27 for details on ECRC support in the DMA function.

ECRC errors are non-channel specific. Therefore, when an ECRC error occurs, all DMA channels that are processing a descriptor abort processing.

- Due to a design error, DMA channels that are idle at the time an ECRC error is detected also abort processing (e.g., the Abort bit is set in the DMACxSTS register).

Notes

When an ECRC error is detected, the header of the TLP with ECRC error is not utilized by the DMA channels for internal state computations (e.g., the channel's outstanding byte count is not decremented, etc.) In cases where a received completion TLP has an ECRC error, this results in the DMA channel detecting a completion timeout error later in time.

Reception of a Request TLP That is Unsupported

The DMA function supports the following type of received requests:

- Type 0 configuration read or write requests that target the DMA function
- Memory read or write requests that fall into one of the DMA's BAR apertures

All other received requests are treated as unsupported requests and handled as shown in Table 15.12. Note the following:

- Received vendor defined type 1 messages are silently discarded.

TLPs received by an upstream port that are not claimed by any function in the upstream port are treated as unsupported requests and the error is logged in all functions of the port. The reception of a request TLP that is unsupported has no effect on DMA channel descriptor processing.

Completion Timeout

A completion timeout occurs when a DMA channel fails to receive all completions associated with a read request within the selected completion timeout value.

- The completion timeout value for the DMA function is selected by the Completion Timeout Ranges Supported (CTRS) field in the function's PCI Express Device Capabilities 2 (PCIEDCAP2) register.
  - All channels of the DMA function use the same completion timeout value.
- If a DMA channel fails to receive all completions associated with a read request, then this results in a completion time-out error.

When a completion timeout associated with a descriptor read request is detected, the affected DMA channel aborts descriptor processing and the Descriptor Completion Time-Out Error (DSCCT) bit is set in the corresponding DMACxERRSTS register.

Note: In this device, completion timeout disabling is strongly discouraged, as it can result in DMA malfunction in cases in which outstanding DMA requests are not fully completed (e.g., due to an ECRC error in a completion TLP).

Unexpected Completions

The DMA function treats the following received TLPs as unexpected completions.

- A completion TLP that targets the DMA function (i.e., the requester ID field in the completion TLP matches the DMA's requester ID) but for which there is no outstanding DMA channel request.

The reception of an unexpected completion has no effect on DMA channel descriptor processing.

Completion with UR Status Received

When the DMA function receives an expected completion with unsupported request (UR) status, and the completion is associated with a DMA channel's outstanding descriptor read request, the following actions are taken:

- The affected DMA channel aborts descriptor processing and the Descriptor Unsupported Request (DSCUR) bit is set in the corresponding DMACxERRSTS register.

When the DMA function receives an expected completion with unsupported request (UR) status, and the completion is associated with a DMA channels outstanding data read request, the following actions are taken:

- The affected DMA channel aborts descriptor processing and the Data Unsupported Request (DATUR) bit is set in the corresponding DMACxERRSTS register.

Notes

Completion with CA Status Received

When the DMA function receives an expected completion with completer abort (CA) status, and the completion is associated with a DMA channel's outstanding descriptor read request, the following actions are taken:

– The affected DMA channel aborts descriptor processing and the Descriptor Completer Abort (DSCCA) bit is set in the corresponding DMACxERRSTS register.

When the DMA function receives an expected completion with unsupported request (UR) status, and the completion is associated with a DMA channel's outstanding data read request, the following actions are taken:

– The affected DMA channel aborts descriptor processing and the Data Completer Abort (DATCA) bit is set in the corresponding DMACxERRSTS register.

Malformed TLP Errors

Malformed TLP errors for TLPs received by the port from the link are not function-specific. These formation checks are performed when a port receives a TLP, and if an error is found, the error is logged in all functions of the port. The ingress TLP formation checks performed by the switch ports are described in Table 10.12, Ingress TLP Formation Checks associated with the PCI-to-PCI Bridge Function. The DMA does not perform any egress TLP formation checks.

TLP Header Logging

TLP header logging is subject to the rules outlined in section 6.2 of the PCI Express Base Specification 2.1. The PES32NT8xG2 does not support the recording of multiple headers or the recording of headers for uncorrectable internal errors. When an uncorrectable internal error is reported by AER, a header of all ones is recorded.

The following non function-specific errors require that the offending TLP's header be logged in the DMA function's AER capability structure.

– Reception of a TLP with ECRC error on the upstream port's link.
– Reception of a request that is unsupported on the upstream port's link, when no function in the upstream port claims the TLP.
– Reception of an unexpected completion on the upstream port's link, when no function in the port claims the TLP.
– Reception of a malformed TLP on the upstream port's link.

The following function-specific errors require that the offending TLP's header be logged in the DMA function's AER capability structure. These errors are logged in the DMA function regardless of the port that received the TLP.

– Reception of a request that is unsupported and is claimed by the DMA function.
– Reception of an unexpected completion that is claimed by the DMA function.
– Reception of a poisoned TLP on the upstream port's link that is claimed by the DMA function.
  • When the TLP is not received on the link, header logging is not performed.

Error Pollution

The DMA function supports the AER error pollution rules outlined in section 6.2.3.2.3 of PCI Express Base Specification Revision 2.1. Error pollution rules only apply to errors detected on a received TLP. Errors not associated with a received TLP (e.g., completion timeout error) are logged for each occurrence of the error.

In addition, error pollution rules only apply to errors detected by the AER logic. The error bits in legacy PCI registers (e.g., PCI Status (PCISTS)) are not subject to AER error pollution rules.

– For example, the Detected Parity Error (DPE) bit in the PCISTS register of the DMA function is set when the DMA function receives a poisoned TLP, even if error pollution rules result in a higher priority error (e.g., UR) being logged against the TLP.

Notes

Table 15.13 shows the prioritization of transaction layer errors used by the DMA function. All the errors listed in the table are associated with the reception of a TLP. Errors not detected on the reception of a TLP (e.g., completion timeout) or errors that are not applicable to the DMA function (e.g., completer abort issued in response to a received request) are not shown. Higher priority errors have precedence over lower priority errors. Errors with the same priority are mutually exclusive (the errors can't occur simultaneously).

Error	Associated with Packet Reception	Priority
Internal Error	Depends on error.	6 (highest)
Receiver Overflow	Yes	5
ECRC Check failure	Yes	4
Malformed TLP received	Yes	3
Unsupported Request	Yes	2
Unexpected Completion received	Yes	
Poisoned TLP received	Yes	1 (Lowest)

Table 15.13 Prioritization of Transaction Layer Errors

The prioritization of errors shown in Table 15.13 determines the error that is logged and reported when multiple errors are detected simultaneously for the received TLP. Higher priority errors inhibit the logging and reporting of lower priority errors in AER.

Figure 15.15 shows the decision diagram for the DMA function's error checking and logging on a received TLP taking into account the error pollution rules and priorities.

Notes

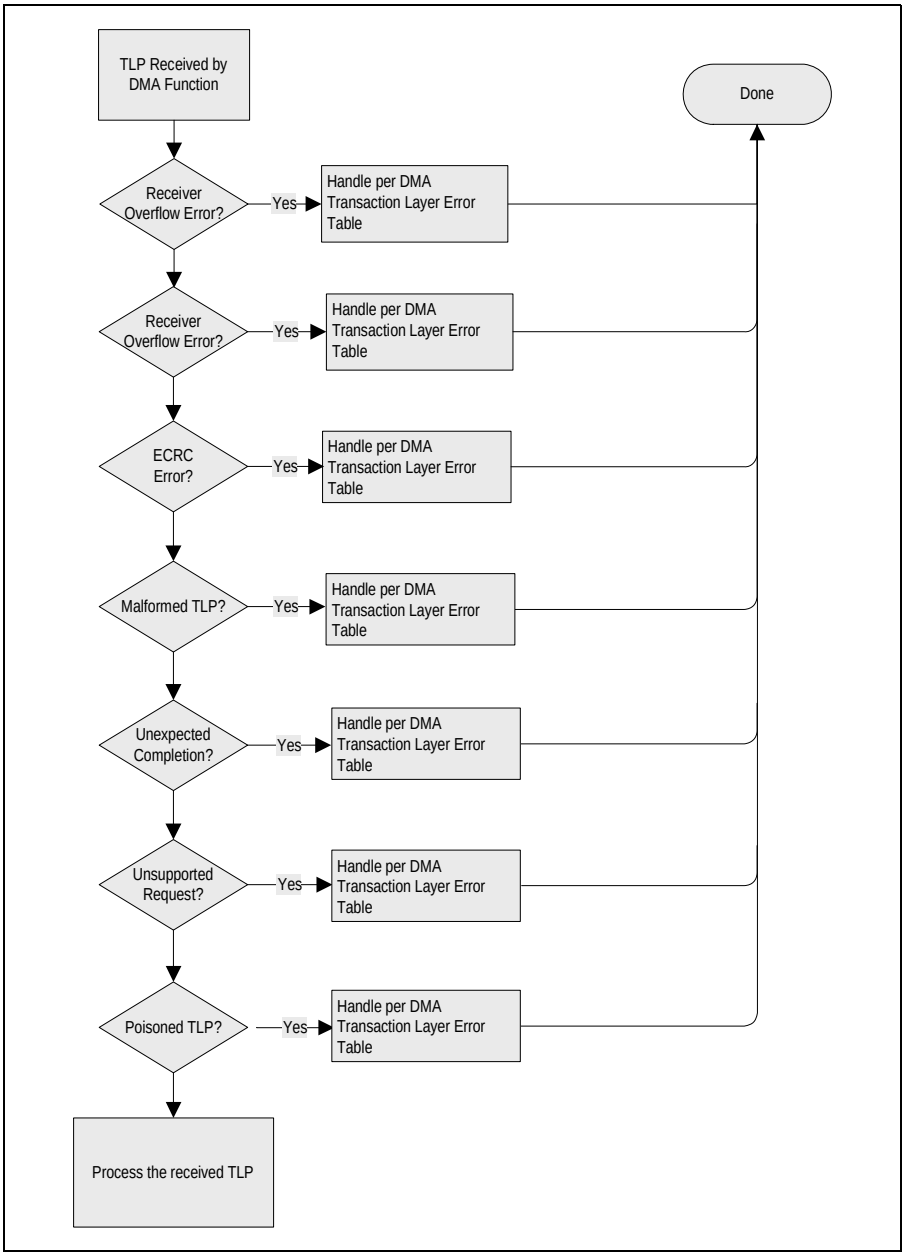

Figure 15.15 DMA Function's Error Checking and Logging on a Received TLP

DMA Limitations and Usage Restrictions

The behavior is undefined in software configurations that result in a race condition between a DMA channel read and write to the same address. For example, if SADDR and DADDR were slightly offset in Figure 15.2, then it might be possible for destination address memory writes to update memory before the same source address is read by the DMA channel.

Switch Events

Notes

Overview

As described in section Switch Events on page 1-16, in a PCI Express switch with multiple partitions a need may exist to signal the occurrence of significant global events to a switch management agent. A need may also exist for communication between roots associated with different partitions as well as for communication between these roots and a management agent. This section describes mechanisms provided by the PES32NT8xG2 to facilitate these forms of communication.

Switch Events

A switch event represents the occurrence of a significant global switch event or an event within a switch partition. Switch event status is logged in global switch control and status registers and may be used to generate an interrupt within selected partitions.

The following events are considered switch events:

- A switch port link going up (i.e., a transition from DL_Down to DL_Up)
- A switch port link going down (i.e., a transition from DL_Up to DL_Down)
- A switch port detecting an AER error
- A fundamental reset in a partition
- A hot reset in a switch partition
- Failover mode change initiated
- Failover mode change completed
- A global signal from a switch partition

Corresponding to each of these events is a status bit in the Switch Event Status (SESTS) register and a mask bit in the Switch Event Mask (SEMSK) register. When a switch event is detected, the event is signaled to all partitions not masked by the Switch Event Partition Mask (SEPMSK) register. Associated with each of these events are further status and mask registers that provide fine grain status and masking control. These are described in section Link Up on page 16-2 through section Port AER Errors on page 16-5.

When a switch event is signaled to a partition, the event may be used to generate an MSI or INTx interrupt within the partition.

- For transparent partitions (i.e., those without an NT endpoint), an interrupt may be reported by the upstream switch port of the partition (i.e., the PCI-to-PCI bridge function in the partition's upstream port). See section Interrupts on page 10-4 for details.
- For partitions consisting only of an NT endpoint port (i.e., a port configured to operate as an NT function or NT with DMA function), an interrupt may be reported by the NT function. See section Interrupts on page 14-20 for details.
- For partitions consisting of both a transparent switch and an NT endpoint (i.e., the upstream port contains a PCI-to-PCI bridge function and an NT function), the event is signaled to both the upstream PCI-to-PCI bridge function and the NT function. An interrupt may be reported by one or both of these functions.

Figure 16.1 shows a simplified representation of the switch event detection and signaling mechanism. As shown, for each type of switch event (e.g., switch port link-up, switch port link-down, failover change initiated, etc.) there is logic that detects the occurrence of that event. The SESTS register logs the occurrence of the event, the SEMSK register controls which events are signaled to the partitions, and the SEPMSK

Notes

register controls which partitions are notified of the occurrence of an event. As mentioned above, each partition's upstream port functions (i.e., PCI-to-PCI bridge and/or NT) may be configured to generate an interrupt to the system when an event is signaled to the partition.

- When a switch event is signaled to multiple partitions and the corresponding interrupts are generated, interrupt handling software running in the root-complex of each partition can determine the cause of the interrupt by probing the interrupt status register in the partition's upstream port function (i.e., P2PINTSTS register in the PCI-to-PCI bridge function or NTINTSTS register in the NT function). From probing this register, the interrupt handler can determine if the interrupt was caused by a switch event. Furthermore, the exact event that caused the interrupt can be determined by probing the SESTS register.
- In order to re-arm the interrupt mechanism due to switch events, such software needs to clear the Switch Event (SEVENT) bit in the P2PINTSTS or NTINTSTS register. In addition, it is recommended that a switch manager device in charge of configuring the event signaling mechanism also be signaled of the occurrence of the event (e.g., by connecting the switch manager to a switch partition's upstream port), so that the switch manager can re-arm the event signaling mechanism by clearing the appropriate status bits in the event status registers (e.g., SELINKUPSTS, SELINKDNSTS, SEFRSTSTS, etc., described in the following sections).

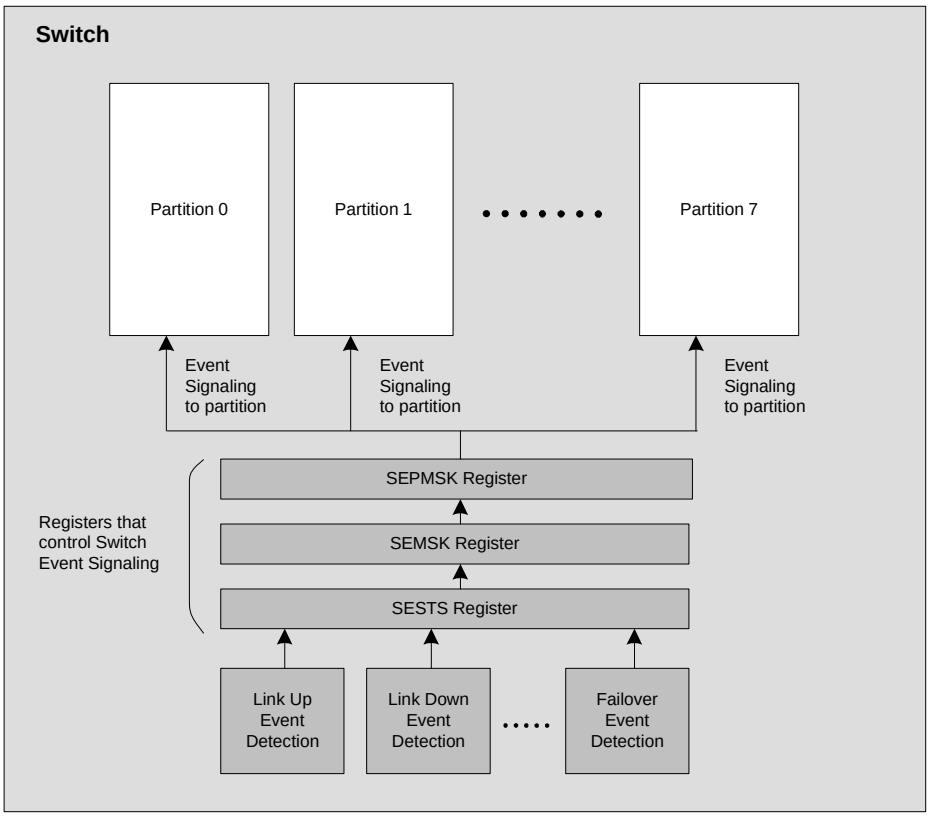

Figure 16.1 Switch Event Detection and Signaling Mechanism

Link Up

A link up event occurs when a port's data link status transitions from DL_Down to DL_Up state. Refer to section Link States on page 7-9 for a details on the conditions for which the data link reports DL_Down and DL_Up status.

Associated with each port of the switch is a status bit in the Switch Event Link Up Status (SELINKUPSTS) register. A bit in the status register is set when a link up event occurs on the corresponding port.

Notes

Associated with each status bit in the SELINKUPSTS register is a mask bit in the Switch Event Link Up Mask (SELINKUPMSK) register. When an unmasked status bit is set in the SELINKUPSTS register, the Link Up (LNKUP) status bit is set in the Switch Event Status (SESTS) register.

Link Down

A link down event occurs when a port's data link status transitions from DL_Up to DL_Down. Refer to section Link States on page 7-9 for a details on the conditions for which the data link reports DL_Down and DL_Up status.

Associated with each port of the switch is a status bit in the Switch Event Link Down Status (SELINKDNSTS) register. A bit in the status register is set when a link down event occurs on the corresponding port.

Associated with each status bit in the SELINKDNSTS register is a mask bit in the Switch Event Link Down Mask (SELINKDNMSK) register. When an unmasked status bit is set in the SELINKUPSTS register, the Link Down (LNKDN) status bit is set in the Switch Event Status (SESTS) register.

Fundamental Reset

A fundamental reset event occurs within a partition when a fundamental reset occurs within a partition as described in section Partition Fundamental Reset on page 3-10. Associated with each partition of the switch is a status bit in the Switch Event Fundamental Reset Status (SEFRSTSTS) register. A bit in the status register is set when a fundamental reset is detected in the corresponding partition. Associated with each status bit in the SEFRSTSTS register is a mask bit in the Switch Event Fundamental Reset Mask (SEFRSTMSK) register. When an unmasked status bit is set in the SEFRSTSTS register, the Fundamental Reset (FRST) status bit is set in the Switch Event Status (SESTS) register.

Hot Reset

A hot reset event occurs within a partition when a partition hot reset is initiated as described in section Partition Hot Reset on page 3-11. Associated with each partition of the switch is a status bit in the Switch Event Hot Reset Status (SEHRSTSTS) register. A bit in the status register is set when a hot reset is detected in the corresponding partition. Associated with each status bit in the SEHRSTSTS register is a mask bit in the Switch Event Hot Reset Mask (SEHRSTMSK) register. When an unmasked status bit is set in the SEHRSTSTS register, the Hot Reset (HRST) status bit is set in the Switch Event Status (SESTS) register.

Failover

The switch reconfiguration caused by a failover event may take some time to complete. Thus, associated with each failover capability are two events. A failover mode change initiated event occurs when a failover event is triggered by a failover capability and a failover mode change completed event occurs when switch reconfiguration resulting from the failover event completes.

- The Failover Mode Change Initiated (FMCI) bit is set in the corresponding Failover Capability Status (FCAPxSTS) register when a failover mode change is initiated.
- The Failover Mode Change Completed (FMCC) bit is set in the corresponding Failover Capability Status (FCAPxSTS) register when a failover mode change completes.

Failover event status bits are located in the corresponding FCAPxSTS register. Associated with each failover event is a mask bit in the Switch Event Failover Mask (SEFOVRMSK) register. When a status bit is set in a FCAPxSTS register that is not masked by a corresponding bit in the SEFOVRMSK register, then the FOVR bit is set in the Switch Event Status (SESTS) register.

Notes

Global Signals

Global signals allow an agent in a switch partition to signal a switch event. This mechanism provides a primitive form of communication that allows an agent in a switch partitition to communicate with agents in other partitions. Such communication may be used to coordinate actions such as dynamic partition re-configuration by a switch management agent.

- A global signal event may be signaled by an NT function by writing a one to the Global Signal (GSIGNAL) bit in the NT Endpoint Global Signal (NTGSIGNAL) register.
- A global signal event may be signaled by an upstream PCI-to-PCI bridge by writing a one to the Global Signal (GSIGNAL) bit in the PCI-to-PCI Bridge Global Signal (P2PGSIGNAL) register.

Associated with each partition is a global signal status bit in the Global Switch Event Signal Status (SEGSIGSTS) register. When a global signal is issued by an agent in a switch partition, the corresponding partition bit in this register is set.

Associated with each status bit in the SEGSIGSTS register is a mask bit in the Switch Event Global Signal Mask (SEGSIGMSK) register. When an unmasked status bit is set in the SEGSIGSTS register, the Global Signal (GSIGNAL) status bit is set in the Switch Event Status (SESTS) register.

Figure 16.2 shows the global signaling mechanism. As shown, any partition can issue a global signal, which is logged in a corresponding bit in the SEGSIGSTS register. This feeds into the registers that control event signaling (i.e., SESTS, SEMSK, SEPMSK), thereby allowing the signaling of a switch event in any partition.

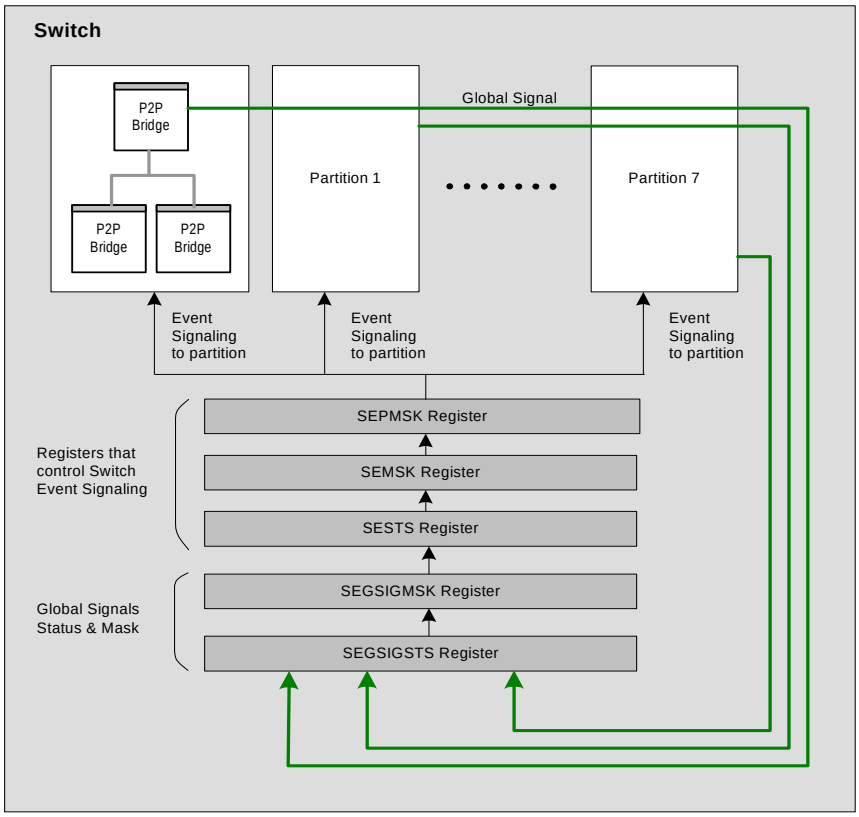

Figure 16.2 Global Signaling Mechanism

Registers associated with global signal events are located in the NT function and the PCI-to-PCI bridge function of a partition's upstream port. These registers provide a complementary communication mechanism for the global signals mechanism. The manner and protocol used for this communication is beyond the scope of this specification. For example, an agent that issues a global signal event may use these registers to store information regarding the reason behind the issuing of the global signal event.

Notes

Within each NT function and upstream PCI-to-PCI bridge function is a general 32-bit read-write register that may be used to pass arbitrary data between an agent associated with a partition and agents in other partitions.

- The NT Endpoint Signal Data (NTSDATA) register is this register in an NT function.
- The PCI-to-PCI Bridge Signal Data (P2PSDATA) register is this register in the upstream PCI-to-PCI bridge function.

The agents that receive notification of the global signal event may read from the NTSDATA and/or P2PSDATA register of the agent that issued the global signal event by accessing the device's global address space (see Chapter 19).

Port AER Errors

It is possible to signal the occurrence of an AER error in any port as a switch event. Each port contains an internal non-software visible Port AER Status register (PAERSTS). The PAERSTS register provides a combined status for the AER correctable and uncorrectable errors of all functions (e.g., PCI-to-PCI bridge, NT, and DMA) in the port. This information may be used to generate a switch event.

Associated witht the PAERSTS register is a software-visible PAERMSK register. The PAERMSK register determines which bits in the PAERSTS register are taken into account for generating a switch event. The Switch Event Status (SESTS) register contains a bit per port (i.e., PxAER) that indicates the occurrence of an unmasked port AER error. The PxAER bit is set when at least one unmasked bit is set in the corresponding port's PAERSTS register

Notes

Multicast

Notes

Overview

The PES32NT8xG2 implements multicast within switch partitions as defined by the PCI Express Base Specification 2.1. The term *transparent multicast* is used to refer to this type of multicast operation. In addition, the switch supports *non-transparent multicast*, using a proprietary implementation. This allows TLPs received by the NT endpoint to be multicasted to ports in other switch partitions. Transparent and non-transparent multicast may operate concurrently in a partition.

The multicast capability enables a single TLP to be forwarded to multiple destinations. The destinations to which a multicast TLP is forwarded is referred to as a multicast group. A multicast group may contain zero or more destinations.

The PES32NT8xG2 supports up to 64 multicast groups -- this is the maximum allowed by the PCI Express standard. Non-transparent (NT) multicast operation is limited to four groups. A function need not be a member of a multicast group in order to generate a multicast TLP that is forwarded to a multicast group. For example, any endpoint or root may generate a multicast TLP by transmitting a posted TLP with an address that maps to a multicast group.

Multicast is compatible with legacy PCI Express roots and endpoints. This chapter describes transparent and non-transparent multicast operation.

Transparent Multicast Operation

This section describes the PES32NT8xG2 transparent multicast. Transparent multicast adheres to the Multicast functionality described in PCI Express Base Specification 2.1. Multicast operation is contingent upon the Memory Access Enable (MAE) and Bus Master Enable (BME) controls being enabled in the PCICMD register of the PCI-to-PCI bridge functions that receive or transmit the multicast packet.

Addressing and Routing

Multicast addressing and routing may be partitioned into the task of determining that a TLP is a multicast TLP, routing a multicast TLP to functions (e.g., PCI-to-PCI bridges associated with egress ports), and multicast egress processing performed at each function. These tasks are described in the following sections.

Multicast TLP Determination

The determination of whether or not a TLP is a multicast TLP is made by functions that receive the TLP. All functions associated with a PES32NT8xG2 switch partition are expected to have identical multicast routing configuration[1]. Thus, multicast TLP determination may be made using register values associated with the capability structure of any function in the partition. Modification of multicast routing fields requires that multicast traffic be quiesced.

1. The switch allows boot-time programming (e.g., via EEPROM) of the capability structure link list within each function's configuration space. In order to use multicast within a switch partition, the Multicast Capability Structure must be linked in the capabilities list in the configuration space of all functions in the partition. The DMA function is excluded as it does not contain a Multicast Capability Structure.

Notes

The following multicast register fields must be configured to the same value in all functions associated with a switch partition. Violating this requirement results in undefined behavior on receipt of a multicast TLP. Non-multicast TLPs are not affected.

- MCCTL Register
  - NUMGROUP
  - MEN
- MCBARL Register
  - INDEXPOS
  - MCBARL
- MCBARH Register
  - MCBARH

Unless otherwise noted, TLP processing associated with a multicast TLP is the same as that for any other TLP. For example, malformed checks are the same, poison bit processing is the same, ECRC checking and error reporting is the same, and so on. When the Multicast Enable (MEN) bit is cleared in the Multicast Control (MCCTL) register, multicast is disabled and no TLP received on the link associated with that port is a multicast TLP.

A TLP determined not to be a multicast TLP is routed using traditional unicast PCI Express routing rules. Thus, unroutable “multicast TLPs” are handled in the same manner as any other unroutable TLP.

Only posted memory write TLPs and address routed message TLPs can be multicast TLPs. The primary determinant of whether or not a memory write or address routed message TLP is a multicast TLP is its address and the address associated with multicast address regions. A multicast address region may overlap a non-multicast address region.

Multicast TLPs that target a multicast address region are routed to all multicast group members while other TLPs, such as non-posted reads, may be routed to only one, possibly different, destination. Multicast TLPs are posted TLPs and have the same ordering requirements as other posted TLPs. There are no new multicast TLP ordering rules.

The maximum number of multicast groups supported in an implementation is contained in the Max Multicast Groups (MAXGROUP) field in the Multicast Capability (MCCAP) register. The number of multicast groups that are actually enabled is determined by the value in the Number of Multicast Groups (NUMGROUP) field in the Multicast Control (MCCTL) register.

As illustrated in Figure 17.1, multicast TLP group membership is determined by address. Associated with each multicast group is an address region. Posted memory write and address routed message TLPs whose address is equal to that associated with a multicast group when the Multicast Enable (MEN) bit is set, are defined to be multicast TLPs associated with that group.

Notes

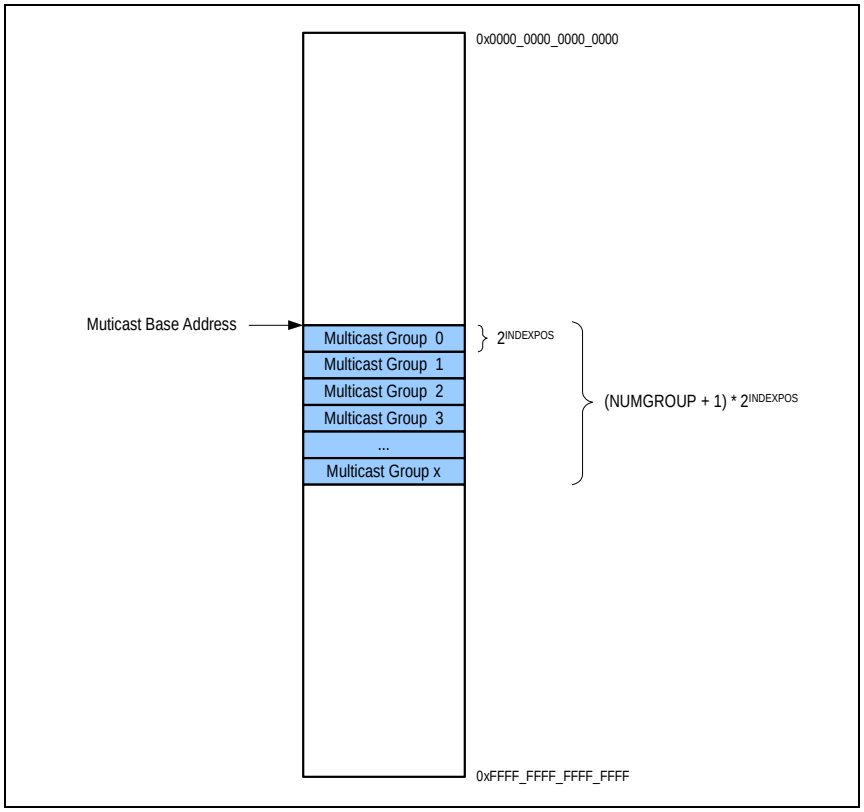

Figure 17.1 Multicast Group Address Ranges

The multicast address region associated with a TLP is determined as follows.

– Multicast group address regions are laid out contiguously in memory from low to high starting with multicast group zero. The number of regions is determined by the value of the NUMGROUP field in the MCCTL register.
  • There are no address regions allocated for multicast group numbers that are greater than that enabled by the NUMGROUP field in the MCCTL register.
– The size of each multicast group address region is determined by the value of the Index Position (INDEXPOS) field in the Multicast Base Address Low (MCBARL) register. The size of each multicast group region is equal to $2^{INDEXPOS}$.
– The starting address of the region associated multicast group zero is equal to the multicast base address defined by the Multicast Base Address Low (MCBARL) field in the MCBARL register and the Multicast Base Address High (MCBARH) field in the Multicast Base Address High (MCBARH) register.
  • The multicast base address is a 64-bit quantity and may start anywhere in memory.
– In general, the starting address of the region associated with multicast group *n* is equal to the multicast base address plus the quantity ($n * 2^{INDEXPOS}$).

Notes

Since bits in the multicast base address that correspond to the multicast group number or are less than the multicast index position (i.e., INDEXPOS) must be zero, the multicast group ID associated with a TLP may be determined as shown in Figure 17.2.

– For the purpose of multicast TLP determination, address bits in the TLP address less than the multicast index position and bits associated with the multicast group ID are zeroed. This address is then compared to the multicast base address. If the two are not equal, then the TLP is not a multicast TLP.
– If the two addresses are equal, then the multicast group ID is extracted from the address bits that correspond to enabled multicast groups. The number of bits is equal to $\lceil \log_2(\text{NUMGROUP} + 1) \rceil$ and start with the address bit corresponding to the value of INDEXPOS.
– If the multicast group ID is greater than NUMGROUP, then the TLP is not a multicast TLP. Otherwise, the TLP is a multicast TLP.

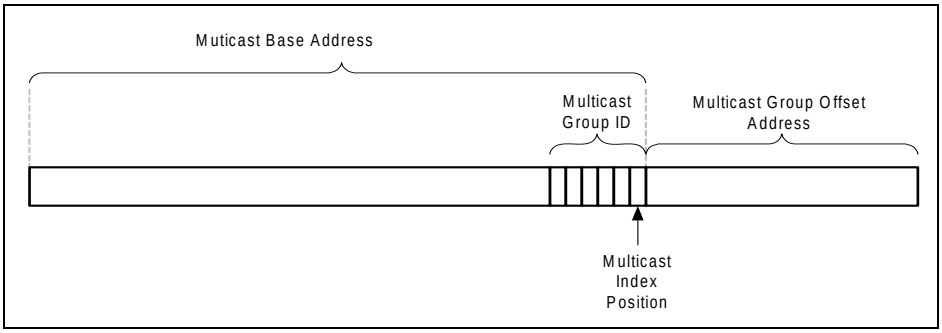

Figure 17.2 Multicast Group Address Region Determination

Once a TLP has been determined to be a multicast TLP and the multicast group ID has been determined, the following error checks are performed.

– If the multicast TLP fails the source validation ACS check, then it is handled as specified by ACS. No other ACS checks are performed on multicast TLPs.
– Associated with each multicast group is a “block all” bit. If the block all bit corresponding to the multicast group ID associated with a received multicast TLP is set in the ingress port, then the multicast TLP is treated as a blocked multicast TLP. The block all bits are contained in the ingress port’s Multicast Block All (MCBLKALL) fields of the Multicast Block All Low (MCBLKALLL) and Multicast Block All High (MCBLKALLH) registers.
– Associated with each multicast group is a “block untranslated” bit. If the block untranslated bit corresponding to the multicast group ID associated with a received multicast TLP is set in the ingress port and the TLP is untranslated, as determined by the Address Type (AT) field in the TLP header, then the multicast TLP is treated as a blocked multicast TLP. The block untranslated bits are contained in the ingress port’s Multicast Block Untranslated (MCBLKUT) fields of the Multicast Block Untranslated Low (MCBLKUTL) and Multicast Block Untranslated High (MCBLKUTH) registers.
– A blocked multicast TLP is treated in the following manner.
  • The TLP is dropped and flow control credits are returned.
  • The error is reported to AER as a MC Blocked TLP error and the header is logged.
  • In an upstream port, the Signaled Target Abort (STAS) bit is set in the PCI Status (PCISTS) register.
  • In a downstream switch port, the Signaled Target Abort (STAS) bit is set in the Secondary Status (SECSTS) register.
– Note that the “block all” and “block untranslated” functions are performed at the ingress port on which the multicast TLP was received.

A received multicast TLP without errors is forwarded to egress ports as described in the next section.

Notes

Multicast TLP Routing

A multicast TLP received without error by a function is forwarded as described in this section. Traditional unicast routing rules do not apply to multicast TLPs. Unlike unicast routing rules that depend on whether the TLP was received on the primary or secondary side of a PCI-to-PCI bridge and are thus different for upstream and downstream switch ports, multicast TLP routing is symmetric. The same multicast routing rules apply to all functions.

A multicast TLP received by a function is forwarded to the virtual PCI bus of the associated switch partition. All functions connected to the virtual PCI bus examine the multicast group ID associated with the multicast TLP and perform the following actions.

- The function on which the multicast TLP was received ignores the multicast TLP.
- If the multicast enable (MCEN) bit is cleared, then the function ignores the multicast TLP.
- Associated each function is a multicast receive vector that contains a bit corresponding to each multicast group. If the MCEN bit is set and the bit corresponding to the multicast group ID associated with the multicast TLP is set in the multicast receive vector, then the multicast TLP is accepted by the function. The multicast receive vector is contained in the Multicast Receive (MCRCV) fields of the Multicast Receive Low (MCRCVL) and Multicast Receive High (MCRCVH) registers.
- A function that accepts a multicast TLP forwards the TLP after multicast egress processing is performed. For a PCI-to-PCI bridge, forwarding a TLP means transmitting the TLP on the link associated with the switch port corresponding to the PCI-to-PCI bridge.
- If no function accepts a multicast TLP, then the TLP is silently discarded. This is not an error.

Note: This section described multicast TLP routing from a functional perspective to aid in understanding. This functional definition does not represent the actual multicast routing implementation in the switch.

Multicast Egress Processing

Each PES32NT8xG2 PCI-to-PCI bridge function implements multicast overlay processing. When the Overlay Size (OVRSIZE) field in the Multicast Overlay Base Address Low (MCOVRBARL) register is set to less than six, multicast overlay processing is disabled and multicast TLPs are forwarded without modification. When the OVRSIZE field value is six or greater, multicast overlay processing is performed on all multicast TLPs accepted by the function as described below.

- Address bits in the accepted multicast TLP with bit positions greater than or equal to OVRSIZE are replaced by the corresponding address bits in the multicast overlay base address.
  - The multicast overlay base address is contained in the Multicast Overlay BAR Low (MCBARL) field in the Multicast Overlay Base Address Low (MCOVRBARL) register and the Multicast Overlay BAR High (MCBARH) field in the Multicast Overlay Base Address High (MCOVRBARH) register.
- Address bits less than OVRSIZE are not modified.
- As a result of multicast overlay processing, a multicast TLP with an original address above 4 GB may be translated into a multicast TLP with address below 4 GB, and vice-versa. Thus, address translation may change the size of a multicast TLP header (e.g., from 4 DWords to 3 DWords).
- Multicast overlay processing is performed independently on all functions. Therefore, it is possible to enable this capability in some functions and not others. The overlay base address associated with different functions will likely have different values. This capability is available on both upstream and downstream switch ports and operates in the same manner regardless of port type.

Notes

A side-effect of modifying the address due to multicast overlay processing is that the ECRC associated with the original TLP may not be correct for the new modified TLP. The PES32NT8xG2 supports ECRC regeneration for multicast overlay.[1] Therefore, functions perform the following ECRC processing.

- If multicast overlay processing is disabled, then no ECRC processing is performed as part of multicast egress processing.
- If a multicast TLP does not contain an ECRC, then no ECRC processing is performed as part of multicast egress processing.
- If a multicast TLP contains an ECRC and multicast overlay processing is enabled, then the following actions are performed.

  The ECRC of the original multicast TLP is checked while simultaneously the ECRC for the new modified TLP is computed or "regenerated." This is implemented in the same pipeline stage such that there is virtually no possibility of silent data corruption (e.g., a TLP bit flip that does not result in a computed ECRC error in the original or regenerated ECRC).

  If no error is detected in the ECRC associated with the original TLP, then the modified TLP is forwarded with the regenerated ECRC.

  If an error is detected in the ECRC associated with the original TLP, then the modified TLP is forwarded with inverted regenerated ECRC (i.e., the computed ECRC of the modified TLP is inverted).

  No errors are reported due to multicast egress processing.

Usage Restrictions

The switch does not support the following transparent multicast transfers. All other transfers are allowed.

- A multicast TLP received on a downstream port that is multicasted to the partition's upstream port must not contain an address that maps into the upstream port's NT function or DMA function BAR apertures.
- A TLP that crosses partitions via the non-transparent bridge (NTB) must not fall into a multicast window in the destination partition. The NT translation in the NT function must be programmed to prevent this scenario. Breaking this rule produces undefined results.

Non-Transparent Multicast Operation

This section describes the switch's non-transparent (NT) multicast. NT multicast allows TLPs received by a port to be multicasted to one or more output ports located in other switch partitions.

NT multicast requires the presence of an NT function in the upstream port of the partition that receives the TLP to be multicasted. A TLP received on any port of such partition may be NT multicasted.

- NT Multicast operation requires that the Memory Access Enable (MAE) control be enabled in the PCICMD register of the NT function that receives the multicast packet.

NT multicast is based on a proprietary implementation that resembles transparent multicast. In particular, the NT function contains a Multicast Capability Structure[2] as defined in the PCI Express Base Specification. This capability structure allows configuration of a multicast range, segmentation of the range into multicast windows, and association of these windows with multicast groups. Each group is associated with a set of ports, located in other switch partitions, to which the multicast TLP is delivered.

Note: NT multicast is only supported for groups 0 to 3.

1. Note that ECRC regeneration is not dependent on the setting of the ECRC Checking Enable (ECRCCE) or ECRC Generation Enable (ECRCGE) bits in the AERCTL register of any of the port functions.

2. The switch allows boot-time programming (e.g., via EEPROM) of the capability structure link list within each function's configuration space. In order to use multicast within a switch partition, the Multicast Capability Structure must be linked in the capabilities list in the configuration space of all functions in the partition. The DMA function is excluded as it does not contain a Multicast Capability Structure.

Notes

When the upstream port operates in a mode that contains an NT function but not a PCI-to-PCI bridge function (e.g., NT function mode, or NT with DMA function mode), NT multicast allows TLPs received by the NT function to be multicasted to ports in other partitions.

When the upstream port operates in a mode that contains an NT and PCI-to-PCI bridge function, NT multicast co-exists with transparent multicast. For example, in a switch configuration such as the one shown in Figure 17.3, transparent multicast is configured to multicast TLPs to ports within the partition, and NT multicast configured to multicast TLPs to ports in other partitions. TLPs received by any port in the partition are processed as both transparent multicast and NT multicast TLPs.

Note: A received TLP that does not fall into groups 0 to 3 is not NT multicasted.

Processing of NT multicast TLPs may be divided into the task of determining that a TLP is an NT multicast TLP, sending the multicast TLP to egress ports, and multicast egress processing performed at each egress port. These tasks are described in the following sub-sections. Prior to this, NT multicast configuration is described.

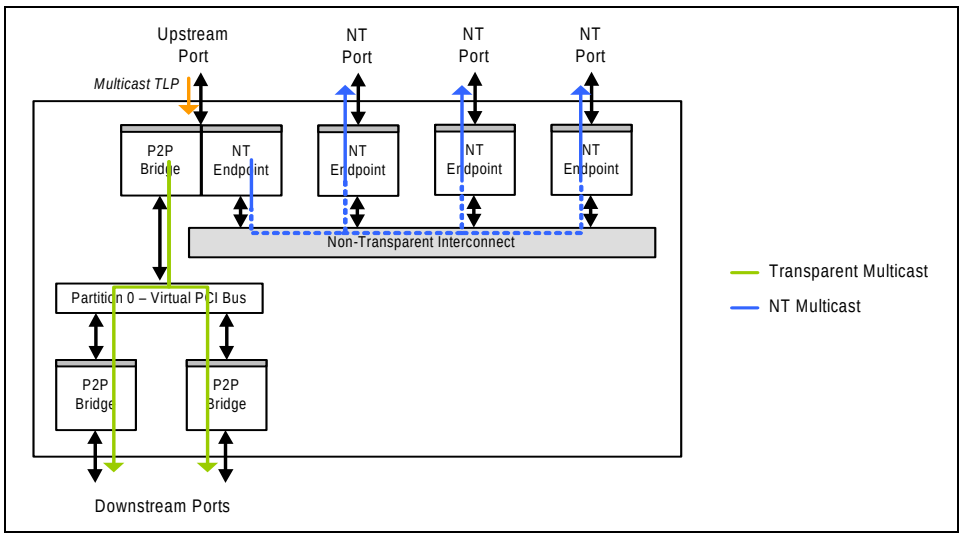

Figure 17.3 Transparent and Non-Transparent Multicast

NT Multicast Configuration

The following register fields in the Multicast Capability Structure of the NT function must be configured to the same value as the Multicast Capability Structure of all other functions[1] associated with a switch partition. Violating this requirement results in undefined behavior on receipt of a multicast TLP. Non-multicast TLPs are not affected.

- MCCTL Register
  - NUMGROUP
  - MEN
- MCBARL Register
  - INDEXPOS
  - MCBARL
- MCBARH Register
  - MCBARH

For a switch partition with a transparent switch and an NT function such as the one shown in Figure 17.3, the above requirement implies that when multicast is enabled, both transparent and NT multicast are enabled. Further, the multicast window location and size, and the number of multicast groups must be

1. The DMA function is excluded as it does not contain a Multicast Capability Structure.

Notes

configured identically for transparent and NT multicast. But transparent and NT multicast configurations differ in their group/port associations. Specifically, transparent multicast groups are associated with ports within the switch partition, and NT multicast groups are associated with ports in other switch partitions.

- NT multicast group to port association is described in section NT Multicast TLP Routing on page 17-8.
- Transparent multicast group to port association is described in section Multicast TLP Routing on page 17-5.

When NT multicast is enabled in conjunction with transparent multicast, the number of multicast groups for both transparent and NT multicast may be set to any value between 0 and the value of the MAXGROUP field in the Multicast Capability (MCCAP) register. NT multicast is only performed for TLPs that fall into groups 0 to 3. TLPs that fall in other groups are not NT multicasted by the receiving port.

The MAXGROUP field in the NT function's MCCAP register has a default value of 0x3, to indicate support for groups 0 to 3. This field has RWL type and may be modified to a value between 0 to 63. Modifying this field allows the NT function to advertise support for up to 64 groups, which in turn allows system software to program the NUMGROUP field to a value between 0 and 63. Thus, in a partition in which multicast and NT multicast are enabled, it is possible to configure up to 64 multicast groups. TLPs received by any port in the partition that fall between groups 0 to 3 may be multicasted and NT multicasted. TLPs received by any port in the partition that fall between groups 4 to 63 can only be multicasted within the partition (i.e., transparent multicast).

NT Multicast TLP Determination

The first step in NT multicast processing requires that the port that receives the TLP from the link determine if the TLP is to be NT multicasted. NT Multicast TLP determination is identical to transparent multicast TLP determination as described in section Multicast TLP Determination on page 17-1, except for the following requirements.

- An NT multicast address region must not overlap a non-multicast NT address region. That is, the NT multicast address region must not overlap with the BAR apertures of the NT function. Overlapping these ranges produces undefined results.

  Note that since all functions of a switch partition must be configured with the same multicast address region, the NT function's multicast region overlaps with the multicast region of the PCI-to-PCI bridge functions (if any) in the same partition. Furthermore, overlapping the multicast address regions of PCI-to-PCI bridge functions with the non-multicast regions of these functions (e.g., bridge's base and limit registers) is allowed. As a result, the NT function's multicast region may overlap with the unicast region of one or more PCI-to-PCI bridge function's in the partition.

Once a TLP has been determined to be a multicast TLP and the multicast group ID has been resolved, the following checks are performed.

- If the multicast TLP is received on a downstream port and fails the source validation ACS check, then it is handled as specified by ACS. No other ACS checks are performed on multicast TLPs.
- The Multicast Receive Low (MCRCVL) and Multicast Receive High (MCRCVH) registers associated with the NT function in the partition are checked. Only multicast TLPs whose group ID corresponds to a set bit in the multicast receive registers are NT multicasted.
  - If the multicast TLP's group ID does not correspond to a set bit in the multicast receive registers, the TLP is ignored by the receiving function and no further NT multicast processing is done by this function.

Note that the "block all" and "block untranslated" bits are not checked during NT multicast processing. A received NT multicast TLP without errors is forwarded to egress ports as described in the next section.

NT Multicast TLP Routing

An NT multicast TLP received without error is forwarded as described in this section. Unicast non-transparent routing rules (see Chapter 14) do not apply to NT multicast TLPs. When a port receives an TLP that is determined to be NT multicasted, it sends the TLP to zero or more switch ports. These switch port(s)

Notes

perform NT multicast egress processing (see section NT Multicast Egress Processing on page 17-9) and transmit the TLP on their data-link. The determination of which ports transmit the TLP is based on the following:

- The received TLP's multicast group ID.
- The programming of the NT Multicast Group x Port Association (NTMCG[3:0]PA) registers, located in the configuration space of the NT function in the partition.
- The setting of the NT Multicast Transmit Enable (NTMCTEN) bit in NT Multicast Control (NTMCC) register of the switch egress port(s).

On reception of a multicast TLP, the receiving port computes the multicast group ID of the TLP, checks the appropriate NT Multicast Group Port Association register to determine the egress port(s) associated with that group ID, and sends the TLPs to these egress ports.

Associated with each egress port in the switch is an NT Multicast Control (NTMCC) register. When the NT Multicast Transmit Enable (NTMCTEN) bit is set in this register, the egress port is enabled for transmission of NT multicast TLPs. Otherwise, the port does not transmit NT multicast TLPs, even if the port is part of an NT multicast group.

For egress ports enabled for NT multicast transmission, the NT multicast TLP logically emerges at the data-link layer of these ports. Therefore, none of the legacy PCI control bits have effect on the TLP (e.g., the Memory Access Enable (MAE) or Bus Master Enable (BME) bits at the egress port's PCICMD register have no effect on NT multicast TLPs). The only processing performed by an egress port on NT multicast TLPs is the processing described in section NT Multicast Egress Processing on page 17-9.

The NT Multicast Group x Port Association (NTMCG[3:0]PA) registers create an association between the multicast group ID of the received TLP and the egress switch ports to which the NT multicast TLP is forwarded.

- NT Multicast is only supported for multicast group IDs 0, 1, 2, and 3.

Each NT Multicast Group x Port Association register corresponds to a multicast group ID (e.g., NTMCG[0]PA corresponds to group 0, NTMCG[1]PA corresponds to group 1, etc.). Each NTMCGxPA register may be programmed to select zero or more ports associated with the corresponding multicast group. Refer to the definition of the NTMCG[3:0]PA registers for details.

- The NTMCGxPA registers must be programmed with ports that are associated with other partitions. These registers must not be programmed with ports that are in the same partition as the NT function that receives the NT multicast TLP. Violation of this rule produces undefined results.
- When the NTMCGxPA register is programmed to select zero ports, multicast TLPs associated with the corresponding group ID are silently dropped by the NT function.

Note that by configuring the multicast groups in the PCI-to-PCI bridge and NT functions appropriately, it is possible that a received TLP be simultaneously multicasted within the partition (i.e., transparent multicast) and across partitions (NT multicast).

NT Multicast Egress Processing

PES32NT8xG2 ports perform NT multicast egress processing, which consists of address and requester ID overlay, prior to transmitting the NT multicast TLP on the data-link. NT multicast address and requester ID overlay processing is performed independently by each port. Therefore, it is possible to configure this functionality independently for each port. Furthermore, within a port, NT multicast address overlay and requester ID overlay may be independently enabled.

- NT multicast address overlay is enabled via the NTMCAOE bit in the NT Multicast Control (NTMCC) register.
- NT multicast address overlay is enabled via the NTMCRIDOE bit in the NT Multicast Control (NTMCC) register.

NT multicast egress processing is performed by the port, and not by a specific function in the port. Therefore, the egress port need not contain an NT function in order to perform NT multicast egress processing. Further, the egress port need not be in a partition that contains an NT function[1].

Notes

In order to perform NT multicast egress processing, each port contains four sets of NT multicast overlay registers, and each set is associated with a source partition and multicast group. Depending on the partition and group on which the NT multicast TLP is received, one of the four NT multicast overlay register sets is selected to control the manner in which the overlay operation is performed on the TLP.

– When none of the four NT multicast overlay register sets matches the partition and group of the received NT multicast TLP, the TLP is transmitted without change (i.e., NT multicast egress processing is not applied).

NT multicast address overlay is similar in nature to transparent multicast address overlay processing, described in section Multicast Egress Processing on page 17-5. It consists of replacing the address of the TLP with an address programmed in a register associated with NT multicast egress processing.

– Since the address maps are independent between switch partitions, this feature allows translation of the TLP's address between the source partition and destination partition(s) for an NT multicast transfer.

NT multicast requester ID overlay processing consists of replacing the requester ID in the NT multicast TLP with a requester ID programmed in a register associated with NT multicast egress processing.

– Since the requester IDs are independent between switch partitions, this feature allows translation of the requester ID between the source partition and destination partition(s) for an NT multicast transfer.

Each port contains four sets of NT multicast overlay registers. Each set is composed of the following registers:

– NT Multicast Overlay x Configuration (NTMCOVRxC)
– NT Multicast Overlay x Base Address Low (NTMCOVRxBARL)
– NT Multicast Overlay x Base Address High (NTMCOVRxBARH)

The NTMCOVRxC register controls the partition and group ID associated with the corresponding set of NT multicast overlay registers.

– The Partition (PART) field in the NTMCOVRxC register selects the partition on which the NT multicast must be received (i.e., the source partition) in order for the NT multicast address and requester ID overlay to be performed.

  The PART field is a bit-vector that may be programmed to associate the corresponding overlay address registers with zero, one, some, or all partitions.

– The Group Association (GROUP) field in the NTMCOVRxC register selects the NT multicast group ID to which the received NT multicast TLP must belong (in the source partition) in order for the NT multicast address and requester ID overlay to be performed.

  The GROUP field is a bit-vector that may be programmed to associate the corresponding overlay address registers with zero, one, some, or all NT multicast groups.

– The programmer must ensure that no conflicts exist between the partition and group selection of the NT Multicast Overlay x Control registers. Specifically, it is prohibited to program these registers such that a partition is associated with two or more NT multicast overlay registers. In addition, it is prohibited to program these registers such that a multicast group in a partition is associated with two or more NT multicast overlay registers. Breaking this rule produces undefined results.

In addition to the PART and GROUP fields, the NTMCOVRxC register has a field that controls requester ID overlay. Specifically, the Overlay Requester ID (OVRREQID) field in this register is used to replace the requester ID field of a multicast TLP transmitted by this port.

For NT Multicast address overlay, the NT Multicast Overlay x Base Address Low (NTMCOVRxBARL) and NT Multicast Overlay x Base Address High (NTMCOVRxBARH) registers are programmed. These registers operate in a manner similar to the transparent operation address overlay registers.

[1] In general, it is expected that the NT egress port will be part of a switch partition that contains an NT function. This allows unicast TLP communication between the source and destination partitions involved in the NT multicast transfer.

Notes

When the OVRSIZE field value is six or greater, NT multicast address overlay processing is performed on all NT multicast TLPs transmitted by the port as described below.

- Address bits in the NT multicast TLP with bit positions greater than or equal to OVRSIZE are replaced by the corresponding address bits in the multicast overlay base address. The multicast overlay base address is contained in the Multicast Overlay BAR Low (MCBARL) field in the NTMCOVRxBARL register and the Multicast Overlay BAR High (MCBARH) field in the MCOVRBARH register.
- Address bits less than OVRSIZE are not modified. As a result of multicast overlay processing, a multicast TLP with an original address above 4 GB may be translated into a multicast TLP with address below 4 GB, and vice-versa. Thus, address translation may change the size of a multicast TLP header (e.g., from 4 DWords to 3 DWords).
- Unlike transparent multicast, for NT multicast the OVRSIZE field in the NTMCOVRxBARL register must not be set to less than six. Otherwise the operation of NT multicast egress processing is undefined. As noted earlier, NT multicast address overlay can be explicitly disabled via the NTMCAOE bit in the NT Multicast Control (NTMCC) register.

A side-effect of modifying the TLP's address and requester ID due to NT multicast overlay processing is that the ECRC associated with the original TLP may not be correct for the new modified TLP. Therefore, egress ports perform the following ECRC processing.[1]

- If NT multicast overlay processing is disabled, then no ECRC processing is performed as part of NT multicast egress processing.
- If an NT multicast TLP does not contain an ECRC, then no ECRC processing is performed as part of NT multicast egress processing.
- If an NT multicast TLP contains an ECRC and NT multicast overlay processing is enabled (i.e., either requester ID overlay or address overlay), then the following actions are performed.

  The ECRC of the original multicast TLP is checked while simultaneously the ECRC for the new modified TLP is computed or "regenerated." This is implemented in the same pipeline stage such that there is virtually no possibility of silent data corruption (e.g., a TLP bit flip that does not result in a computed ECRC error in the original or regenerated ECRC).

  If no error is detected in the ECRC associated with the original TLP, then the modified TLP is forwarded with the regenerated ECRC.

  If an error is detected in the ECRC associated with the original TLP, then the modified TLP is forwarded with inverted regenerated ECRC (i.e., the computed ECRC of the modified TLP is inverted).

After performing egress processing of an NT multicast TLP, the port transmits the TLP on the link. No routing checks are performed and no errors are reported due to NT multicast egress processing.

- The NT multicast TLP transmitted by a port must never be claimed by a function in that port. Otherwise operation is undefined.
- Egress port control registers that normally enable the capability of a port to transmit TLPs (e.g., Bus Master Enable it in the PCICMD register) do not have any effect on NT Multicast TLPs emitted by the port.

Usage Restrictions

The following is a usage restriction associated with NT multicast operation:

- Modifying the port operating mode of a switch port configured to transmit NT multicast TLPs requires that the Operating Mode Change Action (OMA) field in the SWPORTxCTL register be set to reset. Note that this results in the NT Multicast Transmit Enable (NTMCTEN) bit in the port's NTMCC register to be cleared, thus causing the port to stop transmission of NT multicast TLPs. Software must explicitly set the NTMCTEN bit in order to re-enable NT multicast transmission by the port.

1. Note that NT Multicast ECRC processing is not dependent on the setting of the ECRC Checking Enable (ECRCCE) or ECRC Generation Enable (ECRCGE) bits in the AERCTL register of any of the port functions.

Notes

Temperature Sensor

Notes

Overview

The PES32NT8xG2 contains an on-chip temperature sensor that measures junction temperature. The sensor has three programmable temperature thresholds and a temperature history capability. An alarm is generated when the temperature is above or below one of three programmable temperature thresholds.

- These thresholds are programmed via the Temperature Sensor Control (TMPCTL) register.

When an alarm is generated, a status bit is set in the Temperature Sensor Status (TMPSTS). Optionally, an interrupt may be generated by the PCI-to-PCI bridge function or NT function in any partition's upstream port.

- Fields in the TMPCTL register allow selection of which temperature thresholds generate an interrupt.
- The TMPSENSOR bit in the P2PINTSTS register of the PCI-to-PCI bridge function is set when an enabled temperature sensor alarm is set. When unmasked, this bit causes the upstream PCI-to-PCI bridge function to generate an interrupt (i.e., INTx or MSI).
- The TMPSENSOR bit in the NTINTSTS register of the NT function is also set when an enabled temperature sensor alarm is set. When unmasked, this bit causes the NT function to generate an interrupt (i.e., INTx or MSI).

Maximum high and minimum low temperature history is automatically recorded between software accesses. These information is recorded in the TMPSTS register. The temperature sensor has an accuracy of +/- 5 degrees Celsius, with a read-out in the range of 0 to 127.5 degrees Celsius in 0.5 degree increments.

- The die temperature may vary by as much as 10 degrees Celsius.
- Fields in the Temperature Sensor Adjustment (TMPADJ) and Temperature Sensor Slope (TSSLOPE) registers allow the gain and offset to be adjusted for improved accuracy.

Notes

Register Organization

Notes

Overview

All software visible registers in the switch are contained in a 512 KB *global address space*. The address of a register in this address range is referred to as the *system address* of the register.

- The system address is 19-bits in size.
- Currently, the lower 256 KB of the global address space are used for port and switch registers. The upper 256 KB are not used and reserved.

System addresses are used for serial EEPROM initialization and slave SMBus register access. The global address range is divided into regions as shown inTable 19.1.

- There is a 4 KB region for the configuration registers of each PCI-to-PCI bridge function in the switch.
- There is a 4 KB region for the configuration registers of each NT function in the switch.
- There is a 4 KB region for the configuration registers of each DMA function in the switch.
- In addition, there is one 8 KB region for switch configuration and status registers.

Base Address	Address Range
0x00000	Port 0 PCI-to-PCI Bridge Registers
0x01000	Port 0 NT Endpoint Registers
0x02000 - 0x03000	Reserved
0x04000	Port 2 PCI-to-PCI Bridge Registers
0x05000	Port 2 NT Endpoint Registers
0x06000 - 0x07000	Reserved
0x08000	Port 4 PCI-to-PCI Bridge Registers
0x09000	Port 4 NT Endpoint Registers
0x0A000 - 0x0B000	Reserved
0x0C000	Port 6 PCI-to-PCI Bridge Registers
0x0D000	Port 6 NT Endpoint Registers
0x0E000 - 0x0F000	Reserved
0x10000	Port 8 PCI-to-PCI Bridge Registers
0x11000	Port 8 NT Endpoint Registers
0x12000 - 0x17000	Reserved
0x18000	Port 12 PCI-to-PCI Bridge Registers
0x19000	Port 12 NT Endpoint Registers
0x1A000 - 0x1B000	Reserved
0x1C000	Port 14 PCI-to-PCI Bridge Registers
0x1D000 - 0x1F000	Reserved
0x20000	Port 16 PCI-to-PCI Bridge Registers

Table 19.1 Global Address Space Layout (Part 1 of 2)

Notes

Base Address	Address Range
0x21000	Port 16 NT Endpoint Registers
0x22000 - 0x27000	Reserved
0x28000	Port 20 PCI-to-PCI Bridge Registers
0x29000	Port 20 NT Endpoint Registers
0x2A000 - 0x2F000	Reserved
0x30000 - 0x39FFF	Reserved
0x3A000	Port 0 DMA Endpoint
0x3C000	Port 8 DMA Endpoint
0x3E000 - 0x3FFFF	Switch Configuration and Status Registers
Others	Reserved

Table 19.1 Global Address Space Layout (Part 2 of 2)

PCI-to-PCI bridge registers correspond to the configuration registers associated with the PCI-to-PCI bridge function of a port. In addition, proprietary port registers associated with proprietary features are mapped into PCI-to-PCI bridge function's configuration space, starting at offset 0x400 (refer to section PCI-to-PCI Bridge Function Registers on page 19-3). NT endpoint registers correspond to the configuration registers associated with the NT function of a port (refer to section NT Function Registers on page 19-14). DMA endpoint registers correspond to the configuration registers associated with the DMA function of a port (refer to section DMA Function Registers on page 19-23).

The switch configuration and status register region contains registers that control general operation of the switch and are proprietary in nature (e.g., registers to configure the switch ports and partitions, etc.). The offset address for switch configuration and status registers is defined in section Switch Configuration and Status Registers on page 19-29.

The entire device's global address space may be accessed using PCI Express configuration requests from any the device's PCI Express function (e.g., PCI-to-PCI bridge function, NT function, DMA function, etc.).

- Located in each function is a Global Address Space Access Address (GASAADDR) register and a Global Address Space Access Data (GASADATA) register.
- The DWord system address of the register to be accessed is written to the Address (ADDR) field in the GASAADDR register. When a read is performed to the Data (DATA) field in the GASADATA register, the value of the corresponding register selected by the ADDR field is returned. When a write is performed to the DATA field, the value of the corresponding register selected by the ADDR field is updated with the value written.
- Any software visible register in the entire PES32NT8xG2 switch may be accessed using a function's GASAADDR and GASADATA registers, even those associated with functions in other ports and partitions. In some applications it is desirable to restrict access to these registers.
  - Associated with each port is a bit in the Port (PORT) field of the Global Address Space Access Protection (GASAPROT) register. When a bit in this field is set, access to the global address space using the GASAADDR and GASADATA registers from the corresponding port is disabled and all fields in these registers become read only with a value of zero.
- Access to the global address space registers may be done via PCI Express configuration accesses, via the SMBus slave interface, or via serial EEPROM.
  - SMBus or serial EEPROM accesses are not affected by the global address space protection register.

Notes

Partial-Byte Access to Word and DWord Registers

Configuration registers in the switch have different sizes (e.g., Byte, Word, DWord). Registers should be accessed with byte-enables that correspond to their native size or a size of one DWord. For example, a Byte register should be read or written with only one byte enable set, or with all four byte enables set. A DWord register should be read or written with all the byte-enables set.

Configuration Register Side-Effects

There are software visible configuration registers that have a side-effect action when written and this side-effect action may affect the ability of the switch to respond with a completion. A configuration write to such a register always returns a completion to the link partner before the side-effect action is performed.

This is implemented by delaying the side-effect action by 1ms following generation of the completion. If the completion is not accepted by the link partner in this time interval, then the completion will be lost.

The following registers, when written[1], have a side-effect action delay.

- PCI-to-PCI Bridge function registers
  - PHYLSTATE0.FLRET
- Switch Configuration and Status Registers
  - SWPORTxCTL.MODE

Address Maps

This section describes the address maps for regions of the global address space outlined in Table 20.1. Reserved address ranges are outlined in Table 19.1. Reading from a reserved address range returns and undefined value. Writes to a reserved address range complete successfully and have an undefined behavior.

PCI-to-PCI Bridge Function Registers

This section outlines the configuration space associated with a PCI-to-PCI bridge function. These registers are accessible via PCI Express configuration requests to function 0 when the port is configured in the following modes.

- Upstream switch port
- Upstream switch port with DMA function
- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions
- Downstream switch port

These registers are not directly accessible by PCI Express configuration request when a port is configured to operate in the following modes.

- Disabled
- Unattached
- NT function
- NT with DMA function

These registers are always accessible, regardless of the port mode, using global address space access registers (i.e., GASAADDR and GASADATA located in each switch function), via the SMBus slave interface, or via serial EEPROM.

- Access to the Extended Configuration Space Address indirection registers (i.e., ECFGADDR and ECFGDATA) located in the PCI-to-PCI Bridge function is not allowed via the GASAADDR and GASADATA registers.

[1] The side-effect delay is applied by the hardware when the registers listed are written via PCI Express configuration requests, as well as EEPROM or SMBus accesses.

Notes

The PCI-to-PCI bridge function configuration space contains standard[1] registers and capabilities defined for PCI Express, as well as proprietary registers associated with proprietary port-specific features.

- section PCI-to-PCI Bridge Registers on page 19-5 lists the configuration registers defined by PCI Express.
- section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 lists the proprietary port-specific registers.

Figure 20.1 shows the organization of the configuration space for the PCI-to-PCI bridge function.

- Registers with offsets between 0x000 and 0x0FF are associated with PCI Express configuration space.
- Registers with offsets between 0x100 and 0x3FF are associated with PCI Express extended configuration space.

Registers with offsets between 0x400 and 0xFFF are associated with PCI Express extended configuration space but are used for IDT proprietary port-specific registers

- IDT proprietary port-specific registers are described in section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11.

- In order to facilitate access to the PCI Express extended configuration space by legacy PCI software, the PCI-to-PCI bridge configuration space contains the Extended Configuration Space Access Address and Data indirection registers (ECFGADDR and ECFGDATA). Refer to the definition of these registers for further details.
  - The ECFGADDR and ECFGDATA registers can't be used to access the global address space access registers (i.e., GASAADDR and GASADATA).

[1] I.e., Registers defined by PCI Express Base Specification 2.1, PCI Local Bus Specification Revision 3.0, and PCI-to-PCI Bridge Architecture Specification Revision 1.2.

Notes

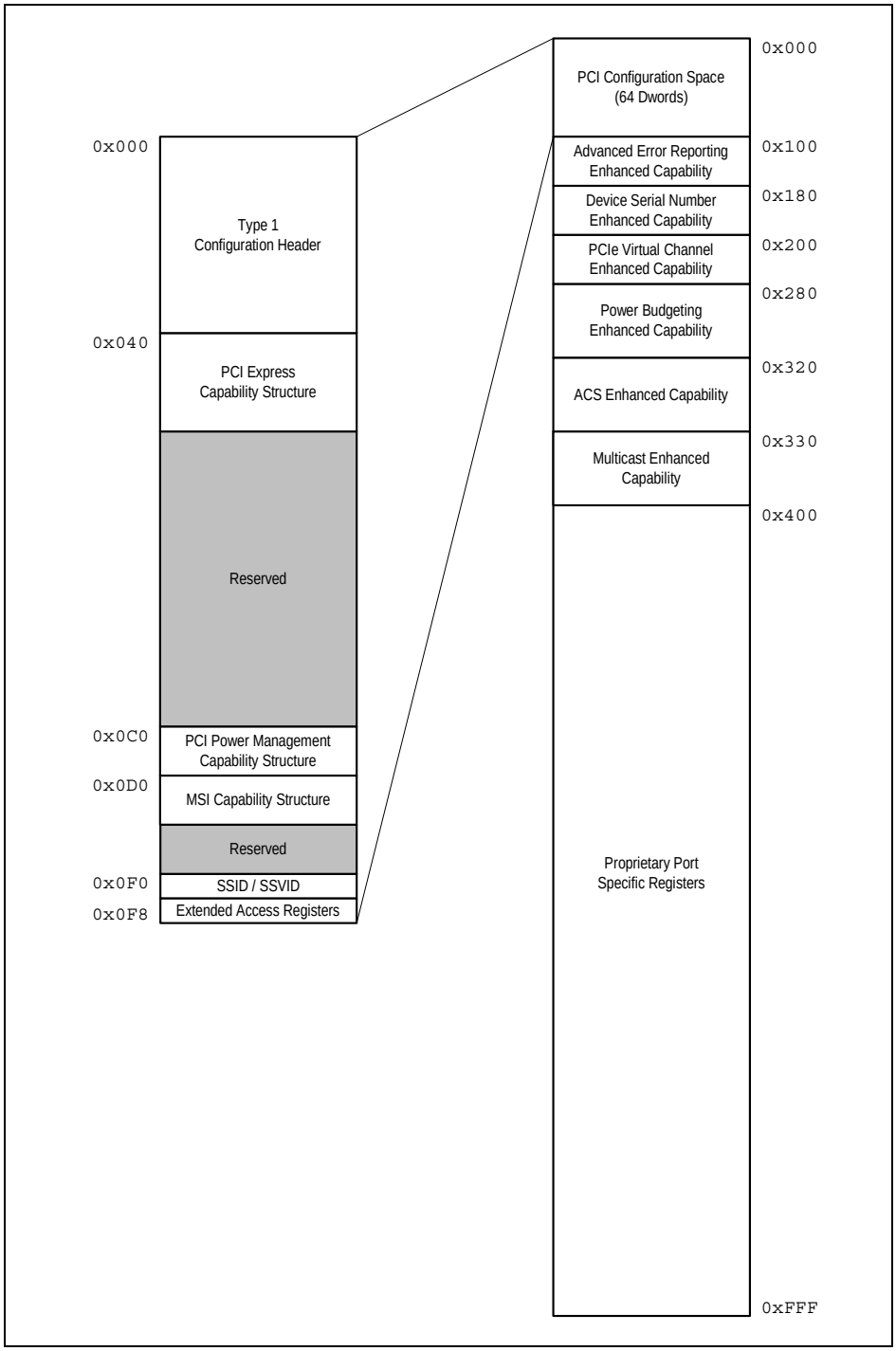

Figure 19.1 PCI-to-PCI Bridge Configuration Space Organization

PCI-to-PCI Bridge Registers

Offset addresses for the standard (i.e., non proprietary) PCI-to-PCI bridge function registers are listed in Table 19.2. Registers in this address range are referenced as `PxP2P_REGNAME` where `x` represents the the switch's port number and `REGNAME` represents the register name in Table 19.2. Reading from an address not defined in Table 19.2 returns a value of zero. Writes to an address not defined in Table 19.2 completes successfully, but modifies no data and has no other effect.

The port operating mode (e.g., upstream switch port, downstream switch port, etc.) determines the presence of some configuration registers within the PCI-to-PCI bridge function's configuration space. For example, the slot capability, slot control, and slot status registers are not present in the configuration space of a PCI-to-PCI bridge function associated with an upstream port. Table 19.2 has two columns indicating the presence of each register within the PCI-to-PCI bridge function's space depending on the port operating mode. Column 'US' refers to a port in upstream mode and column 'DS' refers to a port in downstream switch mode.

The following port operating modes are considered upstream modes:

Upstream switch port

Upstream switch port with DMA function

Upstream switch port with NT function

Upstream switch port with NT and DMA functions

The following port operating modes are considered downstream modes:

Downstream switch port

Unattached[1]

A mark of 'N' in the column indicates that the corresponding register is not present in the configuration space. Otherwise, the register is present in the configuration space. Registers that are not present in the configuration space are considered "reserved" when the port operates in the corresponding operating mode.

Cfg. Offset	Size	Register Mnemonic	Register Definition	US	DS
0x000	Word	VID	VID - Vendor Identification Register (0x000) on page 20-1		
0x002	Word	DID	DID - Device Identification Register (0x002) on page 20-1		
0x004	Word	PCICMD	PCICMD - PCI Command Register (0x004) on page 20-2		
0x006	Word	PCISTS	PCISTS - PCI Status Register (0x006) on page 20-3		
0x008	Byte	RID	RID - Revision Identification Register (0x008) on page 20-4		
0x009	3 Bytes	CCODE	CCODE - Class Code Register (0x009) on page 20-4		
0x00C	Byte	CLS	CLS - Cache Line Size Register (0x00C) on page 20-5		
0x00D	Byte	PLTIMER	PLTIMER - Primary Latency Timer (0x00D) on page 20-5		
0x00E	Byte	HDR	HDR - Header Type Register (0x00E) on page 20-5		
0x00F	Byte	BIST	BIST - Built-in Self Test Register (0x00F) on page 20-5		
0x010	DWord	BAR0	BAR0 - Base Address Register 0 (0x010) on page 20-5		
0x014	DWord	BAR1	BAR1 - Base Address Register (0x014) on page 20-6		
0x018	Byte	PBUSN	PBUSN - Primary Bus Number Register (0x018) on page 20-6		
0x019	Byte	SBUSN	SBUSN - Secondary Bus Number Register (0x019) on page 20-6		
0x01A	Byte	SUBUSN	SUBUSN - Subordinate Bus Number Register (0x01A) on page 20-6		
0x01B	Byte	SLTIMER	SLTIMER - Secondary Latency Timer Register (0x01B) on page 20-6		
0x01C	Byte	IOBASE	IOBASE - I/O Base Register (0x01C) on page 20-7		
0x01D	Byte	IOLIMIT	IOLIMIT - I/O Limit Register (0x01D) on page 20-7		
0x01E	Word	SECSTS	SECSTS - Secondary Status Register (0x01E) on page 20-7		
0x020	Word	MBASE	MBASE - Memory Base Register (0x020) on page 20-8		

Table 19.2 PCI-to-PCI Bridge Function Configuration Space Registers (Part 1 of 4)

1. Refer to section Unattached on page 5-8 for a description of the configuration register space behavior of a port in unattached mode.

Cfg. Offset	Size	Register Mnemonic	Register Definition	US	DS
0x022	Word	MLIMIT	MLIMIT - Memory Limit Register (0x022) on page 20-8		
0x024	Word	PMBASE	PMBASE - Prefetchable Memory Base Register (0x024) on page 20-9		
0x026	Word	PMLIMIT	PMLIMIT - Prefetchable Memory Limit Register (0x026) on page 20-9		
0x028	DWord	PMBASEU	PMBASEU - Prefetchable Memory Base Upper Register (0x028) on page 20-9		
0x02C	DWord	PMLIMITU	PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) on page 20-10		
0x030	Word	IOBASEU	IOBASEU - I/O Base Upper Register (0x030) on page 20-10		
0x032	Word	IOLIMITU	IOLIMITU - I/O Limit Upper Register (0x032) on page 20-10		
0x034	Byte	CAPPTR	CAPPTR - Capabilities Pointer Register (0x034) on page 20-10		
0x038	DWord	EROMBASE	EROMBASE - Expansion ROM Base Address Register (0x038) on page 20-10		
0x03C	Byte	INTRLINE	INTRLINE - Interrupt Line Register (0x03C) on page 20-11		
0x03D	Byte	INTRPIN	INTRPIN - Interrupt PIN Register (0x03D) on page 20-11		
0x03E	Word	BCTL	BCTL - Bridge Control Register (0x03E) on page 20-12		
0x040	DWord	PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 20-13		
0x044	DWord	PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 20-14		
0x048	Word	PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 20-16		
0x04A	Word	PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 20-17		
0x04C	DWord	PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 20-18		
0x050	Word	PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 20-20		
0x052	Word	PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 20-22		
0x054	DWord	PCIESCAP	PCIESCAP - PCI Express Slot Capabilities (0x054) on page 20-24	N	
0x058	Word	PCIESCTL	PCIESCTL - PCI Express Slot Control (0x058) on page 20-26	N	
0x05A	Word	PCIESSTS	PCIESSTS - PCI Express Slot Status (0x05A) on page 20-29	N	
0x064	DWord	PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 20-30		
0x068	Word	PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068) on page 20-31		
0x06A	Word	PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 20-32		
0x06C	DWord	PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 20-32		
0x070	Word	PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 20-32		
0x072	Word	PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 20-34		
0x074	DWord	PCIESCAP2	PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074) on page 20-34	N	
0x078	Word	PCIESCTL2	PCIESCTL2 - PCI Express Slot Control 2 (0x078) on page 20-34	N	
0x07A	Word	PCIESSTS2	PCIESSTS2 - PCI Express Slot Status 2 (0x07A) on page 20-35	N	
0x0C0	DWord	PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 20-35		
0x0C4	DWord	PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 20-36		

Table 19.2 PCI-to-PCI Bridge Function Configuration Space Registers (Part 2 of 4)

Cfg. Offset	Size	Register Mnemonic	Register Definition	US	DS
0x0D0	DWord	MSICAP	MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) on page 20-37	N	
0x0D4	DWord	MSIADDR	MSIADDR - Message Signaled Interrupt Address (0x0D4) on page 20-37	N	
0x0D8	DWord	MSIUADDR	MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) on page 20-38	N	
0x0DC	DWord	MSIMDATA	MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) on page 20-38	N	
0x0F0	DWord	SSIDSSVIDCAP	SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) on page 20-38		
0x0F4	DWord	SSIDSSVID	SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) on page 20-39		
0x0F8	DWord	ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 20-39		
0x0FC	DWord	ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 20-40		
0x100	DWord	AERCAP	AERCAP - AER Capabilities (0x100) on page 20-40		
0x104	DWord	AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 20-40		
0x108	DWord	AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 20-42		
0x10C	DWord	AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 20-44		
0x110	DWord	AERCES	AERCES - AER Correctable Error Status (0x110) on page 20-46		
0x114	DWord	AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 20-47		
0x118	DWord	AERCTL	AERCTL - AER Capabilities and Control (0x118) on page 20-49		
0x11C	DWord	AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 20-49		
0x120	DWord	AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 20-49		
0x124	DWord	AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 20-50		
0x128	DWord	AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 20-50		
0x180	DWord	SNUMCAP	SNUMCAP - Serial Number Capabilities (0x180) on page 20-50		
0x184	DWord	SNUMLDW	SNUMLDW - Serial Number Lower Doubleword (0x184) on page 20-50		
0x188	DWord	SNUMUDW	SNUMUDW - Serial Number Upper Doubleword (0x188) on page 20-51		
0x200	DWord	PCIEVCECAP	PCIEVCECAP - PCI Express VC Extended Capability Header (0x200) on page 20-51		
0x204	DWord	PVCCAP1	PVCCAP1- Port VC Capability 1 (0x204) on page 20-51		
0x208	DWord	PVCCAP2	PVCCAP2- Port VC Capability 2 (0x208) on page 20-52		
0x20C	Word	PVCCTL	PVCCTL - Port VC Control (0x20C) on page 20-52		
0x20E	Word	PVCSTS	PVCSTS - Port VC Status (0x20E) on page 20-52		
0x210	DWord	VCR0CAP	VCR0CAP- VC Resource 0 Capability (0x210) on page 20-53		
0x214	DWord	VCR0CTL	VCR0CTL- VC Resource 0 Control (0x214) on page 20-53		
0x218	DWord	VCR0STS	VCR0STS - VC Resource 0 Status (0x218) on page 20-54		
0x320	DWord	ACSECAPH	ACSECAPH - ACS Extended Capability Header (0x320) on page 20-54		

Table 19.2 PCI-to-PCI Bridge Function Configuration Space Registers (Part 3 of 4)

Cfg. Offset	Size	Register Mnemonic	Register Definition	US	DS
0x324	Word	ACSCAP	ACSCAP - ACS Capability Register (0x324) on page 20-55		
0x326	Word	ACSCTL	ACSCTL - ACS Control Register (0x326) on page 20-57		
0x328	DWord	ACSECV	ACSECV - ACS Egress Control Vector (0x328) on page 20-58		
0x330	DWord	MCCAPH	MCCAPH - Multicast Extended Capability Header (0x330) on page 20-59		
0x334	Word	MCCAP	MCCAP - Multicast Capability (0x334) on page 20-59		
0x336	Word	MCCTL	MCCTL- Multicast Control (0x336) on page 20-60		
0x338	DWord	MCBARL	MCBARL- Multicast Base Address Low (0x338) on page 20-60		
0x33C	DWord	MCBARH	MCBARH- Multicast Base Address High (0x33C) on page 20-61		
0x340	DWord	MCRCVL	MCRCVL- Multicast Receive Low (0x340) on page 20-61		
0x344	DWord	MCRCVH	MCRCVH- Multicast Receive High (0x344) on page 20-61		
0x348	DWord	MCBLKALLL	MCBLKALLL- Multicast Block All Low (0x348) on page 20-62		
0x34C	DWord	MCBLKALLH	MCBLKALLH- Multicast Block All High (0x34C) on page 20-62		
0x350	DWord	MCBLKUTL	MCBLKUTL- Multicast Block Untranslated Low (0x350) on page 20-62		
0x354	DWord	MCBLKUTH	MCBLKUTH - Multicast Block Untranslated High (0x354) on page 20-63		
0x358	DWord	MCOVRBARL	MCOVRBARL- Multicast Overlay Base Address Low (0x358) on page 20-63		
0x35C	DWord	MCOVRBARH	MCOVRBARH- Multicast Overlay Base Address High (0x35C) on page 20-63		

Table 19.2 PCI-to-PCI Bridge Function Configuration Space Registers (Part 4 of 4)

PCI-to-PCI Bridge Capability Structures

A PCI-to-PCI bridge function contains a number of PCI Express capability structures. Following a fundamental reset, some of these capabilities are linked by default (i.e., via the capability structure's next pointer field) and are visible to software while others need to be explicitly linked to become visible (e.g., using firmware, serial EEPROM, or SMBus accesses to modify the next pointer field).

Note: There are two capability lists within the configuration space: PCI Express capabilities list and PCI Express Extended capabilities list.

The default linkage of capabilities depends on the port operating mode. This is necessary as some capabilities are only applicable to downstream switch ports, while others are only applicable for multi-function upstream ports (e.g., ACS capability structure).

Table 19.3 shows the default linkage of the PCI-to-PCI bridge function for ports that operate in the following modes.

- Upstream switch port
- Upstream switch port with NT function
- Upstream switch port with DMA function
- Upstream switch port with NT and DMA functions

Table entries shaded in green indicate capabilities that are linked by default. Entries shaded in pink are capabilities that may be linked by firmware (e.g., via the EEPROM). Note that the ACS Extended Capability structure is not applicable and must not be linked when the port operates in upstream switch port mode (since the upstream port only has one function in this mode).

Capability List	Capability Structure Name	Cfg Space Offset	Default Value of Next Pointer field (NXTPTR)
PCI Express Capabilities List	PCI Express Capability Structure	0x040	0x0C0
	PCI Power Management Capability Structure	0x0C0	0x0
	Message Signaled Interrupt Capability Structure	0x0D0	0x0
	Subsystem ID and Subsystem Vendor ID	0x0F0	0x0
PCI Express Extended Capabilities List	Advanced Error Reporting (AER) Extended Capability	0x100	0x200
	Device Serial Number Extended Capability	0x180	0x0
	PCI Express Virtual Channel Capability	0x200	0x330
	ACS Extended Capability[1]	0x320	0x0
	Multicast Extended Capability	0x330	0x0

Table 19.3 Default Linkage of Capability Structures for a PCI-to-PCI Bridge Function in the Upstream Switch Port Mode

1. The ACS capability structure must not be linked when the upstream port operates in upstream switch port mode, as ACS is only applicable for multi-function port modes in an upstream port.

Table 19.4 shows the default linkage of the PCI-to-PCI bridge function for ports that operate in the following modes:

- Downstream switch port
- Unattached

Table entries shaded in green indicate capabilities that are linked by default. Entries shaded in pink are capabilities that may be linked by firmware (e.g., via the EEPROM).

Capability List	Capability Structure Name	Cfg Space Offset	Default Value of Next Pointer field (NXTPTR)
PCI Express Capabilities List	PCI Express Capability Structure	0x040	0x0C0
	PCI Power Management Capability Structure	0x0C0	0x0D0
	Message Signaled Interrupt Capability Structure	0x0D0	0x0
	Subsystem ID and Subsystem Vendor ID	0x0F0	0x0
PCI Express Extended Capabilities List	Advanced Error Reporting (AER) Extended Capability	0x100	0x200
	Device Serial Number Extended Capability	0x180	0x0
	PCI Express Virtual Channel Capability	0x200	0x320
	ACS Extended Capability	0x320	0x330
	Multicast Extended Capability	0x330	0x0

Table 19.4 Default Linkage of Capability Structures for a PCI-to-PCI Bridge Function in a Downstream or Unattached Port

Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function

This section outlines the address range 0x400 through 0xFFF in the configuration space of the PCI-to-PCI bridge function. This address range contains IDT proprietary registers that are port-specific (i.e., provide control or status on a per-port basis). These registers control proprietary functionality or provide status beyond the functionality outlined in the PCI Express Base Specification. In some cases, the proprietary functionality is associated with a specific port function (e.g., PCI-to-PCI bridge function), and in other cases with the operation of the port (i.e., affecting all it's functions).

Proprietary port-specific registers always affect the operation of the port and always reflect status associated with the port, regardless of the port's operating mode. For port operating modes in which the PCI-to-PCI bridge function is not present in the port, the proprietary port registers are only accessible via the switch's global address space. In such operating modes, the proprietary port registers continue to have effect on the operation of the port.

Registers in this address range may be accessed using PCI Express configuration requests to the corresponding PCI-to-PCI bridge function (if present in the port), through the global address space access registers, via the SMBus slave interface, or via serial EEPROM. Offset addresses for proprietary port specific registers are listed in Table 19.5. Registers in this address range are referenced as `Px_REGNAME` where `x` represents the switch port number and `REGNAME` represents the register name in Table 19.5. Reading from an address not defined in Table 19.5 returns a value of zero. Writes to an address not defined in Table 19.5 completes successfully, but modifies no data and has no other effect.

Figure 19.2 shows the organization of the proprietary port specific registers.

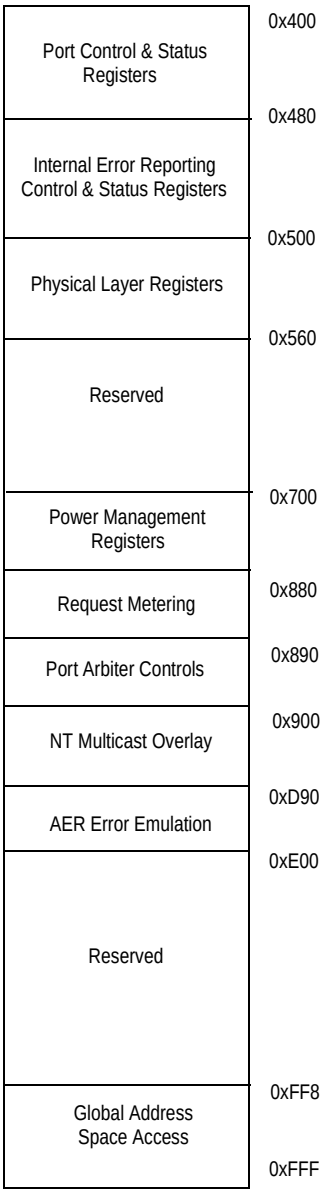

Figure 19.2 Proprietary Port Specific Register Organization

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x400	DWord	PORTCTL	PORTCTL - Port Control (0x400) on page 21-1
0x404	DWord	P2PINTSTS	P2PINTSTS - PCI-to-PCI Bridge Interrupt Status (0x404) on page 21-1
0x408	DWord	P2PINTMSK	P2PINTMSK - PCI-to-PCI Bridge Interrupt Mask (0x408) on page 21-2
0x410	DWord	P2PSDATA	P2PSDATA - PCI-to-PCI Bridge Signal Data (0x410) on page 21-3
0x414	DWord	P2PGSIGNAL	P2PGSIGNAL - PCI-to-PCI Bridge Global Signal (0x414) on page 21-3
0x424	DWord	PAERMSK	PAERMSK - Port AER Mask (0x424) on page 21-3
0x430	DWord	PCIESCTLIV	PCIESCTLIV - PCI Express Slot Control Initial Value (0x430) on page 21-5
0x480	DWord	IERRORCTL	IERRORCTL - Internal Error Reporting Control (0x480) on page 21-7
0x484	DWord	IERRORSTS0	IERRORSTS0 - Internal Error Reporting Status 0 (0x484) on page 21-7
0x488	DWord	IERRORSTS1	IERRORSTS1 - Internal Error Reporting Status 1 (0x488) on page 21-10
0x48C	DWord	IERRORSEV0	IERRORSEV0 - Internal Error Reporting Severity 0 (0x48C) on page 21-11
0x490	DWord	IERRORSEV1	IERRORSEV1 - Internal Error Reporting Severity 1 (0x490) on page 21-14
0x494	DWord	IERRORTST0	IERRORTST0 - Internal Error Reporting Test 0 (0x494) on page 21-16
0x498	DWord	IERRORTST1	IERRORTST1 - Internal Error Reporting Test 1 (0x498) on page 21-18
0x4A0	DWord	P2PIERRORMSK0	P2PIERRORMSK0 - PCI-to-PCI Bridge Internal Error Reporting Mask 0 (0x4A0) on page 21-18
0x4A4	DWord	P2PIERRORMSK1	P2PIERRORMSK1 - PCI-to-PCI Bridge Internal Error Reporting Mask 1 (0x4A4) on page 21-22
0x510	DWord	SERDESCFG	SERDESCFG - SerDes Configuration (0x510) on page 21-24
0x51C	DWord	LANESTS0	LANESTS0 - Lane Status 0 (0x51C) on page 21-24
0x520	DWord	LANESTS1	LANESTS1 - Lane Status 1 (0x520) on page 21-25
0x530	DWord	PHYLCFG0	PHYLCFG0 - Phy Link Configuration 0 (0x530) on page 21-25
0x540	DWord	PHYLSTATE0	PHYLSTATE0 - Phy Link State 0 (0x540) on page 21-26
0x55C	DWord	PHYPRBS	PHYPRBS - Phy PRBS Seed (0x55C) on page 21-27
0x690	DWord	TLCNTCFG	TLCNTCFG - Transaction Layer Countables Configuration (0x690) on page 21-27
0x710	DWord	L1ASPMRTC	L1ASPMRTC - L1 ASPM Rejection Timer Control (0x710) on page 21-27
0x880	DWord	RMCTL	RMCTL - Requester Metering Control (0x880) on page 21-28
0x88C	DWord	RMCOUNT	RMCOUNT - Requester Metering Count (0x88C) on page 21-29
0x890	DWord	VC0PARBCI0	VC0PARBCI0 - VC0 Port Arbiter Counter Initialization 0 (0x890) on page 21-29
0x894	DWord	VC0PARBCI1	VC0PARBCI1 - VC0 Port Arbiter Counter Initialization 1 (0x894) on page 21-30
0x898	DWord	VC0PARBCI2	VC0PARBCI2 - VC0 Port Arbiter Counter Initialization 2 (0x898) on page 21-30
0x89C	DWord	VC0PARBCI3	VC0PARBCI3 - VC0 Port Arbiter Counter Initialization 3 (0x89C) on page 21-30
0x8A0	DWord	VC0PARBCI4	VC0PARBCI4 - VC0 Port Arbiter Counter Initialization 4 (0x8A0) on page 21-31
0x8A4	DWord	VC0PARBCI5	VC0PARBCI5 - VC0 Port Arbiter Counter Initialization 5 (0x8A4) on page 21-31
0x8A8	DWord	VC0PARBCI6	VC0PARBCI6 - VC0 Port Arbiter Counter Initialization 6 (0x8A8) on page 21-31

Table 19.5 Proprietary Port Specific Registers (Part 1 of 2)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x900	DWord	NTMCC	NTMCC - NT Multicast Control (0x900) on page 21-32
0x904	DWord	NTMCOVR0C	NTMCOVR[3:0]C - NT Multicast Overlay x Configuration on page 21-33
0x908	DWord	NTMCOVR0BARL	NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low on page 21-34
0x90C	DWord	NTMCOVR0BARH	NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High on page 21-34
0x910	DWord	NTMCOVR1C	NTMCOVR[3:0]C - NT Multicast Overlay x Configuration on page 21-33
0x914	DWord	NTMCOVR1BARL	NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low on page 21-34
0x918	DWord	NTMCOVR1BARH	NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High on page 21-34
0x91C	DWord	NTMCOVR2C	NTMCOVR[3:0]C - NT Multicast Overlay x Configuration on page 21-33
0x920	DWord	NTMCOVR2BARL	NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low on page 21-34
0x924	DWord	NTMCOVR2BARH	NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High on page 21-34
0x928	DWord	NTMCOVR3C	NTMCOVR[3:0]C - NT Multicast Overlay x Configuration on page 21-33
0x92C	DWord	NTMCOVR3BARL	NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low on page 21-34
0x930	DWord	NTMCOVR3BARH	NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High on page 21-34
0xD90	DWord	P2PUEEM	P2PUEEM - PCI-to-PCI Bridge Uncorrectable Error Emulation (0xD90) on page 21-34
0xD94	DWord	P2PCEEM	P2PCEEM - PCI-to-PCI Bridge Correctable Error Emulation (0xD94) on page 21-36
0xFF8	DWord	GASAADDR	GASAADDR - Global Address Space Access Address (0xFF8) on page 21-37
0xFFC	DWord	GASADATA	GASADATA - Global Address Space Access Data (0xFFC) on page 21-37

Table 19.5 Proprietary Port Specific Registers (Part 2 of 2)

NT Function Registers

This section outlines the configuration space associated the NT function. These registers are accessible via PCI Express configuration requests to function 0 of a port when the port is configured to operate in the following modes:

- NT function
- NT with DMA function

These registers are accessible via PCI Express configuration requests to function 1 of a port when the port is configured to operate in the following modes:

- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions

These registers are not directly accessible by PCI Express configuration request when a port is configured to operate in the following modes:

- Disabled
- Unattached
- Upstream switch port
- Upstream switch port with DMA function
- Downstream switch port

These registers are always accessible regardless of the port operating mode via the global address space access registers (i.e., GASAADDR and GASADATA), via the SMBus slave interface, or via serial EEPROM. Restrictions apply when using the GASAADDR and GASADATA registers. Refer to the definition of these registers for details. Note that the NT function allows mapping of its configuration space to memory space using the BAR 0 aperture. Thus, these configuration space registers may be accessed via memory read or writes. See section Mapping NT Configuration Space to BAR 0 on page 14-4 for details.

Figure 19.3 shows the organization of the configuration space.

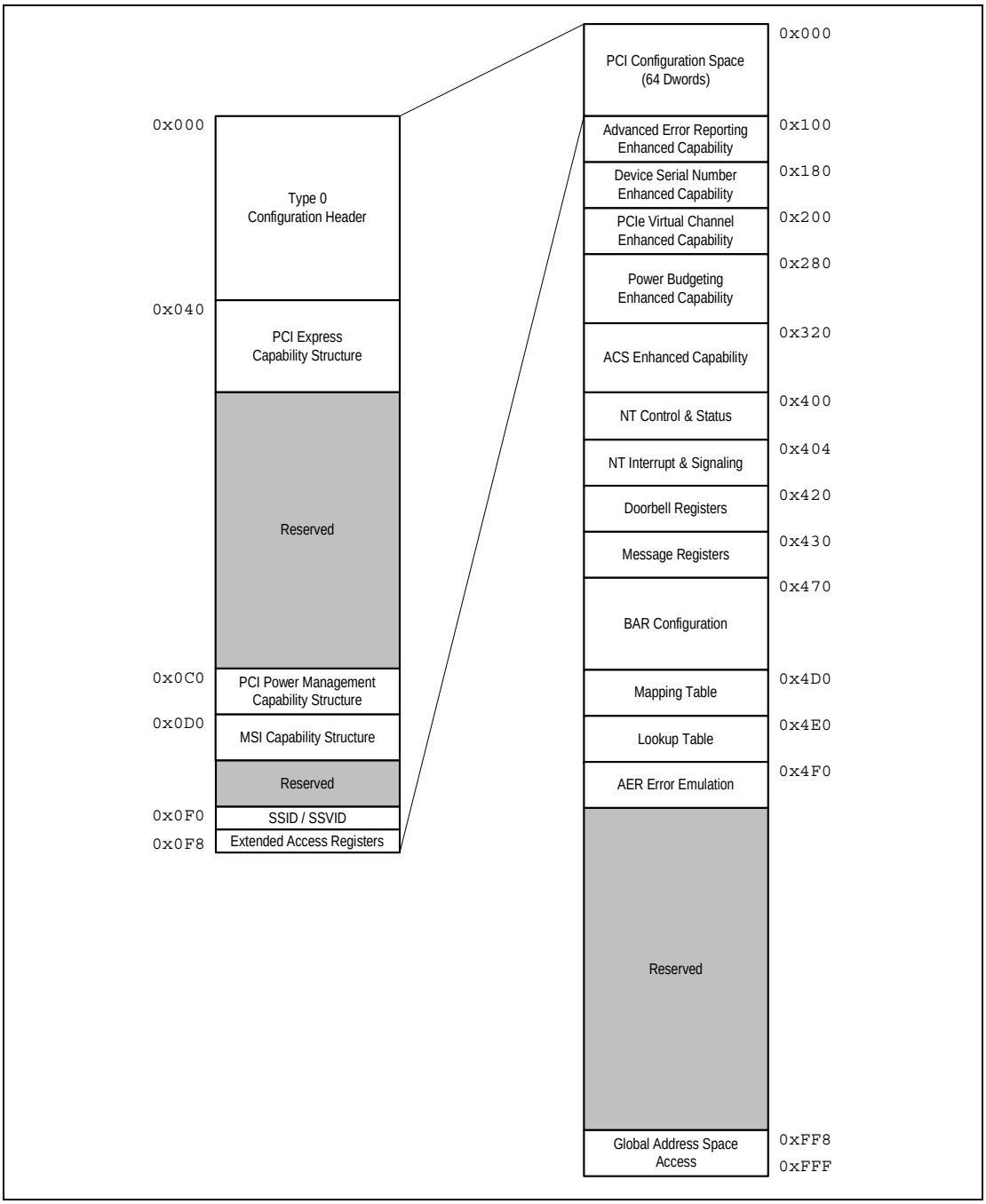

Figure 19.3 NT Function Configuration Space Organization

Offset addresses for NT function registers are listed in Table 19.6. Registers in this address range are referenced as `PxNT_REGNAME` where `x` represents the switch's port number and `REGNAME` represents the register name in Table 19.6. Reading from an address not defined in Table 19.6 returns a value of zero. Writes to an address not defined in Table 19.6 completes successfully, but modifies no data and has no other effect.

In order to facilitate PCI legacy software access to the PCI Express extended configuration space within the NT endpoint's configuration space, the NT endpoint's configuration space contains the Extended Configuration Space Access Address and Data registers (ECFGADDR and ECFGDATA). Refer to the definition of these registers for further details.

The port operating mode (e.g., NT function mode, upstream switch port with NT function mode, etc.) determines the presence of some capability structures within the NT function's configuration space. For example, the VC capability structure is only present in the configuration space when the port operates in NT function mode (since the NT function is function 0 of the port). Refer to section NT Function Capability Structures on page 19-22 for details on the capability structures present in the NT function depending on the port operating mode.

Table 19.6 has columns indicating the presence of each register within the NT function's space depending on the port operating mode. This matches the default capability structure linkage described in section NT Function Capability Structures on page 19-22. Column 'F0' refers to a port in the following operating modes. In these modes, the NT function is function 0 of the port.

- NT function
- NT with DMA function

Column 'F1' refers to a port in the following operating modes. In these modes, the NT function is function 1 of the port.

- Upstream switch port with NT
- Upstream switch port with NT and DMA functions

A mark of 'N' in the column indicates that the corresponding register is not present in the configuration space. Else, the register is present in the configuration space.

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x000	Word	VID	VID - Vendor Identification (0x000) on page 22-1		
0x002	Word	DID	DID - Device Identification (0x002) on page 22-1		
0x004	Word	PCICMD	PCICMD - PCI Command (0x004) on page 22-1		
0x006	Word	PCISTS	PCISTS - PCI Status (0x006) on page 22-2		
0x008	Byte	RID	RID - Revision Identification (0x008) on page 22-4		
0x009	3 Bytes	CCODE	CCODE - Class Code (0x009) on page 22-4		
0x00C	Byte	CLS	CLS - Cache Line Size (0x00C) on page 22-4		
0x00D	Byte	LTIMER	LTIMER - Latency Time (0x00D) on page 22-4		
0x00E	Byte	HDR	HDR - Header Type (0x00E) on page 22-4		
0x00F	Byte	BIST	BIST - Built-in Self Test Register (0x00F) on page 22-5		
0x010	DWord	BAR0	BAR0 - Base Address Register 0 (0x010) on page 22-5		
0x014	DWord	BAR1	BAR1 - Base Address Register 1 (0x014) on page 22-6		
0x018	DWord	BAR2	BAR2 - Base Address Register 2 (0x018) on page 22-7		
0x01C	DWord	BAR3	BAR3 - Base Address Register 3 (0x01C) on page 22-8		
0x020	DWord	BAR4	BAR4 - Base Address Register 4 (0x020) on page 22-9		
0x024	DWord	BAR5	BAR5 - Base Address Register 5 (0x024) on page 22-10		
0x028	DWord	CCISPTR	CCISPTR - CardBus CIS Pointer (0x028) on page 22-11		
0x02C	Word	SUBVID	SUBVID - Subsystem Vendor ID Pointer (0x02C) on page 22-11		
0x02E	Word	SUBID	SUBID - Subsystem ID Pointer (0x02E) on page 22-11		
0x030	Word	EROMBASE	EROMBASE - Expansion ROM Base (0x030) on page 22-11		
0x034	Byte	CAPPTR	CAPPTR - Capabilities Pointer (0x034) on page 22-11		
0x03C	Byte	INTRLINE	INTRLINE - Interrupt Line (0x03C) on page 22-12		
0x03D	Byte	INTRPIN	INTRPIN - Interrupt PIN (0x03D) on page 22-12		

Table 19.6 NT Function Registers (Part 1 of 6)

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x03E	Byte	MINGNT	MINGNT - Minimum Grant (0x03E) on page 22-12		
0x03F	Byte	MAXLAT	MAXLAT - Maximum Latency (0x03F) on page 22-12		
0x040	DWord	PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 22-13		
0x044	DWord	PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 22-13		
0x048	Word	PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 22-15		
0x04A	Word	PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 22-17		
0x04C	DWord	PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 22-18		
0x050	Word	PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 22-19		
0x052	Word	PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 22-21		
0x064	DWord	PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 22-22		
0x068	Word	PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068) on page 22-23		
0x06A	Word	PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 22-24		
0x06C	DWord	PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 22-24		
0x070	Word	PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 22-24		
0x072	Word	PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 22-26		
0x0C0	DWord	PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 22-27		
0x0C4	DWord	PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 22-27		
0x0D0	DWord	MSICAP	MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) on page 22-28		
0x0D4	DWord	MSIADDR	MSIADDR - Message Signaled Interrupt Address (0x0D4) on page 22-29		
0x0D8	DWord	MSIUADDR	MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) on page 22-29		
0x0DC	DWord	MSIMDATA	MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) on page 22-29		
0x0F0	DWord	SSIDSSVIDCAP	SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) on page 22-30		
0x0F4	DWord	SSIDSSVID	SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) on page 22-30		
0x0F8	DWord	ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 22-30		
0x0FC	DWord	ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 22-31		
0x100	DWord	AERCAP	AERCAP - AER Capabilities (0x100) on page 22-32		
0x104	DWord	AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 22-32		
0x108	DWord	AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 22-33		
0x10C	DWord	AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 22-36		
0x110	DWord	AERCES	AERCES - AER Correctable Error Status (0x110) on page 22-38		
0x114	DWord	AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 22-39		

Table 19.6 NT Function Registers (Part 2 of 6)

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x118	DWord	AERCTL	AERCTL - AER Control (0x118) on page 22-41		
0x11C	DWord	AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 22-42		
0x120	DWord	AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 22-42		
0x124	DWord	AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 22-42		
0x128	DWord	AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 22-42		
0x180	DWord	SNUMCAP	SNUMCAP - Serial Number Capabilities (0x180) on page 22-42		
0x184	DWord	SNUMLDW	SNUMLDW - Serial Number Lower Doubleword (0x184) on page 22-43		
0x188	DWord	SNUMUDW	SNUMUDW - Serial Number Upper Doubleword (0x188) on page 22-43		
0x200	DWord	PCIEVCECAP	PCIEVCECAP - PCI Express VC Extended Capability Header (0x200) on page 22-44		N
0x204	DWord	PVCCAP1	PVCCAP1- Port VC Capability 1 (0x204) on page 22-44		N
0x208	DWord	PVCCAP2	PVCCAP2- Port VC Capability 2 (0x208) on page 22-45		N
0x20C	Word	PVCCTL	PVCCTL - Port VC Control (0x20C) on page 22-45		N
0x20E	Word	PVCSTS	PVCSTS - Port VC Status (0x20E) on page 22-45		N
0x210	DWord	VCR0CAP	VCR0CAP- VC Resource 0 Capability (0x210) on page 22-45		N
0x214	DWord	VCR0CTL	VCR0CTL- VC Resource 0 Control (0x214) on page 22-46		N
0x218	DWord	VCR0STS	VCR0STS - VC Resource 0 Status (0x218) on page 22-46		N
0x320	DWord	ACSECAPH	ACSECAPH - ACS Extended Capability Header (0x320) on page 22-47		
0x324	Word	ACSCAP	ACSCAP - ACS Capability (0x324) on page 22-47		
0x326	Word	ACSCTL	ACSCTL - ACS Control (0x326) on page 22-48		
0x330	DWord	MCCAPH	MCCAPH - Multicast Extended Capability Header (0x330) on page 22-49		
0x334	Word	MCCAP	MCCAP - Multicast Capability (0x334) on page 22-49		
0x336	Word	MCCTL	MCCTL- Multicast Control (0x336) on page 22-49		
0x338	DWord	MCBARL	MCBARL- Multicast Base Address Low (0x338) on page 22-50		
0x33C	DWord	MCBARH	MCBARH- Multicast Base Address High (0x33C) on page 22-50		
0x340	DWord	MCRCVL	MCRCVL- Multicast Receive Low (0x340) on page 22-51		
0x344	DWord	MCRCVH	MCRCVH- Multicast Receive High (0x344) on page 22-51		
0x348	DWord	MCBLKALLL	MCBLKALLL- Multicast Block All Low (0x348) on page 22-51		
0x34C	DWord	MCBLKALLH	MCBLKALLH- Multicast Block All High (0x34C) on page 22-51		
0x350	DWord	MCBLKUTL	MCBLKUTL- Multicast Block Untranslated Low (0x350) on page 22-52		
0x354	DWord	MCBLKUTH	MCBLKUTH - Multicast Block Untranslated High (0x354) on page 22-52		
0x400	DWord	NTCTL	NTCTL - NT Endpoint Control (0x400) on page 22-52		
0x404	DWord	NTINTSTS	NTINTSTS - NT Endpoint Interrupt Status (0x404) on page 22-53		
0x408	DWord	NTINTMSK	NTINTMSK - NT Endpoint Interrupt Mask (0x408) on page 22-54		
0x40C	DWord	NTSDATA	NTSDATA - NT Endpoint Signal Data (0x40C) on page 22-54		
0x410	DWord	NTGSIGNAL	NTGSIGNAL - NT Endpoint Global Signal (0x410) on page 22-55		

Table 19.6 NT Function Registers (Part 3 of 6)

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x414	DWord	NTIERRORMSK0	NTIERRORMSK0 - Internal Error Reporting Mask 0 (0x414) on page 22-55		
0x418	DWord	NTIERRORMSK1	NTIERRORMSK1 - Internal Error Reporting Mask 1 (0x418) on page 22-59	N	
0x420	DWord	OUTDBELLSET	OUTDBELLSET - NT Outbound Doorbell Set (0x420) on page 22-61	N	
0x428	DWord	INDBELLSTS	INDBELLSTS - NT Inbound Doorbell Status (0x428) on page 22-61	N	
0x42C	DWord	INDBELLMSK	INDBELLMSK - NT Inbound Doorbell Mask (0x42C) on page 22-61		
0x430	DWord	OUTMSG0	OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C) on page 22-62		
0x434	DWord	OUTMSG1	OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C) on page 22-62		
0x438	DWord	OUTMSG2	OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C) on page 22-62		
0x43C	DWord	OUTMSG3	OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C) on page 22-62		
0x440	DWord	INMSG0	INMSG[3:0] - Inbound Message [3:0] (0x440-44C) on page 22-62		
0x444	DWord	INMSG1	INMSG[3:0] - Inbound Message [3:0] (0x440-44C) on page 22-62		
0x448	DWord	INMSG2	INMSG[3:0] - Inbound Message [3:0] (0x440-44C) on page 22-62		
0x44C	DWord	INMSG3	INMSG[3:0] - Inbound Message [3:0] (0x440-44C) on page 22-62		
0x450	DWord	INMSGSRC0	INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C) on page 22-62		
0x454	DWord	INMSGSRC1	INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C) on page 22-62		
0x458	DWord	INMSGSRC2	INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C) on page 22-62		
0x45C	DWord	INMSGSRC3	INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C) on page 22-62		
0x460	DWord	MSGSTS	MSGSTS - Message Status (0x460) on page 22-62		
0x464	DWord	MSGTSMSK	MSGSTSMSK - Message Status Mask (0x464) on page 22-63		
0x470	DWord	BARSETUP0	BARSETUP0 - BAR 0 Setup (0x470) on page 22-64		
0x474	DWord	BARLIMIT0	BARLIMIT0 - BAR 0 Limit Address (0x474) on page 22-66		
0x478	DWord	BARLTBASE0	BARLTBASE0 - BAR 0 Lower Translated Base Address (0x478) on page 22-66		
0x47C	DWord	BARUTBASE0	BARUTBASE0 - BAR 0 Upper Translated Base Address (0x47C) on page 22-67		
0x480	DWord	BARSETUP1	BARSETUP1 - BAR 1 Setup (0x480) on page 22-67		
0x484	DWord	BARLIMIT1	BARLIMIT1 - BAR 1 Limit Address (0x484) on page 22-69		
0x488	DWord	BARLTBASE1	BARLTBASE1 - BAR 1 Lower Translated Base Address (0x488) on page 22-70		
0x48C	DWord	BARUTBASE1	BARUTBASE1 - BAR 1 Upper Translated Base Address (0x48C) on page 22-70		
0x490	DWord	BARSETUP2	BARSETUP2 - BAR 2 Setup (0x490) on page 22-70		
0x494	DWord	BARLIMIT2	BARLIMIT2 - BAR 2 Limit Address (0x494) on page 22-72		
0x498	DWord	BARLTBASE2	BARLTBASE2 - BAR 2 Lower Translated Base Address (0x498) on page 22-72		

Table 19.6 NT Function Registers (Part 4 of 6)

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x49C	DWord	BARUTBASE2	BARUTBASE2 - BAR 2 Upper Translated Base Address (0x49C) on page 22-73		
0x4A0	DWord	BARSETUP3	BARSETUP3 - BAR 3 Setup (0x4A0) on page 22-73		
0x4A4	DWord	BARLIMIT3	BARLIMIT3 - BAR 3 Limit Address (0x4A4) on page 22-75		
0x4A8	DWord	BARLTBASE3	BARLTBASE3 - BAR 3 Lower Translated Base Address (0x4A8) on page 22-75		
0x4AC	DWord	BARUTBASE3	BARUTBASE3 - BAR 3 Upper Translated Base Address (0x4AC) on page 22-76		
0x4B0	DWord	BARSETUP4	BARSETUP4 - BAR 4 Setup (0x4B0) on page 22-76		
0x4B4	DWord	BARLIMIT4	BARLIMIT4 - BAR 4 Limit Address (0x4B4) on page 22-78		
0x4B8	DWord	BARLTBASE4	BARLTBASE4 - BAR 4 Lower Translated Base Address (0x4B8) on page 22-78		
0x4BC	DWord	BARUTBASE4	BARUTBASE4 - BAR 4 Upper Translated Base Address (0x4BC) on page 22-79		
0x4C0	DWord	BARSETUP5	BARSETUP5 - BAR 5 Setup (0x4C0) on page 22-79		
0x4C4	DWord	BARLIMIT5	BARLIMIT5 - BAR 5 Limit Address (0x4C4) on page 22-81		
0x4C8	DWord	BARLTBASE5	BARLTBASE5 - BAR 5 Lower Translated Base Address (0x4C8) on page 22-82		
0x4CC	DWord	BARUTBASE5	BARUTBASE5 - BAR 5 Upper Translated Base Address (0x4CC) on page 22-82		
0x4D0	DWord	NTMTBLADDR	NTMTBLADDR - NT Mapping Table Address (0x4D0) on page 22-82		
0x4D4	DWord	NTMTBLSTS	NTMTBLSTS - NT Mapping Table Status (0x4D4) on page 22-83		
0x4D8	DWord	NTMTBLDATA	NTMTBLDATA - NT Mapping Table Data (0x4D8) on page 22-83		
0x4DC	DWord	REQIDCAP	REQIDCAP - Requester ID Capture (0x4DC) on page 22-84		
0x4E0	DWord	LUTOFFSET	LUTOFFSET - Lookup Table Offset (0x4E0) on page 22-85		
0x4E4	DWord	LUTLDATA	LUTLDATA - Lookup Table Lower Data (0x4E4) on page 22-85		
0x4E8	DWord	LUTMDATA	LUTMDATA - Lookup Table Middle Data (0x4E8) on page 22-86		
0x4EC	DWord	LUTUDATA	LUTUDATA - Lookup Table Upper Data (0x4EC) on page 22-86		
0x4F0	DWord	NTUEEM	NTUEEM - NT Endpoint Uncorrectable Error Emulation (0x4F0) on page 22-86		
0x4F4	DWord	NTCEEM	NTCEEM - NT Endpoint Correctable Error Emulation (0x4F4) on page 22-88		
0x510	DWord	PTCCTL0	PTCCTL0 - Punch-Through Configuration Control 0 (0x510) on page 22-89		
0x514	DWord	PTCCTL1	PTCCTL1 - Punch-Through Configuration Control 1 (0x514) on page 22-90		
0x518	DWord	PTCDATA	PTCDATA - Punch-Through Data (0x518) on page 22-90		
0x51C	DWord	PTCSTS	PTCSTS - Punch-Through Status (0x51C) on page 22-90		
0x600	DWord	NTMCG0PA	NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C) on page 22-91		
0x604	DWord	NTMCG1PA	NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C) on page 22-91		

Table 19.6 NT Function Registers (Part 5 of 6)

Cfg. Offset	Size	Register Mnemonic	Register Definition	F0	F1
0x608	DWord	NTMCG2PA	NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C) on page 22-91		
0x60C	DWord	NTMCG3PA	NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C) on page 22-91		
0xFF8	DWord	GASAADDR	GASAADDR - Global Address Space Access Address (0xFF8) on page 22-92		
0xFFC	DWord	GASADATA	GASADATA - Global Address Space Access Data (0xFFC) on page 22-92		

Table 19.6 NT Function Registers (Part 6 of 6)

NT Function Capability Structures

The NT function contains within its configuration space a number of PCI capability structures and PCI Express extended capability structures. Following a fundamental reset, some of these capabilities are linked by default (i.e., via the capability structure's next pointer field) and are visible to software while others need to be explicitly linked to become visible (e.g., using firmware, serial EEPROM, or SMBus slave interface accesses to modify the next pointer field). Note that there are two capability lists within the configuration space: PCI Express capabilities list and PCI Express Extended capabilities list.

The default linkage of capabilities in the NT function depends on the port operating mode. This is necessary as some capabilities are only applicable for multi-function upstream ports (e.g., ACS capability structure). Table 19.7 shows the default capabilities linkage of the NT function for ports that operate in the following modes:

- NT Function
- NT with DMA function

Table entries shaded in green indicate capabilities that are linked by default. Entries shaded in pink are capabilities that may be linked by firmware (e.g., via the EEPROM).

Note: In NT function mode, the ACS Extended Capability structure is not applicable and must never be linked (ACS is not applicable to single function endpoints).

Capability List	Capability Structure Name	Cfg Space Offset	Default Value of Next Pointer field (NXTPTR)
PCI Express Capabilities List	PCI Express Capability Structure	0x040	0x0C0
	PCI Power Management Capability Structure	0x0C0	0x0D0
	Message Signaled Interrupt Capability Structure	0x0D0	0x0
	Subsystem ID and Subsystem Vendor ID	0x0F0	0x0
PCI Express Extended Capabilities List	Advanced Error Reporting (AER) Extended Capability	0x100	0x200
	Device Serial Number Extended Capability	0x180	0x0
	PCI Express Virtual Channel Capability	0x200	0x330
	ACS Extended Capability[1]	0x320	0x0
	Multicast Extended Capability	0x330	0x0

Table 19.7 Default Linkage of Capability Structures for the NT Function When Operating as Function 0 of the Port

1. The ACS capability structure must not be linked when the port operates in NT function mode, as ACS is not applicable to single-function endpoints.

Table 19.8 shows the default capabilities linkage of the NT function for ports that operate in the following modes.

- Upstream switch port with NT function
- Upstream switch port with NT and DMA functions

Table entries shaded in green indicate capabilities that are linked by default. Entries shaded in pink are capabilities that may be linked by firmware (e.g., via the EEPROM).

Note: In these operating modes, the VC Capability structure is not applicable to the NT function and must never be linked.

Capability List	Capability Structure Name	Cfg Space Offset	Default Value of Next Pointer field (NXTPTR)
PCI Express Capabilities List	PCI Express Capability Structure	0x040	0x0C0
	PCI Power Management Capability Structure	0x0C0	0x0D0
	Message Signaled Interrupt Capability Structure	0x0D0	0x0
	Subsystem ID and Subsystem Vendor ID	0x0F0	0x0
PCI Express Extended Capabilities List	Advanced Error Reporting (AER) Extended Capability	0x100	0x330
	Device Serial Number Extended Capability	0x180	0x0
	PCI Express Virtual Channel Capability (must not be linked)	0x200	0x0
	ACS Extended Capability	0x320	0x0
	Multicast Extended Capability	0x330	0x0

Table 19.8 Default Linkage of Capability Structures for the NT Function When Operating as Function 1 of the Port

DMA Function Registers

This section outlines the configuration space associated the DMA function. These registers are accessible via PCI Express configuration requests to function 2 when the port is configured to operate in the following modes:

- Upstream switch port with DMA function
- NT with DMA function
- Upstream switch port with NT and DMA functions

These registers are not directly accessible by PCI Express configuration request when a port is configured to operate in the following modes:

- Disabled
- Unattached
- Downstream switch port
- Upstream switch port
- NT function
- Upstream switch port with NT function

These registers are always accessible regardless of the port operating mode via the global address space access registers (i.e., GASAADDR and GASADATA located in each switch function), via the SMBus slave interface, or via serial EEPROM. Restrictions apply when using the GASAADDR and GASADATA registers. Refer to the definition of these registers for details. Note that the DMA function allows mapping of its configuration space to memory space using the BAR 0 aperture. Thus, these configuration space registers may be accessed via memory read or writes. See section Base Address Registers on page 15-1 for details.

Table 19.9 lists the capability structures in the DMA function and their default linkage. Figure 19.4 shows the organization of the configuration space.

Capability List	Capability Structure Name	Cfg Space Offset	Default Value of Next Pointer field (NXTPTR)
PCI Express Capabilities List	PCI Express Capability Structure	0x040	0x0C0
	PCI Power Management Capability Structure	0x0C0	0x0D0
	Message Signaled Interrupt Capability Structure	0x0D0	0x0
PCI Express Extended Capa-bilities List	Advanced Error Reporting (AER) Extended Capability	0x100	0x0
	ACS Extended Capability	0x320	0x0

Table 19.9 Default Linkage of Capability Structures for the DMA Function

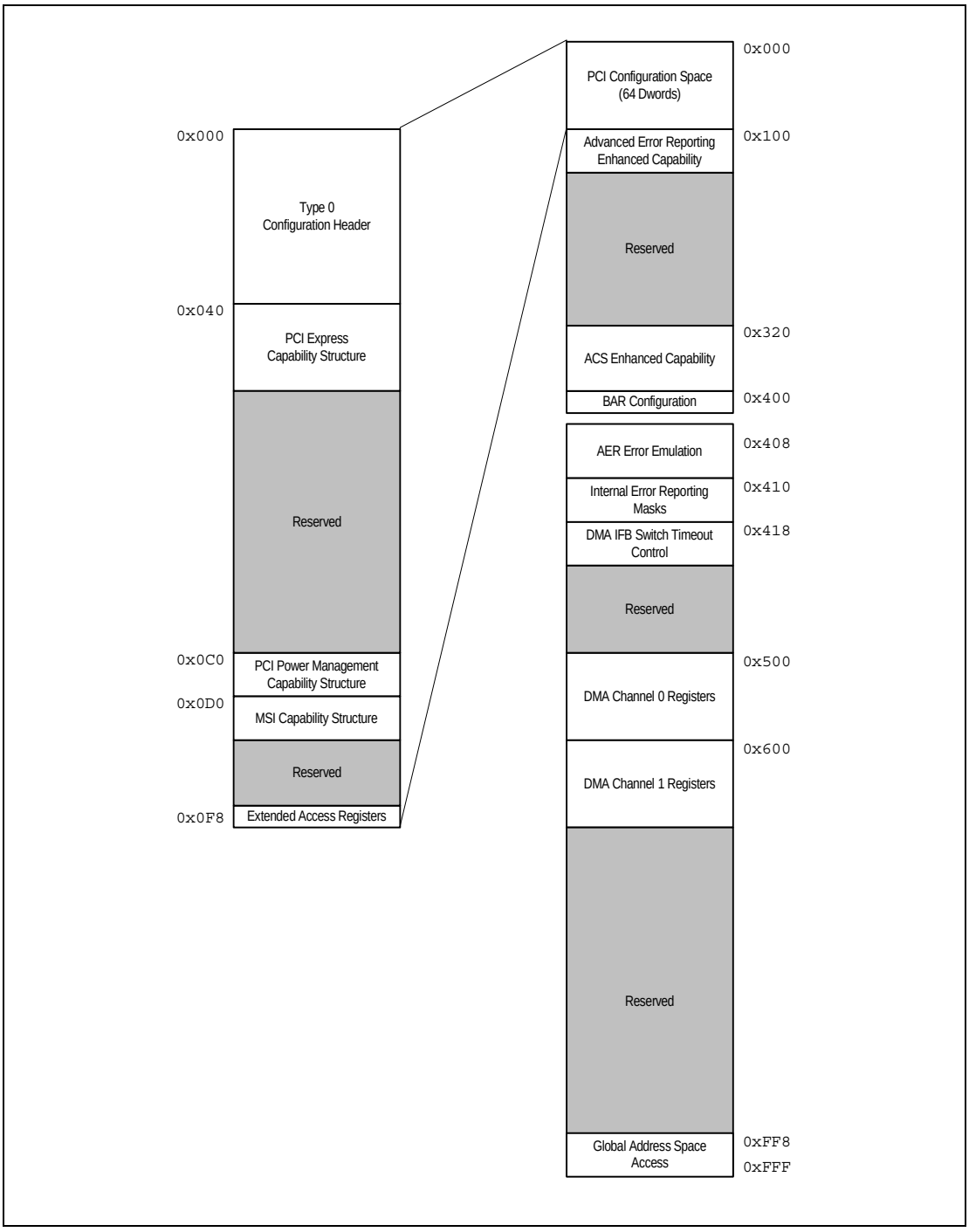

Figure 19.4 DMA Function Configuration Space Organization

Offset addresses for DMA function registers are listed in Table 19.10. Registers in this address range are referenced as PxDMA_REGNAME where x represents the switch's port number and REGNAME represents the register name in Table 19.10. Reading from an address not defined in Table 19.10 returns a value of zero. Writes to an address not defined in Table 19.10 completes successfully, but modifies no data and has no other effect.

In order to facilitate PCI legacy software access to the PCI Express extended configuration space within the NT endpoint's configuration space, the NT endpoint's configuration space contains the Extended Configuration Space Access Address and Data registers (ECFGADDR and ECFGDATA). Refer to the definition of these registers for further details. The ECFGADDR and ECFGDATA registers can't be used to access the global address space access registers (i.e., GASAADDR and GASADATA).

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x000	Word	VID	VID - Vendor Identification (0x000) on page 23-1
0x002	Word	DID	DID - Device Identification (0x002) on page 23-1
0x004	Word	PCICMD	PCICMD - PCI Command (0x004) on page 23-1
0x006	Word	PCISTS	PCISTS - PCI Status (0x006) on page 23-3
0x008	Byte	RID	RID - Revision Identification (0x008) on page 23-4
0x009	3 Bytes	CCODE	CCODE - Class Code (0x009) on page 23-4
0x00C	Byte	CLS	CLS - Cache Line Size (0x00C) on page 23-4
0x00D	Byte	LTIMER	LTIMER - Latency Timer (0x00D) on page 23-4
0x00E	Byte	HDR	HDR - Header Type (0x00E) on page 23-5
0x00F	Byte	BIST	BIST - Built-in Self Test Register (0x00F) on page 23-5
0x010	DWord	BAR0	BAR0 - Base Address Register 0 (0x010) on page 23-5
0x014	DWord	BAR1	BAR1 - Base Address Register 1 (0x014) on page 23-6
0x018	DWord	BAR2	BAR2 - Base Address Register 2 (0x018) on page 23-6
0x01C	DWord	BAR3	BAR3 - Base Address Register 3 (0x01C) on page 23-6
0x020	DWord	BAR4	BAR4 - Base Address Register 4 (0x020) on page 23-6
0x024	DWord	BAR5	BAR5 - Base Address Register 5 (0x024) on page 23-7
0x028	DWord	CCISPTR	CCISPTR - CardBus CIS Pointer (0x028) on page 23-7
0x02C	Word	SUBVID	SUBVID - Subsystem Vendor ID Pointer (0x02C) on page 23-7
0x02E	Word	SUBID	SUBID - Subsystem ID Pointer (0x02E) on page 23-7
0x030	Word	EROMBASE	EROMBASE - Expansion ROM Base (0x030) on page 23-7
0x034	Byte	CAPPTR	CAPPTR - Capabilities Pointer (0x034) on page 23-8
0x03C	Byte	INTRLINE	INTRLINE - Interrupt Line (0x03C) on page 23-8
0x03D	Byte	INTRPIN	INTRPIN - Interrupt PIN (0x03D) on page 23-8
0x03E	Byte	MINGNT	MINGNT - Minimum Grant (0x03E) on page 23-8
0x03F	Byte	MAXLAT	MAXLAT - Maximum Latency (0x03F) on page 23-9
0x040	DWord	PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 23-9
0x044	DWord	PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 23-10
0x048	Word	PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 23-11
0x04A	Word	PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 23-13
0x04C	DWord	PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 23-14
0x050	Word	PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 23-16
0x052	Word	PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 23-17
0x064	DWord	PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 23-18
0x068	Word	PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068) on page 23-19
0x06A	Word	PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 23-20

Table 19.10 DMA Function Registers (Part 1 of 4)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x06C	DWord	PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 23-20
0x070	Word	PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 23-20
0x072	Word	PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 23-21
0x0C0	DWord	PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 23-21
0x0C4	DWord	PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 23-22
0x0D0	DWord	MSICAP	MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) on page 23-23
0x0D4	DWord	MSIADDR	MSIADDR - Message Signaled Interrupt Address (0x0D4) on page 23-24
0x0D8	DWord	MSIUADDR	MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) on page 23-24
0x0DC	DWord	MSIMDATA	MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) on page 23-24
0x0F8	DWord	ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 23-24
0x0FC	DWord	ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 23-25
0x100	DWord	AERCAP	AERCAP - AER Capabilities (0x100) on page 23-26
0x104	DWord	AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 23-26
0x108	DWord	AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 23-27
0x10C	DWord	AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 23-30
0x110	DWord	AERCES	AERCES - AER Correctable Error Status (0x110) on page 23-32
0x114	DWord	AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 23-33
0x118	DWord	AERCTL	AERCTL - AER Control (0x118) on page 23-35
0x11C	DWord	AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 23-36
0x120	DWord	AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 23-36
0x124	DWord	AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 23-36
0x128	DWord	AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 23-36
0x320	DWord	ACSECAPH	ACSECAPH - ACS Extended Capability Header (0x320) on page 23-36
0x324	Word	ACSCAP	ACSCAP - ACS Capability (0x324) on page 23-37
0x326	Word	ACSCTL	ACSCTL - ACS Control (0x326) on page 23-37
0x400	DWord	BARSETUP0	BARSETUP0 - BAR 0 Setup (0x400) on page 23-38
0x408	DWord	DMAUEEM	DMAUEEM - DMA Uncorrectable Error Emulation (0x408) on page 23-39
0x40C	DWord	DMACEEM	DMACEEM - DMA Correctable Error Emulation (0x40C) on page 23-40
0x410	DWord	DMAIERRORMSK0	DMAIERRORMSK0 - Internal Error Reporting Mask 0 (0x410) on page 23-41
0x414	DWord	DMAIERRORMSK1	DMAIERRORMSK1 - Internal Error Reporting Mask 1 (0x414) on page 23-45
0x4FC	DWord	MCRCVINT	MCRCVINT - Multicast Receive Interpretation (0x4FC) on page 23-47
0x500	DWord	DMAC0CTL	DMAC[1:0]CTL - DMA Channel Control (0x500/600) on page 23-48
0x504	DWord	DMAC0CFG	DMAC[1:0]CFG - DMA Channel Configuration (0x504/604) on page 23-48

Table 19.10 DMA Function Registers (Part 2 of 4)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x508	DWord	DMAC0STS	DMAC[1:0]STS - DMA Channel Status (0x508/608) on page 23-50
0x50C	DWord	DMAC0MSK	DMAC[1:0]MSK - DMA Channel Status Mask (0x50C/60C) on page 23-51
0x510	DWord	DMAC0ERRSTS	DMAC[1:0]ERRSTS - DMA Channel Error Status (0x510/610) on page 23-52
0x514	DWord	DMAC0ERRMSK	DMAC[1:0]ERRMSK - DMA Channel Error Mask (0x514/614) on page 23-53
0x518	DWord	DMAC0SSIZE	DMAC[1:0]SSIZE - DMA Channel Stride Size (0x518/618) on page 23-54
0x51C	DWord	DMAC0SSCTL	DMAC[1:0]SSCTL - DMA Channel Source Stride Control (0x51C/61C) on page 23-55
0x520	DWord	DMAC0DSCTL	DMAC[1:0]DSCTL - DMA Channel Destination Stride Control (0x520/620) on page 23-55
0x524	DWord	DMAC0RRCTL	DMAC[1:0]RRCTL - DMA Channel Request Rate Control (0x524/624) on page 23-55
0x528	DWord	DMAC0DPTRL	DMAC[1:0]DPTRL - DMA Channel Descriptor Pointer Low (0x528/628) on page 23-56
0x52C	DWord	DMAC0DPTRH	DMAC[1:0]DPTRH - DMA Channel Descriptor Pointer High (0x52C/62C) on page 23-56
0x530	DWord	DMAC0NDPTRL	DMAC[1:0]NDPTRL - DMA Channel Next Descriptor Pointer Low (0x530/630) on page 23-57
0x534	DWord	DMAC0NDPTRH	DMAC[1:0]NDPTRH - DMA Channel Next Descriptor Pointer High (0x534/634) on page 23-57
0x600	DWord	DMAC1CTL	DMAC[1:0]CTL - DMA Channel Control (0x500/600) on page 23-48
0x604	DWord	DMAC1CFG	DMAC[1:0]CFG - DMA Channel Configuration (0x504/604) on page 23-48
0x608	DWord	DMAC1STS	DMAC[1:0]STS - DMA Channel Status (0x508/608) on page 23-50
0x60C	DWord	DMAC1MSK	DMAC[1:0]MSK - DMA Channel Status Mask (0x50C/60C) on page 23-51
0x610	DWord	DMAC1ERRSTS	DMAC[1:0]ERRSTS - DMA Channel Error Status (0x510/610) on page 23-52
0x614	DWord	DMAC1ERRMSK	DMAC[1:0]ERRMSK - DMA Channel Error Mask (0x514/614) on page 23-53
0x618	DWord	DMAC1SSIZE	DMAC[1:0]SSIZE - DMA Channel Stride Size (0x518/618) on page 23-54
0x61C	DWord	DMAC1SSCTL	DMAC[1:0]SSCTL - DMA Channel Source Stride Control (0x51C/61C) on page 23-55
0x620	DWord	DMAC1DSCTL	DMAC[1:0]DSCTL - DMA Channel Destination Stride Control (0x520/620) on page 23-55
0x624	DWord	DMAC1RRCTL	DMAC[1:0]RRCTL - DMA Channel Request Rate Control (0x524/624) on page 23-55
0x628	DWord	DMAC1DPTRL	DMAC[1:0]DPTRL - DMA Channel Descriptor Pointer Low (0x528/628) on page 23-56
0x62C	DWord	DMAC1DPTRH	DMAC[1:0]DPTRH - DMA Channel Descriptor Pointer High (0x52C/62C) on page 23-56
0x630	DWord	DMAC1NDPTRL	DMAC[1:0]NDPTRL - DMA Channel Next Descriptor Pointer Low (0x530/630) on page 23-57

Table 19.10 DMA Function Registers (Part 3 of 4)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x634	DWord	DMAC1NDPTRH	DMAC[1:0]NDPTRH - DMA Channel Next Descriptor Pointer High (0x534/634) on page 23-57
0xFF8	DWord	GASAADDR	GASAADDR - Global Address Space Access Address (0xFF8) on page 23-57
0xFFC	DWord	GASADATA	GASADATA - Global Address Space Access Data (0xFFC) on page 23-58

Table 19.10 DMA Function Registers (Part 4 of 4)

Switch Configuration and Status Registers

This section outlines switch configuration and status registers. These registers contain control and status bits associated with the operation or configuration of the switch. These registers are accessible using the global address space access registers (i.e., GASAADDR and GASADATA) located in the PCI-to-PCI bridge, NT, or DMA functions, or directly via the SMBus slave interface or serial EEPROM. Figure 19.5 shows the organization of the address space.

Offset addresses for these registers are shown in Table 19.11. Registers in this address range are referenced as `REGNAME` where `REGNAME` represents the register name in Table 19.11. Reading from an address not defined in Table 19.11 returns a value of zero. Writes to an address not defined in Table 19.11 completes successfully, but modifies no data and has no other effect.

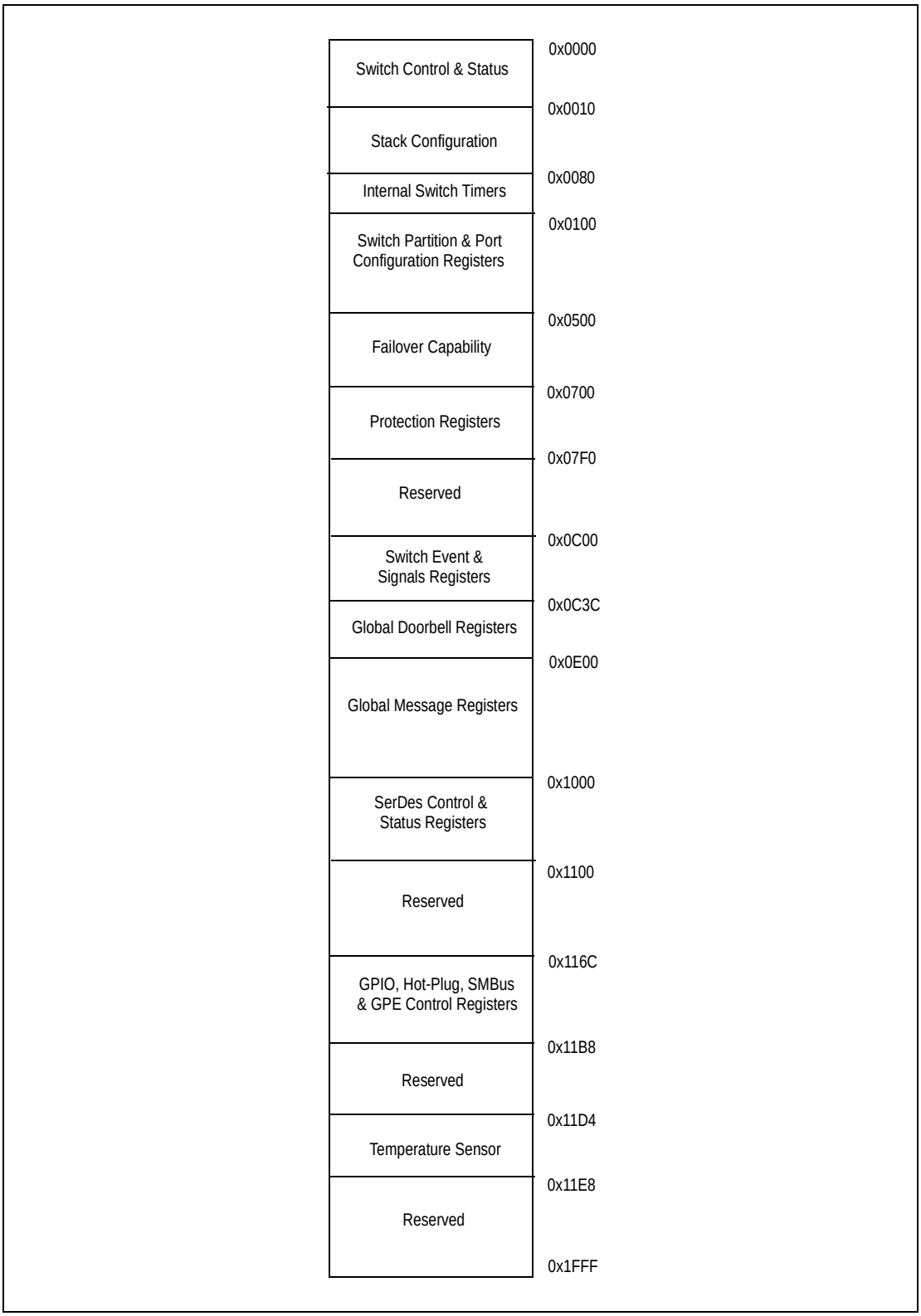

Figure 19.5 Switch Configuration and Status Space Organization

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0000	DWord	SWCTL	SWCTL - Switch Control (0x0000) on page 24-1
0x0004	DWord	BCVSTS	BCVSTS - Boot Configuration Vector Status (0x0004) on page 24-2
0x0008	DWord	PCLKMODE	PCLKMODE - Port Clocking Mode (0x0008) on page 24-3
0x0010	DWord	STK0CFG	STK0CFG - Stack Configuration (0x0010) on page 24-3
0x0014	DWord	STK1CFG	STK1CFG - Stack Configuration (0x0014) on page 24-4
0x0018	DWord	STK2CFG	STK2CFG - Stack Configuration (0x0018) on page 24-4
0x001C	DWord	STK3CFG	STK3CFG - Stack Configuration (0x001C) on page 24-4
0x0080	DWord	RDRAINDELAY	RDRAINDELAY - Reset Drain Delay (0x0080) on page 24-5
0x0084	DWord	POMCDELAY	POMCDELAY - Port Operating Mode Change Drain Delay (0x0084) on page 24-5
0x0088	DWord	SEDELAY	SEDELAY - Side Effect Delay (0x0088) on page 24-6
0x008C	DWord	USSBRDELAY	USSBRDELAY - Upstream Secondary Bus Reset Delay (0x008C) on page 24-6
0x0100	DWord	SWPART0CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x0104	DWord	SWPART0STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x0108	DWord	SWPART0FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x0120	DWord	SWPART1CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x0124	DWord	SWPART1STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x0128	DWord	SWPART1FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x0140	DWord	SWPART2CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x0144	DWord	SWPART2STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x0148	DWord	SWPART2FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x0160	DWord	SWPART3CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x0164	DWord	SWPART3STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x0168	DWord	SWPART3FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x0180	DWord	SWPART4CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x0184	DWord	SWPART4STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x0188	DWord	SWPART4FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x01A0	DWord	SWPART5CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x01A4	DWord	SWPART5STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x01A8	DWord	SWPART5FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x01C0	DWord	SWPART6CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x01C4	DWord	SWPART6STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x01C8	DWord	SWPART6FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9
0x01E0	DWord	SWPART7CTL	SWPART[7:0]CTL - Switch Partition x Control on page 24-7
0x01E4	DWord	SWPART7STS	SWPART[7:0]STS - Switch Partition x Status on page 24-8
0x01E8	DWord	SWPART7FCTL	SWPART[7:0]FCTL - Switch Partition x Failover Control on page 24-9

Table 19.11 Switch Configuration and Status (Part 1 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0200	DWord	SWPORT0CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0204	DWord	SWPORT0STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0208	DWord	SWPORT0FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0240	DWord	SWPORT2CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0244	DWord	SWPORT2STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0248	DWord	SWPORT2FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0280	DWord	SWPORT4CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0284	DWord	SWPORT4STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0288	DWord	SWPORT4FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x02C0	DWord	SWPORT6CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x02C4	DWord	SWPORT6STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x02C8	DWord	SWPORT6FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0300	DWord	SWPORT8CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0304	DWord	SWPORT8STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0308	DWord	SWPORT8FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0380	DWord	SWPORT12CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0384	DWord	SWPORT12STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0388	DWord	SWPORT12FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0400	DWord	SWPORT16CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0404	DWord	SWPORT16STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0408	DWord	SWPORT16FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0480	DWord	SWPORT20CTL	SWPORTxCTL - Switch Port x Control on page 24-9
0x0484	DWord	SWPORT20STS	SWPORTxSTS - Switch Port x Status on page 24-10
0x0488	DWord	SWPORT20FCTL	SWPORTxFCTL - Switch Port x Failover Control on page 24-12
0x0500	DWord	FCAP0CTL	FCAP[3:0]CTL - Failover Capability x Control on page 24-13
0x0504	DWord	FCAP0STS	FCAP[3:0]STS - Failover Capability x Status on page 24-14
0x0508	DWord	FCAP0TIMER	FCAP[3:0]TIMER - Failover Capability x Watchdog Timer on page 24-14
0x0520	DWord	FCAP1CTL	FCAP[3:0]CTL - Failover Capability x Control on page 24-13
0x0524	DWord	FCAP1STS	FCAP[3:0]STS - Failover Capability x Status on page 24-14
0x0528	DWord	FCAP1TIMER	FCAP[3:0]TIMER - Failover Capability x Watchdog Timer on page 24-14
0x0540	DWord	FCAP2CTL	FCAP[3:0]CTL - Failover Capability x Control on page 24-13
0x0544	DWord	FCAP2STS	FCAP[3:0]STS - Failover Capability x Status on page 24-14
0x0548	DWord	FCAP2TIMER	FCAP[3:0]TIMER - Failover Capability x Watchdog Timer on page 24-14
0x0560	DWord	FCAP3CTL	FCAP[3:0]CTL - Failover Capability x Control on page 24-13
0x0564	DWord	FCAP3STS	FCAP[3:0]STS - Failover Capability x Status on page 24-14
0x0568	DWord	FCAP3TIMER	FCAP[3:0]TIMER - Failover Capability x Watchdog Timer on page 24-14

Table 19.11 Switch Configuration and Status (Part 2 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0700	DWord	GASAPROT	GASAPROT - Global Address Space Access Protection (0x0700) on page 24-15
0x0710	DWord	NTMTBLPROT0	GASAPROT - Global Address Space Access Protection (0x0700) on page 24-15
0x0714	DWord	NTMTBLPROT1	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x0718	DWord	NTMTBLPROT2	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x071C	DWord	NTMTBLPROT3	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x0720	DWord	NTMTBLPROT4	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x0724	DWord	NTMTBLPROT5	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x0728	DWord	NTMTBLPROT6	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x072C	DWord	NTMTBLPROT7	NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection on page 24-15
0x0C00	DWord	SESTS	SESTS - Switch Event Status (0x0C00) on page 24-16
0x0C04	DWord	SEMSK	SEMSK - Switch Event Mask (0x0C04) on page 24-17
0x0C08	DWord	SEPMSK	SEPMSK - Switch Event Partition Mask (0x0C08) on page 24-19
0x0C0C	DWord	SELINKUPSTS	SELINKUPSTS - Switch Event Link Up Status (0x0C0C) on page 24-19
0x0C10	DWord	SELINKUPMSK	SELINKUPMSK - Switch Event Link Up Mask (0x0C10) on page 24-19
0x0C14	DWord	SELINKDNSTS	SELINKDNSTS - Switch Event Link Down Status (0x0C14) on page 24-20
0x0C18	DWord	SELINKDNMSK	SELINKDNMSK - Switch Event Link Down Mask (0x0C18) on page 24-20
0x0C1C	DWord	SEFRSTSTS	SEFRSTSTS - Switch Event Fundamental Reset Status (0x0C1C) on page 24-20
0x0C20	DWord	SEFRSTMSK	SEFRSTMSK - Switch Event Fundamental Reset Mask (0x0C20) on page 24-20
0x0C24	DWord	SEHRSTSTS	SEHRSTSTS - Switch Event Hot Reset Status (0x0C24) on page 24-21
0x0C28	DWord	SEHRSTMSK	SEHRSTMSK - Switch Event Hot Reset Mask (0x0C28) on page 24-21
0x0C2C	DWord	SEFOVRMSK	SEFOVRMSK - Switch Event Failover Mask (0x0C2C) on page 24-21
0x0C30	DWord	SEGSIGSTS	SEGSIGSTS - Switch Event Global Signal Status (0x0C30) on page 24-22
0x0C34	DWord	SEGSIGMSK	SEGSIGMSK - Switch Event Global Signal Mask (0x0C34) on page 24-22
0x0C3C	DWord	GDBELLSTS	GDBELLSTS - NT Global Doorbell Status (0x0C3C) on page 24-23
0x0D00	DWord	GODBELLMSK0	GDBELLSTS - NT Global Doorbell Status (0x0C3C) on page 24-23
0x0D04	DWord	GODBELLMSK1	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D08	DWord	GODBELLMSK2	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D0C	DWord	GODBELLMSK3	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D10	DWord	GODBELLMSK4	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D14	DWord	GODBELLMSK5	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D18	DWord	GODBELLMSK6	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D1C	DWord	GODBELLMSK7	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D20	DWord	GODBELLMSK8	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D24	DWord	GODBELLMSK9	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D28	DWord	GODBELLMSK10	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D2C	DWord	GODBELLMSK11	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23

Table 19.11 Switch Configuration and Status (Part 3 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0D30	DWord	GODBELLMSK12	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D34	DWord	GODBELLMSK13	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D38	DWord	GODBELLMSK14	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D3C	DWord	GODBELLMSK15	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D40	DWord	GODBELLMSK16	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D44	DWord	GODBELLMSK17	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D48	DWord	GODBELLMSK18	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D4C	DWord	GODBELLMSK19	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D50	DWord	GODBELLMSK20	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D54	DWord	GODBELLMSK21	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D58	DWord	GODBELLMSK22	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D5C	DWord	GODBELLMSK23	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D60	DWord	GODBELLMSK24	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D64	DWord	GODBELLMSK25	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D68	DWord	GODBELLMSK26	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D6C	DWord	GODBELLMSK27	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D70	DWord	GODBELLMSK28	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D74	DWord	GODBELLMSK29	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D78	DWord	GODBELLMSK30	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D7C	DWord	GODBELLMSK31	GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0] on page 24-23
0x0D80	DWord	GIDBELLMSK0	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D84	DWord	GIDBELLMSK1	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D88	DWord	GIDBELLMSK2	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D8C	DWord	GIDBELLMSK3	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D90	DWord	GIDBELLMSK4	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D94	DWord	GIDBELLMSK5	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D98	DWord	GIDBELLMSK6	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0D9C	DWord	GIDBELLMSK7	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DA0	DWord	GIDBELLMSK8	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DA4	DWord	GIDBELLMSK9	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DA8	DWord	GIDBELLMSK10	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DAC	DWord	GIDBELLMSK11	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DB0	DWord	GIDBELLMSK12	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DB4	DWord	GIDBELLMSK13	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DB8	DWord	GIDBELLMSK14	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DBC	DWord	GIDBELLMSK15	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23

Table 19.11 Switch Configuration and Status (Part 4 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0DC0	DWord	GIDBELLMSK16	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DC4	DWord	GIDBELLMSK17	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DC8	DWord	GIDBELLMSK18	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DCC	DWord	GIDBELLMSK19	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DD0	DWord	GIDBELLMSK20	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DD4	DWord	GIDBELLMSK21	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DD8	DWord	GIDBELLMSK22	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DDC	DWord	GIDBELLMSK23	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DE0	DWord	GIDBELLMSK24	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DE4	DWord	GIDBELLMSK25	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DE8	DWord	GIDBELLMSK26	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DEC	DWord	GIDBELLMSK27	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DF0	DWord	GIDBELLMSK28	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DF4	DWord	GIDBELLMSK29	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DF8	DWord	GIDBELLMSK30	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0DFC	DWord	GIDBELLMSK31	GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0] on page 24-23
0x0E00	DWord	SWP0MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E04	DWord	SWP1MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E08	DWord	SWP2MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E0C	DWord	SWP3MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E10	DWord	SWP4MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E14	DWord	SWP5MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E18	DWord	SWP6MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E1C	DWord	SWP7MSGCTL0	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E20	DWord	SWP0MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E24	DWord	SWP1MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E28	DWord	SWP2MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E2C	DWord	SWP3MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E30	DWord	SWP4MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E34	DWord	SWP5MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E38	DWord	SWP6MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E3C	DWord	SWP7MSGCTL1	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E40	DWord	SWP0MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E44	DWord	SWP1MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E48	DWord	SWP2MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E4C	DWord	SWP3MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24

Table 19.11 Switch Configuration and Status (Part 5 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0E50	DWord	SWP4MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E54	DWord	SWP5MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E58	DWord	SWP6MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E5C	DWord	SWP7MSGCTL2	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E60	DWord	SWP0MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E64	DWord	SWP1MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E68	DWord	SWP2MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E6C	DWord	SWP3MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E70	DWord	SWP4MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E74	DWord	SWP5MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E78	DWord	SWP6MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x0E7C	DWord	SWP7MSGCTL3	SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0] on page 24-24
0x1000	DWord	S0CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x1004	DWord	S0TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x1008	DWord	S0TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x1010	DWord	S0RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x1020	DWord	S1CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x1024	DWord	S1TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x1028	DWord	S1TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x1030	DWord	S1RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x1040	DWord	S2CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x1044	DWord	S2TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x1048	DWord	S2TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x1050	DWord	S2RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x1060	DWord	S3CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x1064	DWord	S3TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x1068	DWord	S3TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x1070	DWord	S3RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x1080	DWord	S4CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x1084	DWord	S4TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x1088	DWord	S4TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x1090	DWord	S4RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x10A0	DWord	S5CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x10A4	DWord	S5TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x10A8	DWord	S5TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x10B0	DWord	S5RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31

Table 19.11 Switch Configuration and Status (Part 6 of 7)

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x10C0	DWord	S6CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x10C4	DWord	S6TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x10C8	DWord	S6TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x10D0	DWord	S6RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x10E0	DWord	S7CTL	S[7:0]CTL- SerDes x Control on page 24-25
0x10E4	DWord	S7TXLCTL0	S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0 on page 24-26
0x10E8	DWord	S7TXLCTL1	S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1 on page 24-28
0x10F0	DWord	S7RXEQLCTL	S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control on page 24-31
0x116C	DWord	GPIOFUNC	GPIOFUNC - General Purpose I/O Function (0x116C) on page 24-31
0x1170	DWord	GPIOAFSEL	GPIOAFSEL - General Purpose I/O Alternate Function Select (0x1170) on page 24-32
0x1174	DWord	GPIOCFG	GPIOCFG - General Purpose I/O Configuration (0x1174) on page 24-32
0x1178	DWord	GPIOD	GPIOD - General Purpose I/O Data (0x1178) on page 24-33
0x117C	DWord	HPCFGCTL	HPCFGCTL - Hot-Plug Configuration Control (0x117C) on page 24-33
0x1188	DWord	SMBUSSTS	SMBUSSTS - SMBus Status (0x1188) on page 24-35
0x118C	DWord	SMBUSCTL	SMBUSCTL - SMBus Control (0x118C) on page 24-37
0x1190	DWord	EEPROMINTF	EEPROMINTF - Serial EEPROM Interface (0x1190) on page 24-40
0x1198	DWord	IOEXPADDR0	IOEXPADDR0 - SMBus I/O Expander Address 0 (0x1198) on page 24-40
0x119C	DWord	IOEXPADDR1	IOEXPADDR1 - SMBus I/O Expander Address 1 (0x119C) on page 24-41
0x11A0	DWord	IOEXPADDR2	IOEXPADDR2 - SMBus I/O Expander Address 2 (0x11A0) on page 24-41
0x11A4	DWord	IOEXPADDR3	IOEXPADDR3 - SMBus I/O Expander Address 3 (0x11A4) on page 24-42
0x11A8	DWord	IOEXPADDR4	IOEXPADDR4 - SMBus I/O Expander Address 4 (0x11A8) on page 24-42
0x11AC	DWord	IOEXPADDR5	IOEXPADDR5 - SMBus I/O Expander Address 5 (0x11AC) on page 24-43
0x11B0	DWord	GPECTL	GPECTL - General Purpose Event Control (0x11B0) on page 24-43
0x11B4	DWord	GPESTS	GPESTS - General Purpose Event Status (0x11B4) on page 24-43
0x11D4	DWord	TMPCTL	TMPCTL - Temperature Sensor Control (0x11D4) on page 24-44
0x11D8	DWord	TMPSTS	TMPSTS - Temperature Sensor Status (0x11D8) on page 24-45
0x11DC	DWord	TMPALARM	TMPALARM - Temperature Sensor Alarm (0x11DC) on page 24-46
0x11E0	DWord	TMPADJ	TMPADJ - Temperature Sensor Adjustment (0x11E0) on page 24-48
0x11E4	DWord	TSSLOPE	TSSLOPE - Temperature Sensor Slope (0x11E4) on page 24-48
0x11E8	DWord	SMBUSCBHL	SMBUSCBHL - SMBus Configuration Block Header Log (0x11E8) on page 24-39

Table 19.11 Switch Configuration and Status (Part 7 of 7)

PCI-to-PCI Bridge Registers

Notes

Type 1 Configuration Header Registers32NT8

VID - Vendor Identification Register (0x000)

Bit Field	Field Name	Type	Default Value	Description
15:0	VID	RO	0x111D	Vendor Identification. This field contains the 16-bit vendor ID value assigned to IDT. See section Vendor ID on page 1-1.

DID - Device Identification Register (0x002)

Bit Field	Field Name	Type	Default Value	Description
15:0	DID	RO	-	**Device Identification.** This field contains the 16-bit device ID assigned by IDT to this bridge. See section Device ID on page 1-1.

Notes

PCICMD - PCI Command Register (0x004)

Bit Field	Field Name	Type	Default Value	Description
0	IOAE	RW	0x0	**I/O Access Enable.** When this bit is cleared, the bridge function does not respond to I/O accesses from the primary bus specified by IOBASE and IOLIMIT. 0x0 - (disable) Disable I/O space. 0x1 - (enable) Enable I/O space.
1	MAE	RW	0x0	**Memory Access Enable.** When this bit is cleared, the bridge function does not respond to memory and prefetchable memory space access from the primary bus specified by MBASE, MLIMIT, PMBASE and PMLIMIT. 0x0 - (disable) Disable memory space. 0x1 - (enable) Enable memory space.
2	BME	RW	0x0	**Bus Master Enable.** When this bit is cleared, the bridge function does not issue requests (e.g., memory, I/O and MSIs since they are in-band writes) on behalf of subordinate devices and handles these as Unsupported Requests (UR). Additionally, the bridge handles non-posted transactions in the upstream direction with a Unsupported Request (UR) completion. This bit does not affect completions in either direction or the forwarding of non memory or I/O requests. 0x0 - (disable) Disable request forwarding. 0x1 - (enable) Enable request forwarding.
3	SSE	RO	0x0	**Special Cycle Enable.** Not applicable.
4	MWI	RO	0x0	**Memory Write Invalidate.** Not applicable.
5	VGAS	RO	0x0	**VGA Palette Snoop.** Not applicable.
6	PERRE	RW	0x0	**Parity Error Response Enable.** This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit (MDPED) in the PCI Status (PCISTS) register.
7	ADSTEP	RO	0x0	**Address Data Stepping.** Not applicable.
8	SERRE	RW	0x0	**SERR Enable.** Non-fatal and fatal errors detected by the bridge are reported to the Root Complex when this bit is set or the bits in the PCI Express Device Control register are set (see PCIEDCTL - PCI Express Device Control (0x048)). In addition, when this bit is set it enables the forwarding of ERR_NONFATAL and ERR_FATAL error messages from the secondary to the primary interface. ERR_CORR messages are unaffected by this bit and are always forwarded. 0x0 - (disable) Disable non-fatal and fatal error reporting if also disabled in Device Control register. 0x1 - (enable) Enable non-fatal and fatal error reporting.

Notes

Bit Field	Field Name	Type	Default Value	Description
9	FB2B	RO	0x0	**Fast Back-to-Back Enable.** Not applicable.
10	INTXD	RW	0x0	**INTx Disable.** Controls the ability of the PCI-to-PCI bridge to generate an INTx interrupt message. When this bit is set, any interrupts generated by this bridge are negated. This may result in a change in the resolved interrupt state of the bridge. This bit has no effect on interrupts forwarded from the secondary to the primary interface.
15:11	Reserved	RO	0x0	Reserved field.

PCISTS - PCI Status Register (0x006)

Bit Field	Field Name	Type	Default Value	Description
2:0	Reserved	RO	0x0	Reserved field.
3	INTS	RO	0x0	**INTx Status.** This bit is set when an INTx interrupt is pending from the function. INTx emulation interrupts forwarded by switch ports from devices downstream of the bridge are not reflected in this bit. For downstream switch ports, this bit is set if an interrupt has been “asserted” by the corresponding port’s hot-plug controller.
4	CAPL	RO	0x1	**Capabilities List.** This bit is hardwired to one to indicate that the bridge implements an extended capability list item.
5	C66MHZ	RO	0x0	**66 MHz Capable.** Not applicable.
6	Reserved	RO	0x0	Reserved field.
7	FB2B	RO	0x0	**Fast Back-to-Back (FB2B).** Not applicable.
8	MDPED	RW1C	0x0	**Master Data Parity Error Detected.** This bit is set by the bridge function if the PERRE bit in the PCI Command register (PCICMD) is set to 0x1 and either of the following two conditions occurs: 1) The function receives a Poisoned Completion going Downstream. 2) The function transmits a Poisoned Request Upstream.
10:9	DEVT	RO	0x0	**DEVSEL# TIming.** Not applicable.
11	STAS	RW1C	0x0	**Signaled Target Abort.** This bit is set when the bridge completes a posted or non-posted request with a completer-abort error[1]. In the switch, this bit is set when the bridge blocks a multicast TLP received on its primary side.

Notes

Bit Field	Field Name	Type	Default Value	Description
12	RTAS	RO	0x0	**Received Target Abort.** Not applicable (the bridge never generates requests on its own behalf).
13	RMAS	RO	0x0	**Received Master Abort.** Not applicable (the bridge never generates requests on its own behalf).
14	SSE	RW1C	0x0	**Signaled System Error.** This bit is set when the bridge sends a ERR_FATAL or ERR_NONFATAL message and the SERR Enable (SERRE) bit is set in the PCICMD register. 0x0 -(noerror) no error. 0x1 - (error) This bit is set when a fatal or non-fatal error is signaled.
15	DPE	RW1C	0x0	**Detected Parity Error.** This bit is set by the bridge whenever it receives a poisoned TLP on the primary side regardless of the state of the PERRE bit in the PCI Command (PCICMD) register.

1. Note that per the PCI Express specification, a "completer" is a component that terminates a "request." A request can be non-posted (e.g., memory read) or posted (e.g., memory write). In the case of a non-posted request, the "completer" also generates a completion. Generally, the function targeted by the request serves as the completer. For cases when an uncorrectable error prevents the request from reaching its targeted function, the function that detects and handles the error serves as the completer.

RID - Revision Identification Register (0x008)

Bit Field	Field Name	Type	Default Value	Description
7:0	RID	RWL	- SWSticky	**Revision ID.** This field contains the revision identification number for the device. See section Revision ID on page 1-1.

CCODE - Class Code Register (0x009)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTF	RO	0x00	**Interface.** This value indicates that the device is a PCI-to-PCI bridge that does not support subtractive decode.
15:8	SUB	RO	0x04	**Sub Class Code.** This value indicates that the device is a PCI-to-PCI bridge.
23:16	BASE	RO	0x06	**Base Class Code.** This value indicates that the device is a bridge.

Notes

CLS - Cache Line Size Register (0x00C)

Bit Field	Field Name	Type	Default Value	Description
7:0	CLS	RW	0x00	**Cache Line Size.** This field has no effect on the bridge's functionality but may be read and written by software. This field is implemented for compatibility with legacy software.

PLTIMER - Primary Latency Timer (0x00D)

Bit Field	Field Name	Type	Default Value	Description
7:0	PLTIMER	RO	0x00	**Primary Latency Timer.** Not applicable.

HDR - Header Type Register (0x00E)

Bit Field	Field Name	Type	Default Value	Description
7:0	HDR	RO	See Description	**Header Type.** This field indicates the configuration space header type for the bridge function (type 1 header). The default value depends on the port's operating mode. If the port operating mode configures the port as a multi-function device, the value of this field is 0x81. Otherwise, the value of this field is 0x01.

BIST - Built-in Self Test Register (0x00F)

Bit Field	Field Name	Type	Default Value	Description
7:0	BIST	RO	0x0	**BIST.** This value indicates that the bridge function does not implement BIST.

BAR0 - Base Address Register 0 (0x010)

Bit Field	Field Name	Type	Default Value	Description
31:0	BAR	RO	0x0	**Base Address Register.** Not applicable.

Notes

BAR1 - Base Address Register (0x014)

Bit Field	Field Name	Type	Default Value	Description
31:0	BAR	RO	0x0	**Base Address Register.** Not applicable.

PBUSN - Primary Bus Number Register (0x018)

Bit Field	Field Name	Type	Default Value	Description
7:0	PBUSN	RW	0x0	**Primary Bus Number.** This field is used to record the bus number of the PCI bus segment to which the primary interface of the bridge is connected. This field has no functional effect within the switch but is implemented as a read/write register for software compatibility

SBUSN - Secondary Bus Number Register (0x019)

r

Bit Field	Field Name	Type	Default Value	Description
7:0	SBUSN	RW	0x0	**Secondary Bus Number.** This field is used to record the bus number of the PCI bus segment to which the secondary interface of the bridge is connected.

SUBUSN - Subordinate Bus Number Register (0x01A)

Bit Field	Field Name	Type	Default Value	Description
7:0	SUBUSN	RW	0x0	**Subordinate Bus Number.** The Subordinate Bus Number register is used to record the bus number of the highest numbered PCI bus segment which is behind (or subordinate to) the bridge.

SLTIMER - Secondary Latency Timer Register (0x01B)

Bit Field	Field Name	Type	Default Value	Description
7:0	SLTIMER	RO	0x0	**Secondary Latency Timer.** Not applicable.

Notes

IOBASE - I/O Base Register (0x01C)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RWL	0x1 SWSticky	**I/O Capability.** Indicates if the bridge supports 16-bit or 32-bit I/O addressing. 0x0 - (io16) 16-bit I/O addressing. 0x1 - (io32) 32-bit I/O addressing.
3:1	Reserved	RO	0x0	Reserved field.
7:4	IOBASE	RW	0xF	**I/O Base.** The IOBASE and IOLIMIT registers are used to control the forwarding of I/O transactions between the primary and secondary interfaces of the bridge. This field contains A[15:12] of the lowest I/O address aligned on a 4 KB boundary that is below the primary interface of the bridge.

IOLIMIT - I/O Limit Register (0x01D)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RO	0x1	**I/O Capability.** Indicates if the bridge supports 16-bit or 32-bit I/O addressing. This bit always reflects the value of the IOCAP field in the IOBASE register.
3:1	Reserved	RO	0x0	Reserved field.
7:4	IOLIMIT	RW	0x0	**I/O Limit.** The IOBASE and IOLIMIT registers are used to control the forwarding of I/O transactions between the primary and secondary interfaces of the bridge. This field contains A[15:12] of the highest I/O address, with A[11:0] assumed to be 0xFFF, that is below the primary interface of the bridge.

SECSTS - Secondary Status Register (0x01E)

Bit Field	Field Name	Type	Default Value	Description
7:0	Reserved	RO	0x0	Reserved field.
8	MDPED	RW1C	0x0	**Master Data Parity Error.** This bit is set by the bridge function if the PERRE bit in the Bridge Control (BCTL) register is set to 0x1 and either of the following two conditions occurs: 1) The function receives a Poisoned Completion going Upstream. 2) The function transmits a Poisoned Request Downstream.
10:9	DVSEL	RO	0x0	Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
11	STAS	RW1C	0x0	**Signaled Target Abort Status.** This bit is set when the bridge completes a posted or non-posted request with a completer-abort error on its secondary side[1]. In the switch, this bit is set when the bridge completes a posted or non-posted request received on its secondary side with completer-abort status as a result of an ACS violation, or when the bridge blocks a multicast TLP received on its secondary side.
12	RTAS	RO	0x0	**Received Target Abort Status.** Not applicable (the bridge never generates requests on its own behalf).
13	RMAS	RO	0x0	**Received Master Abort Status.** Not applicable (the bridge never generates requests on its own behalf).
14	RSE	RW1C	0x0	**Received System Error.** This bit is set if the secondary side of the bridge receives an ERR_FATAL or ERR_NONFATAL message.
15	DPE	RW1C	0x0	**Detected Parity Error.** This bit is set by the bridge whenever it receives a poisoned TLP on the secondary side regardless of the state of the PERRE bit in the PCI Command register

1. Note that per the PCI Express specification, a "completer" is a component that terminates a "request." A request can be non-posted (e.g., memory read) or posted (e.g., memory write). In the case of a non-posted request, the "completer" also generates a completion. Generally, the function targeted by the request serves as the completer. For cases when an uncorrectable error prevents the request from reaching its targeted function, the function that detects and handles the error serves as the completer.

MBASE - Memory Base Register (0x020)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
15:4	MBASE	RW	0xFFF	**Memory Address Base.** The MBASE and MLIMIT registers are used to control the forwarding of non-prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the lowest address aligned on a 1MB boundary that is below the primary interface of the bridge.

MLIMIT - Memory Limit Register (0x022)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
15:4	MLIMIT	RW	0x0	**Memory Address Limit.** The MBASE and MLIMIT registers are used to control the forwarding of non-prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the highest address, with A[19:0] assumed to be 0xF_FFFF, that is below the primary interface of the bridge.

Notes

PMBASE - Prefetchable Memory Base Register (0x024)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RWL	0x1 SWSticky	**Prefetchable Memory Capability.** Indicates if the bridge supports 32-bit or 64-bit prefetchable memory addressing. 0x0 - (prefmem32) 32-bit prefetchable memory addressing. 0x1 - (prefmem64) 64-bit prefetchable memory addressing.
3:1	Reserved	RO	0x0	Reserved field.
15:4	PMBASE	RW	0xFFF	**Prefetchable Memory Address Base.** The PMBASE, PMBASEU, PMLIMIT and PMLIMITU registers are used to control the forwarding of prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the lowest memory address aligned on a 1MB boundary that is below the primary interface of the bridge. PMBASEU specifies the remaining bits.

PMLIMIT - Prefetchable Memory Limit Register (0x026)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RO	0x1	**Prefetchable Memory Capability.** Indicates if the bridge supports 32-bit or 64-bit prefetchable memory addressing. This bit always reflects the value in the PMCAP field in the PMBASE register.
3:1	Reserved	RO	0x0	Reserved field.
15:4	PMLIMIT	RW	0x0	**Prefetchable Memory Address Limit.** The PMBASE, PMBASEU, PMLIMIT and PMLIMITU registers are used to control the forwarding of prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the highest memory address, with A[19:0] assumed to be 0xF_FFFF, that is below the primary interface of the bridge. PMLIMITU specifies the remaining bits

PMBASEU - Prefetchable Memory Base Upper Register (0x028)

Bit Field	Field Name	Type	Default Value	Description
31:0	PMBASEU	RW	0xFFFF_FFFF	**Prefetchable Memory Address Base Upper.** This field specifies the upper 32-bits of PMBASE when 64-bit addressing is used. When the PMCAP field in the PMBASE register is cleared, this field becomes read-only with a value of zero.

Notes

PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C)

Bit Field	Field Name	Type	Default Value	Description
31:0	PMLIMITU	RW	0x0	**Prefetchable Memory Address Limit Upper.** This field specifies the upper 32-bits of PMLIMIT. When the PMCAP field in the PMBASE register is cleared, this field becomes read-only with a value of zero.

IOBASEU - I/O Base Upper Register (0x030)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOBASEU	RW	0xFFFF	**I/O Address Base Upper.** This field specifies the upper 16-bits of IOBASE. When the IOCAP field in the IOBASE register is cleared, this field becomes read-only with a value of zero.

IOLIMITU - I/O Limit Upper Register (0x032)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOLIMITU	RW	0x0	**IO Limit Upper.** This field specifies the upper 16-bits of IOLIMIT. When the IOCAP field in the IOBASE register is cleared, this field becomes read-only with a value of zero.

CAPPTR - Capabilities Pointer Register (0x034)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPPTR	RWL	0x40 SWSticky	**Capabilities Pointer.** This field specifies a pointer to the head of the capabilities structure.

EROMBASE - Expansion ROM Base Address Register (0x038)

Bit Field	Field Name	Type	Default Value	Description
31:0	EROMBASE	RO	0x0	**Expansion ROM Base Address.** This function does not implement an expansion ROM. Thus, this field is hardwired to zero.

Notes

INTRLINE - Interrupt Line Register (0x03C)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRLINE	RW	0x0	**Interrupt Line.** This register communicates interrupt line routing information. Values in this register are programmed by system software and are system architecture specific. This function does not use the value in this register. Legacy interrupts may be implemented by downstream switch ports.

INTRPIN - Interrupt PIN Register (0x03D)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRPIN	RWL	0x0 SWSticky	**Interrupt Pin.** Interrupt pin or legacy interrupt messages are not used by the PCI-to-PCI bridge function by default. However, they can be used for hot-plug by the downstream switch ports, or for switch events and signals by the upstream port. The switch's PCI-to-PCI bridges may only be configured to generate INTA interrupts. Therefore, correct values for this field are only 0x0 and 0x1. 0x0 - (none) Bridge does not generate any INTx interrupts. 0x1 - (INTA) Bridge generates INTA interrupts. 0x2 - Reserved 0x3 - Reserved 0x4 - Reserved Programming this field to 0x0 in effect disables interrupt generation.

Notes

BCTL - Bridge Control Register (0x03E)

Bit Field	Field Name	Type	Default Value	Description
0	PERRE	RW	0x0	**Parity Error Response Enable.** This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit (MDPED) in the Secondary Status (SECSTS) register.
1	SERRE	RW	0x0	**System Error Enable.** This bit controls forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL from the secondary interface of the bridge to the primary interface. Note that error reporting must be enabled in the Command register or PCI Express Capability structure, Device Control register for errors to be reported on the primary interface. 0x0 - (ignore) Do not forward errors from the secondary to the primary interface. 0x1 - (report) Enable forwarding of errors from secondary to the primary interface.
2	ISAEN	RW	0x0	**ISA Enable.** This bit controls the routing of ISA I/O transactions. 0 - (disable) Forward downstream all I/O addresses in the address range defined by the I/O base and I/O limit registers 1 - (enable) Forward upstream ISA I/O addresses in the address range defined by the I/O base and I/O limit registers that are in the first 64 KB of PCI I/O address space (top 768 bytes of each 1-KB block)
3	VGAEN	RW	0x0	**VGA Enable.** Controls the routing of processor-initiated transactions targeting VGA. 0 - (block) Do not forward VGA compatible addresses from the primary interface to the secondary interface 1 - (forward) Forward VGA compatible addresses from the primary to the secondary interface.
4	VGA16EN	RW	0x0	**VGA 16-bit Enable.** This bit only has an effect when the VGAEN bit is set in this register. This read/write bit enables system configuration software to select between 10-bit and 16-bit I/O space decoding for VGA transactions. 0 - (bit10) Perform 10-bit decoding. I/O space aliasing occurs in this mode. 1 - (bit16) Perform 16-bit decoding. No I/O space aliasing occurs in this mode.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	Reserved	RO	0x0	Reserved field.
6	SRESET	RW	0x0	**Secondary Bus Reset.** Setting this bit triggers a secondary bus reset. In the upstream port, setting this bit initiates a Upstream Secondary Bus Reset. In a downstream switch port, setting this bit initiates a Downstream Secondary Bus Reset. Port Configuration Registers must not be changed except as required to update port status.
15:7	Reserved	RO	0x0	Reserved field.

PCI Express Capability Structure

PCIECAP - PCI Express Capability (0x040)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x10	**Capability ID.** The value of 0x10 identifies this capability as a PCI Express capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
19:16	VER	RWL	0x2 SWSticky	**PCI Express Capability Version.** This field indicates the PCI-SIG defined PCI Express capability structure version number.
23:20	TYPE	RO	Upstream Port: 0x5 Downstream Switch Port: 0x6 MSWSticky	**Port Type.** This field identifies the type of switch port (upstream or downstream).
24	SLOT	RWL	0x0 SWSticky	**Slot Implemented.** This bit is set when the PCI Express link associated with this Port is connected to a slot. This field does not apply to an upstream port and should be set to zero.
29:25	IMN	RO	0x0	**Interrupt Message Number.** The function is allocated only one MSI. Therefore, this field is set to zero.
31:30	Reserved	RO	0x0	Reserved field.

Notes

PCIEDCAP - PCI Express Device Capabilities (0x044)

Bit Field	Field Name	Type	Default Value	Description
2:0	MPAYLOAD	RWL	HWINIT (See description) MSWSticky	**Maximum Payload Size Supported.** This field indicates the maximum payload size that the device can support for TLPs. The default value of this field is automatically set by the hardware based on the port's maximum link width as determined by the stack's configuration. If a port has a maximum link width of x1, the default value of this field is 0x3. Otherwise, the default value of this field is 0x4. 0x0 - (s128) 128 bytes max payload size 0x1 - (s256) 256 bytes max payload size 0x2 - (s512) 512 bytes max payload size 0x3 - (s1024) 1024 bytes max payload size 0x4 - (s2048) 2048 bytes max payload size 0x5 - Not supported 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
4:3	PFS	RO	0x0	**Phantom Functions Supported.** This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with the TLP's tag identifier. The value is hardwired to 0x0 to indicate that no function number bits are used for phantom functions.
5	ETAG	RWL	0x1 SWSticky	**Extended Tag Field Support.** This field indicates the maximum supported size of the Tag field as a requester. 0x0 - 5-bit Tag field supported 0x1 - 8-bit Tag field supported
8:6	E0AL	RO	0x0	**Endpoint L0s Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L0s state to the L0 state. The value is hardwired to 0x0 as this field is only applicable to endpoint functions.
11:9	E1AL	RO	0x0	**Endpoint L1 Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L1 state to the L0 state. The value is hardwired to 0x0 as this field is only applicable to endpoint functions.
12	ABP	RO	0x0	**Attention Button Present.** In PCI Express Specification 1.0a, when set, this bit indicates that an Attention Button is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
13	AIP	RO	0x0	**Attention Indicator Present.** In PCI Express base 1.0a when set, this bit indicates that an Attention Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	PIP	RO	0x0	**Power Indicator Present.** In PCI Express Specification 1.0a when set, this bit indicates that a Power Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification.
15	RBERR	RO	0x1	**Role Based Error Reporting.** This bit is set to indicate that this function supports error reporting as defined in the PCI Express Base Specification.
17:16	Reserved	RO	0x0	Reserved field.
25:18	CSPLV	RO	0x0	**Captured Slot Power Limit Value.** This field in combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. The value of this field is set by a Set_Slot_Power_Limit Message and is only applicable for an upstream port.[1] A port in unattached mode does not modify this field as a result of receiving a Set_Slot_Power_Limit Message. This field is always zero in unattached or downstream switch ports.
27:26	CSPLS	RO	0x0	**Captured Slot Power Limit Scale.** This field specifies the scale used for the Slot Power Limit Value. The value of this field is set by a Set_Slot_Power_Limit Message and is only applicable for an upstream port. 0 - (v1) 1.0x 1 - (v1p1) 0.1x 2 - (v0p01) 0.01x 3 - (v0p001x) 0.001x A port in unattached mode does not modify this field as a result of receiving a Set_Slot_Power_Limit Message. This field is always zero in unattached or downstream switch ports.
28	FLR	RO	0x0	**Function Level Reset.** Not applicable to PCI-to-PCI bridge functions.
31:29	Reserved	RO	0x0	Reserved field.

1. NOTE: Set_Slot_Power_Limit messages received by a port implicitly target all functions in the port.

Notes

PCIEDCTL - PCI Express Device Control (0x048)

Bit Field	Field Name	Type	Default Value	Description
0	CEREN	RW	0x0	**Correctable Error Reporting Enable.** This bit controls reporting of correctable errors by this function.
1	NFEREN	RW	0x0	**Non-Fatal Error Reporting Enable.** This bit controls reporting of non-fatal errors by this function.
2	FEREN	RW	0x0	**Fatal Error Reporting Enable.** This bit controls reporting of fatal errors by this function.
3	URREN	RW	0x0	**Unsupported Request Reporting Enable.** This bit controls reporting of unsupported requests by this function.
4	ERO	RO	0x0	**Enable Relaxed Ordering.** Not applicable. The bridge function does not initiate requests other than message requests. Message requests always have the relaxed-ordering bit in the attributes field is always set to 0b0.
7:5	MPS	RW	0x0	**Max Payload Size.** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as large as the set value. As a transmitter, the function must not generate TLPs exceeding the set value. This field should be set to a value less than that advertised by the Maximum Payload Size Supported (MPAYLOAD) field in the PCI Express Device Capabilities (PCIEDCAP) register. Setting this field to a value larger than that advertised in the MPAYLOAD field produces undefined results. Programming of this field is subject to the restrictions outlined in section Maximum Payload Size on page 10-2 and section Maximum Payload Size on page 14-22. 0x0 -(s128) 128 bytes max payload size 0x1 -(s256) 256 bytes max payload size 0x2 -(s512) 512 bytes max payload size 0x3 -(s1024) 1024 bytes max payload size 0x4 -(s2048) 2048 bytes max payload size 0x5 -(s4096) 4096 bytes max payload size 0x6 -reserved (treated as 128 bytes) 0x7 -reserved (treated as 128 bytes)
8	ETFEN	RW	0x0	**Extended Tag Field Enable.** Since the bridge never generates a transaction that requires a completion, this bit has no functional effect on the device during normal operation. To aid in debug, when the SEQTAG field is set in the TLCTL register, this field controls whether tags are generated in the range from 0 through 31 or from 0 through 255.
9	PFEN	RO	0x0	**Phantom Function Enable.** This function does not support phantom function numbers. Therefore, this field is hardwired to zero.
10	AUXPMEN	RO	0x0	**Auxiliary Power PM Enable.** GPIO_ does not implement this capability.

Notes

Bit Field	Field Name	Type	Default Value	Description
11	ENS	RO	0x0	**Enable No Snoop.** Not applicable. The bridge function does not generate transactions with the No Snoop bit set and passes transactions through the bridge with the No Snoop bit unmodified.
14:12	MRRS	RO	0x0	**Maximum Read Request Size.** The bridge function does not generate read requests and passes transactions through the bridge with the size unmodified. Therefore, this field has no functional effect on the behavior of the bridge.
15	Reserved	RO	0x0	Reserved field.

PCIEDSTS - PCI Express Device Status (0x04A)

Bit Field	Field Name	Type	Default Value	Description
0	CED	RW1C	0x0	**Correctable Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
1	NFED	RW1C	0x0	**Non-Fatal Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
2	FED	RW1C	0x0	**Fatal Error Detected.** This bit indicates the status of fatal errors detected by this function. Errors are logged in this registers regardless of whether error reporting is enabled or not.
3	URD	RW1C	0x0	**Unsupported Request Detected.** This bit indicates the function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not.
4	AUXPD	RO	0x0	**Aux Power Detected.** Devices that require AUX power, set this bit when AUX power is detected.This device does not require AUX power, hence the value is hardwired to zero.
5	TP	RO	0x0	**Transactions Pending.** This function does not issue Non-Posted Requests on its own behalf. Therefore, this field is hardwired to zero.
15:6	Reserved	RO	0x0	Reserved field.

Notes

PCIELCAP - PCI Express Link Capabilities (0x04C)

Bit Field	Field Name	Type	Default Value	Description
3:0	MAXLNKSPD	RWL	0x2 SWSticky	**Maximum Link Speed.** This field indicates the supported link speeds of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved Note: This device advertises support for 5 GT/s regardless of the setting of this field. Modifying this field has no effect on the hardware
9:4	MAXLNK-WDTH	RWL	HWINIT (see descrip-tion) MSWSticky	**Maximum Link Width.** This field indicates the maximum link width of the given PCI Express link. This field may be overridden to allow the link width to be forced to a smaller value. When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode). Violating this rule produces undefined results. Setting this field to an invalid or reserved value is allowed, and results in the port operating at it's default value. The default value of this field is automatically set by the hardware as described in section Port Maximum Link Width on page 7-2. 0 - reserved 1 - (x1) x1 link width 2 - (x2) x2 link width 4 - (x4) x4 link width 8 - (x8) x8 link width 12 - (x12) x12 link width 16 - (x16) x16 link width 32-(x32) x32 link width others - reserved Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any stack configuration change.
11:10	ASPMS	RWL	0x3 SWSticky	**Active State Power Management (ASPM) Support.** This default value of this field is 0x3 to indicate that L0s and L1 are supported. This field may be overridden to allow user control over the ASPM capabilities of this port (L0s and/or L1). When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
14:12	L0SEL	RWL	0x6 SWSticky	**L0s Exit Latency.** This field indicates the L0s exit latency for the given PCI Express link. Transitioning from L0s to L0 always requires approximately 2.04 µs. Thus, default value indicates an L0s exit latency between 2 µs and 4 µs. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).

Notes

Bit Field	Field Name	Type	Default Value	Description
17:15	L1EL	RWL	0x2 SWSticky	**L1 Exit Latency.** This field indicates the L1 exit latency for the given PCI Express link. Transitioning from L1 to L0 always requires approximately 2.3 µS. Therefore, a value 2 µs to less than 4 µs is reported with a default value of 0x2. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
18	CPM	RWL	0x0 SWSticky	**Clock Power Management.** This bit indicates if the component tolerates removal of the reference clock via the "CLKREQ#" machanism. The switch does not support the removal of reference clocks.
19	SDERR	RWL	Upstream Port: 0x0 Downstream Switch Port: 0x1 MSWSticky	**Surprise Down Error Reporting.** Downstream switch ports support surprise down error reporting. This field does not apply to an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
20	DLLLA	RWL	Upstream Port: 0x0 Downstream Switch Port: 0x1 MSWSticky	**Data Link Layer Link Active Reporting.** Downstream switch ports support the capability of reporting the DL_Active state of the data link control and management state machine. This field does not apply to an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
21	LBN	RWL	Upstream Port: 0x0 Downstream Switch Port: 0x1 MSWSticky	**Link Bandwidth Notification Capability.** When set, this bit indicates support for the link bandwidth notification status and interrupt mechanisms. Downstream switch ports support the capability. This field does not apply to an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
23:22	Reserved	RO	0x0	Reserved field.
31:24	PORTNUM	RO	Port 0: 0x0 Port 2: 0x2 Port 4: 0x4 Port 6: 0x6 Port 8: 0x8 Port 12: 0xC Port 16: 0x10 Port 20: 0x14	**Port Number.** This field indicates the PCI Express port number for the corresponding link.

Notes

PCIELCTL - PCI Express Link Control (0x050)

Bit Field	Field Name	Type	Default Value	Description
1:0	ASPM	RW	0x0	**Active State Power Management (ASPM) Control.** This field controls the level of ASPM supported by the link. The initial value corresponds to disabled. 0x0 - (disabled) disabled 0x1 - (l0s) L0s enable entry 0x2 - (l1) L1 enable entry 0x3 - (l0sl1) L0s and L1 enable entry Note that "L0s enable entry" corresponds to the transmitter entering L0s (the receiver supports this function and is not affected by this setting). When a port operates in a multi-function mode, only capabilities enabled in all functions of the port are enabled for the port as a whole (e.g., L0s is enabled for the port when all functions of the port have L0s enabled in this field). It is recommended, though not required, that software program the same value in this field for all functions of the port.
2	Reserved	RO	0x0	Reserved field.
3	RCB	RO	0x0	**Read Completion Boundary.** This field is not applicable and is hardwired to zero.
4	LDIS	Upstream Port: RO Downstream Switch Port: RW	0x0	**Link Disable.** When set in a downstream switch port, this bit disables the link. Writes to this bit are immediately reflected in the value of the bit, regardless of the actual link state. This bit is not applicable for an upstream port and is hardwired to zero.
5	LRET	Upstream Port: RWL Downstream Switch Port: RW	0x0	**Link Retrain.** Writing a one to this field initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. This field always returns zero when read. It is permitted to set this bit while simultaneously modifying other fields in this register. When this bit is set and the LTSSM is already in the Recovery or Configuration states, all modifications that affect link retraining are applied in the subsequent retraining. Else, if the LTSSM is not in the Recovery or Configuration states, modifications that affect link retraining are applied immediately. For compliance with the PCI Express Base Specification, this bit has no effect on the upstream port when the REGUNLOCK bit is cleared in the SWCTL register. In this mode the field is hardwired to zero. When the REGUNLOCK bit is set, writing a one to the LRET bit initiates link retraining on the upstream port after a programmed delay. The switch always returns a completion for the request that set this bit, before the effect of this bit is applied.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	CCLK	RW	0x0	**Common Clock Configuration.** When set, this bit indicates that this port and the port at the opposite end of the link are operating with a distributed common reference clock. When a port operates in a multi-function mode, software must set this bit identically for all functions of the port. Otherwise, the port assumes that it is not operating with a distributed common reference clock. After modifying this bit in both components of the link, software must trigger a link retrain by setting the link retrain bit in the upstream component's Link Control register. In the switch, the L0s and L1 exit latencies do not change among common and non-common clock configurations.
7	ESYNC	RW	0x0	**Extended Sync.** When set this bit forces transmission of additional ordered sets when exiting the L0s state and when in the recovery state. When a port operates in a multi-function mode, the effect of this bit is applied when this bit is set in any of the port's functions.
8	CLKPWRMGT	RO	0x0	**Enable Clock Power Management.** The switch does not support this feature.
9	HAWD	RO	0x0	**Hardware Autonomous Width Disable.** Switch ports do not have a hardware autonomous mechanism to change link width, except due to link reliability issues. Therefore, this bit is not applicable and is hardwired to zero.
10	LBWINTEN	Upstream Port: RO Downstream Switch Port: RW	0x0	**Link Bandwidth Management Interrupt Enable.** When set, this bit enables the generation of an interrupt to indicate that the LBWSTS bit has been set in the PCIELSTS register. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is not applicable to an upstream port and is hardwired to zero.
11	LABWINTEN	Upstream Port: RO Downstream Switch Port: RW	0x0	**Link Autonomous Bandwidth Interrupt Enable.** When set, this bit enables the generation of an interrupt to indicate that the LABWSTS bit has been set in the PCIELSTS register. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is not applicable to an upstream port and is hardwired to zero.
15:12	Reserved	RO	0x0	Reserved field.

Notes

PCIELSTS - PCI Express Link Status (0x052)

Bit Field	Field Name	Type	Default Value	Description
3:0	CLS	RO	0x1	**Current Link Speed.** This field indicates the current link speed of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved
9:4	NLW	RO	HWINIT	**Negotiated Link Width.** This field indicates the negotiated width of the link. 00 0001b - x1 00 0010b - x2 00 0100b - x4 00 1000b - x8 00 1100b - x12 01 0000b - x16 10 0000b - x32 When the MAXLNKWDTH field in the PCIELCAP register selects a width not supported by the port, the value of this field corresponds to the setting of the MAXLNKWDTH field, regardless of the actual negotiated link width. When the MAXLNKWDTH field in the PCIELCAP register selects a width supported by the port, but the link is unable to train, the value in this field is set to 0x0. When the port operates in a multi-function mode, the above rules are based on the MAXLNKWDTH field for function 0 of the port. Note that software must ensure that all functions of the port have identical MAXLNKWDTH field values.
10	Reserved	RO	0x0	
11	LTRAIN	RO	0x0	**Link Training.** When set, this bit indicates that link training is in progress. This bit is set when the Physical Layer LTSSM is in Configuration or Recovery state, or when 0x1 is written to LRET bit in the PCIELCTL register but link training has not yet begun. Hardware clears this bit when LTSSM exits Configuration/Recovery state. This bit is only valid for a downstream switch port. For an upstream port, this bit always has a value of 0x0.
12	SCLK	RWL	HWINIT SWSticky	**Slot Clock Configuration.** When set, this bit indicates that the port uses the same physical reference clock used by its link partner (i.e., common-clock configuration). The initial value of this field depends on the port's clocking mode. Refer to Table 2.4 for further details. When the port operates in a multi-function mode, this field reports the same value for all functions of the port.

Notes

Bit Field	Field Name	Type	Default Value	Description
13	DLLLA	RO	0x0	**Data Link Layer Link Active.** This bit indicates the status for the data link control and management state machine. 0x0 - (not_active) Data link layer not active state 0x1 - (active) Data link layer active state This bit is never be set by hardware if the DLLLA bit in the PCIELCAP register is cleared. This field is hardwired to zero in an upstream port.
14	LBWSTS	Upstream Port: RO Downstream Switch Port: RW1C	0x0	**Link Bandwidth Management Status.** This bit is set to indicate that either of the following have occurred without the link transitioning through the DL_Down state. A link retraining initiated by setting the LRET bit in the PCIELCTL register has completed. Note that this bit is set following any write of 0x1 to the LRET bit, even if the link was in the process of retraining for some other reason. The physical layer has autonomously changed link speed or width to attempt to correct unreliable link operation either through an LTSSM time-out or a higher level process. The physical layer reports a speed or width change was initiated by the downstream component that was not indicated as an autonomous change. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is hardwired to zero in an upstream port.
15	LABWSTS	Upstream Port: RO Downstream Switch Port: RW1C	0x0	**Link Autonomous Bandwidth Status.** This bit is set to indicate that either that the physical layer has autonomously changed link speed or width for reasons other than to attempt to correct unreliable link operation. This bit must be set if the physical layer reports a speed or width change was initiated by the downstream component that was indicated as an autonomous change. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is hardwired to zero in an upstream port.

Notes

PCIESCAP - PCI Express Slot Capabilities (0x054)

Bit Field	Field Name	Type	Default Value	Description
0	ABP	RWL	0x0 SWSticky	**Attention Button Present.** This bit is set when the Attention Button is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
1	PCP	RWL	0x0 SWSticky	**Power Control Present.** This bit is set when a Power Controller is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
2	MRLP	RWL	0x0 SWSticky	**MRL Sensor Present.** This bit is set when an MRL Sensor is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
3	ATTIP	RWL	0x0 SWSticky	**Attention Indicator Present.** This bit is set when an Attention Indicator is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
4	PWRIP	RWL	0x0 SWSticky	**Power Indicator Present.** This bit is set when an Power Indicator is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
5	HPS	RWL	0x0 SWSticky	**Hot Plug Surprise.** When set, this bit indicates that a device present in the slot may be removed from the system without notice. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
6	HPC	RWL	0x0 SWSticky	**Hot Plug Capable.** This bit is set if the slot corresponding to the port is capable of supporting hot-plug operations. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
14:7	SPLV	RW	0x0	**Slot Power Limit Value.** In combination with the Slot Power Limit Scale, this field specifies the upper limit on power supplied by the slot. A Set_Slot_Power_Limit message is generated using this field whenever this register is written or when the link transitions from a DL_Down status to a DL_Up status. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.

Notes

Bit Field	Field Name	Type	Default Value	Description
16:15	SPLS	RW	0x0	**Slot Power Limit Scale.** This field specifies the scale used for the Slot Power Limit Value (SPLV). 0x0 - (x1) 1.0x 0x1 - (xp1) 0.1x 0x2 - (xp01) 0.01x 0x3 - (xp001) 0.001x A Set_Slot_Power_Limit message is generated using this field whenever this register is written or when the link transitions from a DL_Down status to a DL_Up status. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
17	EIP	RWL	0x0 SWSticky	**Electromechanical Interlock Present.** This bit is set if an electromechanical interlock is implemented on the chassis for this slot. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
18	NCCS	RO	0x0	**No Command Completed Support.** Software notification is always generated when an issued command is completed by the hot-plug controller. Therefore, this field is hardwired to zero.
31:19	PSLOTNUM	RWL	0x0 SWSticky	**Physical Slot Number.** This field indicates the physical slot number attached to this port. For devices interconnected on the system board, this field should be initialized to zero. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.

Notes

PCIESCTL - PCI Express Slot Control (0x058)

Bit Field	Field Name	Type	Default Value	Description
0	ABPE	RW	HWINIT	**Attention Button Pressed Enable.** This bit when set enables generation of a Hot-Plug interrupt or wake-up event on an attention button pressed event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
1	PFDE	RW	HWINIT	**Power Fault Detected Enable.** This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a power fault event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
2	MRLSCE	RW	HWINIT	**MRL Sensor Change Enable.** This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a MRL sensor change event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
3	PDCE	RW	HWINIT	**Presence Detected Changed Enable.** This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a presence detect change event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	CCIE	RW	HWINIT	**Command Complete Interrupt Enable.** This bit when set enables the generation of a Hot-Plug interrupt when a command is completed by the Hot-Plug Controller. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
5	HPIE	RW	HWINIT	**Hot Plug Interrupt Enable.** This bit when set enables generation of a Hot-Plug interrupt on enabled Hot-Plug events. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
7:6	AIC	RW	HWINIT	**Attention Indicator Control.** When read, this register returns the current state of the Attention Indicator. Writing to this register sets the indicator. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. This field is always zero if the ATTIP bit is cleared in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs. 0x0 - (reserved) Reserved 0x1 - (on) On 0x2 - (blink) Blink 0x3 - (off) Off

Notes

Bit Field	Field Name	Type	Default Value	Description
9:8	PIC	RW	HWINIT	**Power Indicator Control.** When read, this register returns the current state of the Power Indicator. Writing to this register sets the indicator. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. This field is always zero if the PWRIP bit is cleared in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs. 0x0 - (reserved) Reserved 0x1 - (on) On 0x2 - (blink) Blink 0x3 - (off) Off This field has no effect on the upstream port.
10	PCC	RW	HWINIT	**Power Controller Control.** When read, this register returns the current state of the power applied to the slot. Writing to this register sets the power state of the slot. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs. 0x0 -(on) Power on 0x1 -(off) Power off
11	EIC	RW	HWINIT	**Electromechanical Interlock Control.** This field always returns a value of zero when read. If an electromechanical interlock is implemented, a write of a one to this field causes the state of the interlock to toggle and a write of a zero has no effect. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
12	DLLLASCE	RW	HWINIT	**Data Link Layer Link Active State Change Enable.** This bit when set enables generation of a Hot-Plug interrupt or wake-up event on a data link layer active field state change. When the corresponding capability is enabled, the initial value of this field after a partition fundamental reset is equal to the value of the corresponding field in the PCIESCTLIV register. Once this bit is modified, the PCIESCTLIV register has no effect on this register until a subsequent partition fundamental reset occurs.
15:13	Reserved	RO	0x0	Reserved field.

Notes

PCIESSTS - PCI Express Slot Status (0x05A)

Bit Field	Field Name	Type	Default Value	Description
0	ABP	RW1C	0x0	**Attention Button Pressed.** Set when the attention button is pressed.
1	PFD	RW1C	0x0	**Power Fault Detected.** Set when the Power Controller detects a power fault.
2	MRLSC	RW1C	0x0	**MRL Sensor Changed.** Set when an MRL Sensor state change is detected.
3	PDC	RW1C	0x0	**Presence Detected Changed.** Set when a Presence Detected change is detected.
4	CC	RW1C	0x0	**Command Completed.** This bit is set when the Hot-Plug Controller completes an issued command. If the bit is already set, then it remains set. A single write to the PCI Express Slot Control (PCIESCTL) register is considered to be a single command even if it affects more than one field in that register. This command completed bit is not set until processing of all actions associated with all fields in the PCIESCTL register have completed (i.e., all associated SMBus I/O expander transactions have completed).
5	MRLSS	RO	0x0	**MRL Sensor State.** This field enclosed the current state of the MRL sensor. 0x0 -(closed) MRL closed 0x1 -(open) MRL open
6	PDS	RO	0x1	**Presence Detect State.** When the Slot Implemented (SLOT) bit is set in the PCI Express Capabilities (PCIECAP) register, this bit indicates the presence of a card in the slot corresponding to the port and reflects the state of the Presence Detect status. When the SLOT bit is cleared in the PCIECAP register, this bit is hardwired to 0x1 in downstream switch ports (i.e., it is read-only with a constant value of one). This bit is always cleared in upstream ports (i.e., it is read-only with a constant value of zero). 0x0 - (empty) Slot empty 0x1 - (present) Card present

Notes

Bit Field	Field Name	Type	Default Value	Description
7	EIS	RO	0x0	**Electromechanical Interlock Status.** When an electromechanical interlock is implemented, this bit indicates the current status of the interlock. The status of this bit is determined by the state of the corresponding PxILOCKST input signal on the I/O expander. If the I/O expander is not enabled, then the state of this bit defaults to zero (i.e., disengaged). 0x0 - (disengaged) Electromechanical interlock disengaged 0x1 - (engaged) Electromechanical interlock engaged
8	DLLLASC	RW1C	0x0	**Data Link Layer Link Active State Change.** This bit is set when the state of the data link layer active field in the link status register changes state. 0x0 - (nochange) No DLLA state change 0x1 - (changed) DLLA state change
15:9	Reserved	RO	0x0	Reserved field.

PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTRS	RO	0x0	**Completion Timeout Ranges Supported.** Not applicable.
4	CTDS	RO	0x0	**Completion Timeout Disable Supported.** Not applicable.
5	ARIFS	Upstream Port: RO Downstream Switch Port: RWL	Upstream Port: 0x0 Downstream Switch Port: 0x1 MSWSticky	**ARI Forwarding Supported.** This bit is set to indicate that the switch supports Alternative Routing ID (ARI) Forwarding. When this bit is cleared, the ARI Forwarding Enable (ARIFEN) bit in the Device Control 2 register becomes read-only zero. This bit is read-only zero in an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
6	ATOPRS	RO	0x0	**AtomicOp Routing Supported.** The switch does not support routing of Atomic Operations.
7	ATOPC32S	RO	0x0	**32-bit AtomicOp Completer Supported.** Not applicable.
8	ATOPC64S	RO	0x0	**64-bit AtomicOp Completer Supported.** Not applicable.
9	CASC128S	RO	0x0	**128-bit CAS Completer Supported.** Not applicable.
10	NROEP	RO	0x1	**No RO-enabled PR-PR Passing.** The ordering logic in the switch (see section Packet Ordering on page 4-6) never carries out the passing between posted TLPs with the relaxed ordering bit set.

Notes

Bit Field	Field Name	Type	Default Value	Description
11	LTRMS	RO	0x0	**LTR Mechanism Supported.** The switch does not support the Latency Tolerance Reporting mechanism.
13:12	TPHCS	RO	0x0	**TPH Completer Supported.** Not applicable.
19:14	Reserved	RO	0x0	Reserved field.
20	EFMTFS	RO	0x0	**Extended Fmt Field Supported.** The switch does not support the 3-bit definition of the FMT field in TLPs.
21	E2ETPS	RO	0x0	**End-to-End TLP Prefix Supported.** The switch does not support End-to-End TLP Prefixes.
31:22	Reserved	RO	0x0	Reserved field.

PCIEDCTL2 - PCI Express Device Control 2 (0x068)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTRS	RO	0x0	**Completion Timeout Ranges Supported.** Not applicable.
4	CTDS	RO	0x0	**Completion Timeout Disable Supported.** Not applicable.
5	ARIFEN	Upstream Port: RO Downstream Switch Port: RW	0x0	**ARI Forwarding Enable.** When set, the downstream switch port disables its traditional Device Number field being zero enforcement when turning a Type 1 configuration request into a Type 0 configuration request, permitting access to the extended functions in an ARI device immediately below the port. When the ARIFS bit in the PCIEDCAP2 register is cleared, this bit is read-only zero. This bit is always read-only zero in an upstream port.
6	ATOPRE	RO	0x0	**AtomicOp Requester Enable.** Not applicable.
7	ATOPEB	RO	0x0	**AtomicOp Egress Blocking.** The switch does not support routing of Atomic Operations.
8	IDORE	RO	0x0	**IDO Request Enable.** Not supported.
9	IDOCE	RO	0x0	**IDO Completion Enable.** Not supported.
10	LTRME	RO	0x0	**LTR Mechanism Enable.** Not supported.
14:11	Reserved	RO	0x0	Reserved field.
15	E2ETLPPB	RO	0x0	**End-to-End TLP Prefix Blocking.** Not supported.

Notes

PCIEDSTS2 - PCI Express Device Status 2 (0x06A)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

PCIELCTL2 - PCI Express Link Control 2 (0x070)

Bit Field	Field Name	Type	Default Value	Description
3:0	TLS	RW	0x2 Sticky	**Target Link Speed.** For downstream switch ports, this field sets an upper limit on the link operational speed by restricting the values advertised by the downstream switch port (i.e., upstream component) in its training sequences. For both upstream and downstream switch ports, this field is used to set the target compliance mode speed when software is using the ECOMP bit in this register to force a link into compliance mode. The switch supports 2.5 GT/s and 5.0 GT/s operation. Setting this field to an unsupported value produces undefined results. 0x1 - (gen1) 2.5 GT/s 0x2 - (gen2) 5.0 GT/s others-reserved
4	ECOMP	RW	0x0 Sticky	**Enter Compliance.** Software is permitted to force a link into compliance mode at the speed indicated by the TLS field by setting this bit in both components on a link and then initiating a hot reset on the link.
5	HASD	RO	0x0	**Hardware Autonomous Speed Disable.** Switch ports do not have an autonomous mechanism to regulate link speed, except due to link reliability issues. Therefore, this bit is not applicable. Note that this bit does not affect link speed changes triggered by software setting the target link speed and link-retrain bits. Refer to section Software Management of Link Speed on page 7-8 for further details.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	SDE	RWL	0x0 SWSticky	**Selectable De-emphasis.** For a downstream switch port, this bit sets the de-emphasis level when the link operates at 5.0 GT/s. Per the PCI Express Base Specification, this bit is not applicable for upstream ports. Still, for the switch's upstream port, this bit selects the de-emphasis preference advertised via training sets (the actual de-emphasis on the link is selected by the link partner). 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB This bit has no effect when the link operates at 2.5 GT/s, or when the link operates in low-swing mode. When this field is modified, the newly selected de-emphasis is not applied until the PHY LTSSM transitions through the states in which it is allowed to modify the de-emphasis setting on the line (e.g., Recovery.Speed). Therefore, after modifying this field, it is recommended that the link be fully retrained by setting the FLRET bit in the PHYLSTATE0 register.
9:7	TM	RW	0x0 Sticky	**Transmit Margin.** This field controls the value of the non de-emphasized voltage level at the transmitter pins. This field is reset to 0x0 on entry to the LTSSM Polling.Configuration substate. 0x0 - Normal operating range 0x1 - 900 mV for full swing and 500 mV for low-swing 0x2 - 700 mV for full swing and 400 mV for low-swing 0x3 - 500 mV for full swing and 300 mV for low-swing 0x4 - 300 mV for full swing and 200 mv for low-swing 0x5 - 200 mV for full swing and 100 mv for low-swing 0x6 - 0x7 - Reserved This register is intended for debug and compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value. When this field is set to "Normal Operating Range", the SerDes transmitter drive level is selected via the SerDes Transmitter Control registers (S[x]TXLCTL0 and S[x]TXLCTL1). Refer to section SerDes Transmitter Controls on page 8-3. When this field is modified, the newly selected value is not applied until the PHY LTSSM transitions through the states in which it is allowed to modify the transmit margin setting on the line (i.e., Recovery.RcvrLock). Therefore, after modifying this field, it is recommended that the link be retrained by setting the LRET bit in the PCIELCTL register. **Note: This field has no effect when the port operates in SerDes Test mode.**

Notes

Bit Field	Field Name	Type	Default Value	Description
10	EMC	RW	0x0 Sticky	**Enter Modified Compliance.** When this bit is set to 1b, the port transmits the modified compliance pattern if the LTSSM enters Polling.Compliance state. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.
11	CSOS	RW	0x0 Sticky	**Compliance SOS.** When set to 1b, the LTSSM is required to send SOS periodically in between the compliance and modified compliance patterns.
12	CDE	RW	0x0 Sticky	**Compliance De-emphasis.** This bit selects the de-emphasis value in the Polling.Compliance state when this state was entered as a result of setting the Enter Compliance (ECOMP) bit in this register. 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB This bit is intended for debug, compliance testing purposes. System firmware and software is allowed to modify this bit only during debug or compliance testing.
15:13	Reserved	RO	0x0	Reserved field.

PCIELSTS2 - PCI Express Link Status 2 (0x072)

Bit Field	Field Name	Type	Default Value	Description
0	CDE	RO	0x0	**Current De-emphasis.** The value of this bit indicates the current de-emphasis level when the link operates in 5.0 GT/s. 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB The value of this bit in undefined when the link operates at 2.5 GT/s.
15:1	Reserved	RO	0x0	Reserved field.

PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

PCIESCTL2 - PCI Express Slot Control 2 (0x078)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

Notes

PCIESSTS2 - PCI Express Slot Status 2 (0x07A)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

PCI Power Management Capability Structure

PMCAP - PCI Power Management Capabilities (0x0C0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 identifies this capability as a PCI power management capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
18:16	VER	RO	0x3	**Power Management Capability Version.** Complies with version the PCI Bus Power Management Interface Specification, Revision 1.2.
19	PMECLK	RO	0x0	**PME Clock.** Does not apply to PCI Express.
20	Reserved	RO	0x0	Reserved field.
21	DEVSP	RWL	0x0 SWSticky	**Device Specific Initialization.** The value of zero indicates that no device specific initialization is required.
24:22	AUXI	RO	0x0	**AUX Current.** The switch does not use auxiliary current.
25	D1	RO	0x0	**D1 Support.** This field indicates that this function does not support D1.
26	D2	RO	0x0	**D2 Support.** This field indicates this function does not support D2.
31:27	PME	RWL	0b11001 SWSticky	**PME Support.** This field indicates the power states in which the function may generate a PME. Bits 27, 30 and 31 are set to indicate that the bridge will forward PME messages. The switch does not forward PME messages in $D3_{cold}$. This functionality may be supported in the system by routing WAKE# around the switch. Modification of this field modifies the advertised capability value but does not modify the function's behavior (i.e., PME is generated in the states noted in the default value).

Notes

PMCSR - PCI Power Management Control and Status (0x0C4)

Bit Field	Field Name	Type	Default Value	Description
1:0	PSTATE	RW	0x0	**Power State.** This field is used to determine the current power state of the function and to set a new power state. 0x0 - (d0) D0 state 0x1 - (d1) D1 state (not supported by the switch and reserved) 0x2- (d2) D2 state (not supported by the switch and reserved) 0x3 - (d3) $D3_{hot}$ state
2	Reserved	RO	0x0	Reserved field.
3	NOSOFTRST	RWL	0x1 SWSticky	**No Soft Reset.** This bit indicates if the configuration context is preserved by the bridge when the device transitions from a D3hot to D0 power management state. 0x0 - (reset) State reset 0x1 - (preserved) State preserved
7:4	Reserved	RO	0x0	Reserved field.
8	PMEE	RW	0x0 Sticky	**PME Enable.** When this bit is set, PME message generation is enabled for the function. If a hot plug wake-up event is desired when exiting the $D3_{cold}$ state, then this bit should be set during serial EEPROM initialization. A hot reset does not result in modification of this field.
12:9	DSEL	RO	0x0	**Data Select.** The optional data register is not implemented.
14:13	DSCALE	RO	0x0	**Data Scale.** The optional data register is not implemented.
15	PMES	RW1C	0x0 Sticky	**PME Status.** This bit is set if a PME is generated by the function even if the PMEE bit is cleared. This bit is not set when the bridge function is propagating a PME message but the bridge function is not itself generating a PME. Since the upstream port never generates a PME, this bit will never be set in that port.
21:16	Reserved	RO	0x0	Reserved field.
22	B2B3	RO	0x0	**B2/B3 Support.** Does not apply to PCI Express.
23	BPCCE	RO	0x0	**Bus Power/Clock Control Enable.** Does not apply to PCI Express.
31:24	DATA	RO	0x0	**Data.** This optional field is not implemented.

Notes

Message Signaled Interrupt Capability Structure

MSICAP - Message Signaled Interrupt Capability and Control (0x0D0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x5	**Capability ID.** The value of 0x5 identifies this capability as a MSI capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. This field is set to 0x0 indicating that it is the last capability. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
16	EN	RW	0x0	**Enable.** This bit enables MSI. 0x0 - (disable) disabled 0x1 - (enable) enabled
19:17	MMC	RO	0x0	**Multiple Message Capable.** This field contains the number of requested messages.
22:20	MME	RW	0x0	**Multiple Message Enable.** Hardwired to one message.
23	A64	RO	0x1	**64-bit Address Capable.** The bridge is capable of generating messages using a 64-bit address.
31:24	Reserved	RO	0x0	Reserved field.

MSIADDR - Message Signaled Interrupt Address (0x0D4)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	ADDR	RW	0x0	**Message Address.** This field specifies the lower portion of the DWORD address of the MSI memory write transaction. Refer to section Interrupts on page 10-4 for restrictions on the programming of this field.

Notes

MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8)

Bit Field	Field Name	Type	Default Value	Description
31:0	UADDR	RW	0x0	**Upper Message Address.** This field specifies the upper portion of the DWORD address of the MSI memory write transaction. If the contents of this field are non-zero, then 64-bit address is used in the MSI memory write transaction. If the contents of this field are zero, then the 32-bit address specified in the MSI-ADDR field is used. Refer to section Interrupts on page 10-4 for restrictions on the programming of this field.

MSIMDATA - Message Signaled Interrupt Message Data (0x0DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	MDATA	RW	0x0	**Message Data.** This field contains the lower 16-bits of data that are written when a MSI is signaled.
31:16	Reserved	RO	0x0	Reserved field.

Subsystem ID and Subsystem Vendor ID

SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0xD	**Capability ID.** The value of 0xD identifies this capability as a SSID/SSVID capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
31:16	Reserved	RO	0x0	Reserved field.

Notes

SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4)

Bit Field	Field Name	Type	Default Value	Description
15:0	SSVID	RWL	0x0 SWSticky	**Subsystem Vendor ID.** This field identifies the manufacturer of the add-in card or subsystem. SSVID values are assigned by the PCI-SIG to insure uniqueness.
31:16	SSID	RWL	0x0 SWSticky	**Subsystem ID.** This field identifies the add-in card or subsystem. SSID values are assigned by the vendor.

Extended Configuration Space Access Registers

ECFGADDR - Extended Configuration Space Access Address (0x0F8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
7:2	REG	RW	0x0	**Register Number.** This field selects the configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification. The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). Violation of this rule produces undefined results. Also, the value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). Violation of this rule produces undefined results.
11:8	EREG	RW	0x0	**Extended Register Number.** This field selects the extended configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification. The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). Violation of this rule produces undefined results. Also, the value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). Violation of this rule produces undefined results.
31:12	Reserved	RO	0x0	Reserved field.

Notes

ECFGDATA - Extended Configuration Space Access Data (0x0FC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Configuration Data.** A read from this field will return the configuration space register value pointed to by the ECFGADDR register. A write to this field will update the contents of the configuration space register pointed to by the ECFGADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

Advanced Error Reporting (AER) Extended Capability

AERCAP - AER Capabilities (0x100)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 indicates an advanced error reporting capability structure.
19:16	CAPVER	RO	0x2	**Capability Version.** The value of 0x2 indicates compatibility with PCI Express Base Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** Next capability pointer. The value of 0x0 terminates the list. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

AERUES - AER Uncorrectable Error Status (0x104)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW1C	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW1C	0x0 Sticky	**Data Link Protocol Error Status.** This bit is set when a data link layer protocol error is detected.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	SDOENERR	Upstream Port: RO Downstream Switch Port: RW1C	0x0 Sticky	**Surprise Down Error Status.** This bit is set when a surprise down error is detected and the SDERR bit in the PCIELCAP register is set. This bit is not applicable for an upstream port.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW1C	0x0 Sticky	**Poisoned TLP Status.** This bit is set when a poisoned TLP is detected.
13	FCPERR	RO	0x0	**Flow Control Protocol Error Status.** Not applicable (the switch does not support flow control protocol error checking).
14	COMPTO	RO	0x0	**Completion Timeout Status.** Not applicable (this function does not initiate non-posted requests on its own behalf).
15	CABORT	RO	0x0	**Completer Abort Status.** Not applicable. This bit is never set as this function never responds to a non-posted request with a completer abort, except for ACS violations. For this exception case, the error is an ACS violation error and is not logged as a completer abort error.
16	UECOMP	RW1C	0x0 Sticky	**Unexpected Completion Status.** This bit is set when an unexpected completion is detected.
17	RCVOVR	RW1C	0x0 Sticky	**Receiver Overflow Status.** This bit is set when a receiver overflow is detected.
18	MALFORMED	RW1C	0x0 Sticky	**Malformed TLP Status.** This bit is set when a malformed TLP is detected.
19	ECRC	RW1C	0x0 Sticky	**ECRC Status.** This bit is set when an ECRC error is detected.
20	UR	RW1C	0x0 Sticky	**UR Status.** This bit is set when an unsupported request is detected.
21	ACSV	RW1C	0x0 Sticky	**ACS Violation Status.** This bit is set when an ACS violation is detected.
22	UIE	RW1C	0x0 Sticky	**Uncorrectable Internal Error Status.** This bit is set when an uncorrectable internal error associated with this function is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.

Notes

Bit Field	Field Name	Type	Default Value	Description
23	MCBLKTLP	RW1C	0x0 Sticky	**MC Blocked TLP Status.** This bit is set when a multicast TLP is blocked by this function in response to the setting of the MC_Block_All and MC_Block_Untranslated bits in the multicast extended capability structure. When the Disable Multicast Error Reporting (DMCER) bit is set in the Switch Control (SWCTL) register, this field becomes read-only with a value of zero.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Status.** The switch does not support Atomic Operations.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Status.** The switch does not support TLP Prefixes.
31:26	Reserved	RO	0x0	Reserved field.

AERUEM - AER Uncorrectable Error Mask (0x108)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x0 Sticky	**Data Link Protocol Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
5	SDOENERR	Upstream Port: RO Downstream Switch Port: RW	0x0 Sticky	**Surprise Down Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
13	FCPERR	RO	0x0	**Flow Control Protocol Error Mask.** Not applicable.
14	COMPTO	RO	0x0	**Completion Timeout Mask.** Not applicable.
15	CABORT	RO	0x0	**Completer Abort Mask.** Not applicable.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
17	RCVOVR	RW	0x0 Sticky	**Receiver Overflow Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
18	MALFORMED	RW	0x0 Sticky	**Malformed TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
19	ECRC	RW	0x0 Sticky	**ECRC Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
20	UR	RW	0x0 Sticky	**UR Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
21	ACSV	RW	0x0 Sticky	**ACS Violation Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
22	UIE	RW	0x1 Sticky	**Uncorrectable Internal Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.
23	MCBLKTLP	RW	0x0 Sticky	**MC Blocked TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register. When the Disable Multicast Error Reporting (DMCER) bit is set in the Switch Control (SWCTL) register, this field becomes read-only with a value of zero.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Mask**. Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Mask** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERUESV - AER Uncorrectable Error Severity (0x10C)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	
4	DLPERR	RW	0x1 Sticky	**Data Link Protocol Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	SDOENERR	Upstream Port: RO Down-stream Switch Port: RW	0x1 Sticky	**Surprise Down Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Status Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
13	FCPERR	RO	0x1	**Flow Control Protocol Error Severity.** Not applicable.
14	COMPTO	RO	0x0	**Completion Timeout Severity.** Not applicable.
15	CABORT	RO	0x0	**Completer Abort Severity.** Not applicable.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
17	RCVOVR	RW	0x1 Sticky	**Receiver Overflow Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
18	MALFORMED	RW	0x1 Sticky	**Malformed TLP Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
19	ECRC	RW	0x0 Sticky	**ECRC Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
20	UR	RW	0x0 Sticky	**UR Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.
21	ACSV	RW	0x0 Sticky	**ACS Violation Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error.

Notes

Bit Field	Field Name	Type	Default Value	Description
22	UIE	RW	0x1 Sticky	**Uncorrectable Internal Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of one.
23	MCBLKTLP	RW	0x0 Sticky	**MC Blocked TLP Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as a non-fatal error. When the Disable Multicast Error Reporting (DMCER) bit is cleared in the Switch Control (SWCTL) register, this field becomes read-only with a value of zero.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Severity.** Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Severity.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERCES - AER Correctable Error Status (0x110)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW1C	0x0 Sticky	**Receiver Error Status.** This bit is set when the physical layer detects a receiver error.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW1C	0x0 Sticky	**Bad TLP Status.** This bit is set when a bad TLP is detected.
7	BADDLLP	RW1C	0x0 Sticky	**Bad DLLP Status.** This bit is set when a bad DLLP is detected.
8	RPLYROVR	RW1C	0x0 Sticky	**Replay Number Rollover Status.** This bit is set when a replay number rollover has occurred indicating that the data link layer has abandoned replays and has requested that the link be retrained.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW1C	0x0 Sticky	**Replay Timer Timeout Status.** This bit is set when the replay timer in the data link layer times out.
13	ADVISORYNF	RW1C	0x0 Sticky	**Advisory Non-Fatal Error Status.** This bit is set when an advisory non-fatal error is detected as described in Section 6.2.3.2.4 of the PCI Express Base Specification.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	CIE	RW1C	0x0 Sticky	**Correctable Internal Error Status.** This bit is set whenever an correctable internal error associated with the port is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.
15	HLO	RW1C	0x0 Sticky	**Header Log Overflow Status.** This bit is set when an error that requires packet-header logging occurs but the packet header cannot be logged by the port's AER Header Log registers (AERHL[1:4]DW). A packet's header can't be logged in the AER Header Log registers when an error occurs while the First Error Pointer (FEPTR field in the AERCTL register) is valid. The First Error Pointer is valid when it points to a set bit in the AERUES register (i.e., indicating the occurrence of a prior uncorrectable error which has not been cleared by software). When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.
31:16	Reserved	RO	0x0	Reserved field.

AERCEM - AER Correctable Error Mask (0x114)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 Sticky	**Receiver Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 Sticky	**Bad TLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
7	BADDLLP	RW	0x0 Sticky	**Bad DLLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	RPLYROVR	RW	0x0 Sticky	**Replay Number Rollover Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 Sticky	**Replay Timer Timeout Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
13	ADVISORYNF	RW	0x1 Sticky	**Advisory Non-Fatal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
14	CIE	RW	0x1 Sticky	**Correctable Internal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.
15	HLO	RW	0x1 Sticky	**Header Log Overflow Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero.
31:16	Reserved	RO	0x0	Reserved field.

Notes

AERCTL - AER Capabilities and Control (0x118)

Bit Field	Field Name	Type	Default Value	Description
4:0	FEPTR	RO	0x0 Sticky	**First Error Pointer.** This field contains a pointer to the bit in the AERUES register that resulted in the first reported error. This field is valid only when the bit in the AERUES register pointed to by this field is set.
5	ECRCGC	RWL	0x1 SWSticky	**ECRC Generation Capable.** This bit indicates if the function is capable of generating ECRC.
6	ECRCGE	RW	0x0 Sticky	**ECRC Generation Enable.** When this bit is set, ECRC generation is enabled for the function.
7	ECRCCC	RWL	0x1 SWSticky	**ECRC Check Capable.** This bit indicates if the function is capable of checking ECRC.
8	ECRCCE	RW	0x0 Sticky	**ECRC Check Enable.** When this bit is set, ECRC checking is enabled for the function.
9	MHRC	RO	0x0	**Multiple Header Recording Capable.** Device ports do not support recording of multiple packet headers.
10	MHRE	RO	0x0	**Multiple Header Recording Enable.** Device ports do not support recording of multiple packet headers.
31:11	Reserved	RO	0x0	Reserved field.

AERHL1DW - AER Header Log 1st Doubleword (0x11C)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 1st doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL2DW - AER Header Log 2nd Doubleword (0x120)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 2nd doubleword of the TLP header that resulted in the first reported uncorrectable error.

Notes

AERHL3DW - AER Header Log 3rd Doubleword (0x124)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 3rd doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL4DW - AER Header Log 4th Doubleword (0x128)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 4th doubleword of the TLP header that resulted in the first reported uncorrectable error.

Device Serial Number Extended Capability

SNUMCAP - Serial Number Capabilities (0x180)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x3	**Capability ID.** The value of 0x3 indicates a device serial number capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** Next capability pointer. The value of 0x0 terminates the list. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

SNUMLDW - Serial Number Lower Doubleword (0x184)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 SWSticky	**Lower 32-bits of Device Serial Number.** This field contains the lower 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device. When a port operates in a multi-function mode, this field must be programmed identically in all functions that implement this capability.

Notes

SNUMUDW - Serial Number Upper Doubleword (0x188)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 SWSticky	**Upper 32-bits of Device Serial Number.** This field contains the upper 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device. When a port operates in a multi-function mode, this field must be programmed identically in all functions that implement this capability.

PCI Express Virtual Channel Capability

The Virtual Channel (VC) capability structure defined in this section is only applicable for port operating modes in which the PCI-to-PCI bridge function is function 0 of the port. For other port operating modes, this capability structure must not be linked into the extended capabilities list in the PCI-to-PCI bridge function and the registers in this capability structure are 'reserved' [1] (i.e., must not be programmed).

PCIEVCECAP - PCI Express VC Extended Capability Header (0x200)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x2	**Capability ID.** The value of 0x2. indicates a virtual channel capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** Next capability pointer. The value of 0x0 terminates the list. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

PVCCAP1- Port VC Capability 1 (0x204)

Bit Field	Field Name	Type	Default Value	Description
2:0	EVCCNT	RO	0x0	**Extended VC Count.** A value 0x0 indicates that only the default VC (VC0) is implemented.
3	Reserved	RO	0x0	Reserved field.
6:4	LPEVCCNT	RO	0x0	**Low Priority Extended VC Count.** Not applicable (only the default VC0 is implemented).

1. Reading from a reserved address returns and undefined value. Writes to a reserved address complete successfully but produce undefined behavior.

Notes

Bit Field	Field Name	Type	Default Value	Description
7	Reserved	RO	0x0	Reserved field.
9:8	REFCLK	RO	0x0	**Reference Clock.** Not supported (i.e., Time-based WRR Port Arbitration is not implemented).
11:10	PATBLSIZ	RO	0x0	**Port Arbitration Table Entry Size.** This field indicates the size of the port arbitration table. This function only supports hardware fixed round-robin, so the port arbitration table is not implemented.
31:12	Reserved	RO	0x0	Reserved field.

PVCCAP2- Port VC Capability 2 (0x208)

Bit Field	Field Name	Type	Default Value	Description
7:0	VCARBCAP	RO	0x0	**VC Arbitration Capability.** Not applicable (only the default VC0 is implemented).
23:8	Reserved	RO	0x0	Reserved field.
31:24	VCATBLOFF	RO	0x0	**VC Arbitration Table Offset.** Not applicable.

PVCCTL - Port VC Control (0x20C)

Bit Field	Field Name	Type	Default Value	Description
0	LVCAT	RO	0x0	**Load VC Arbitration Table.** Not applicable.
3:1	VCARBSEL	RW	0x0	**VC Arbitration Select.** Not applicable (only the default VC0 is implemented). This field has RW type for compliance with the PCI Express Base Specification.
15:4	Reserved	RO	0x0	Reserved field.

PVCSTS - Port VC Status (0x20E)

Bit Field	Field Name	Type	Default Value	Description
0	VCATS	RO	0x0	**VC Arbitration Table Status.** Not applicable.
15:1	Reserved	RO	0x0	Reserved field.

Notes

VCR0CAP- VC Resource 0 Capability (0x210)

Bit Field	Field Name	Type	Default Value	Description
7:0	PARBC	RO	0x1	**Port Arbitration Capability.** This field indicates the type of port arbitration supported by this VC resource. Each bit corresponds to a port arbitration capability. Device ports support only Hardware Fixed Round Robin. bit 0 - Hardware Fixed (i.e., defaults to round-robin) bit 1 - Weighted Round Robin with 32 phases bit 2 - Weighted Round Robin with 64 phases bit 3 - Weighted Round Robin with 128 phases bit 4 - Time-Based Weighted Round Robin with 128 phases bit 5 - Weighted Round Robin with 128 phases bits 6 and 7 - reserved
14:8	Reserved	RO	0x0	Reserved field.
15	RJST	RO	0x0	**Reject Snoop Transactions.** No supported for switch ports.
22:16	MAXTS	RO	0x0	**Maximum Time Slots.** Since this VC does not support time-based WRR, this field is not valid.
23	Reserved	RO	0x0	Reserved field.
31:24	PATBLOFF	RO	0x0	**Port Arbitration Table Offset.** Device ports only support Hardware Fixed Round Robin. Therefore, the port arbitration table is not present.

VCR0CTL- VC Resource 0 Control (0x214)

Bit Field	Field Name	Type	Default Value	Description
7:0	TCVCMAP	bit 0: RO bits 1 through 7: RW	0xFF	**TC/VC Map.** This field indicates the TCs that are mapped to the VC resource. Each bit corresponds to a TC. When a bit is set, the corresponding TC is mapped to the VC.
15:8	Reserved	RO	0x0	Reserved field.
16	LPAT	RO	0x0	**Load Port Arbitration Table.** Not applicable (i.e., the port arbitration table is not present).
19:17	PARBSEL	RW	0x0	**Port Arbitration Select.** This field configures the VC resource to provide a particular port arbitration service. 0x0 - Hardware Fixed Round Robin Others - Reserved (not supported) If the port arbitration scheme selected in this field is not one of the supported advertised schemes, the operation of the device is undefined.
23:20	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
26:24	VCID	RO	0x0	**VC ID.** This field assigns a VC ID to the VC resource. For VC0, this field is always hardwired to zero.
30:27	Reserved	RO	0x0	Reserved field.
31	VCEN	RO	0x1	**VC Enable.** This field, when set, enables a virtual channel. For VC0, this field is hardwired to 0x1 (enabled).

VCR0STS - VC Resource 0 Status (0x218)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.
16	PATS	RO	0x0	**Port Arbitration Table Status.** Not applicable (i.e., the port arbitration table is not present).
17	VCNEG	RO	0x0	**VC Negotiation Pending.** This bit is not applicable for VC0 and is therefore hardwired to 0x0.
31:18	Reserved	RO	0x0	Reserved field.

ACS Extended Capability

ACSECAPH - ACS Extended Capability Header (0x320)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0xD	**Capability ID.** The value of 0xD indicates an ACS extended capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** Next capability pointer. The value of 0x0 terminates the list. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

Notes

ACSCAP - ACS Capability Register (0x324)

Bit Field	Field Name	Type	Default Value	Description
0	V	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS Source Validation.** If set, indicates that this function implements ACS Source Validation. This field must never be set to 0x1 in an upstream port.
1	B	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS Translation Blocking.** If set, indicates that this function implements ACS Translation Blocking. This field must never be set to 0x1 in an upstream port.
2	R	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS P2P Request Redirect.** If set, indicates that this function implements ACS Peer-to-Peer Request Redirect. For a downstream switch port, peer-to-peer refers to transfers among downstream switch ports in the same partition. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. In an upstream port, this field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function).
3	C	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS P2P Completion Redirect.** If set, indicates that this function implements ACS Peer-to-Peer Completion Redirect. For a downstream switch port, peer-to-peer refers to transfers among downstream switch ports in the same partition. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. In an upstream port, this field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function).
4	U	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS Upstream Forwarding.** If set, indicates that this function implements ACS Upstream Forwarding. This field must never be set to 0x1 in an upstream port.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	E	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS P2P Egress Control.** If set, indicates that this function implements ACS Peer-to-Peer Egress Control. For a downstream switch port, peer-to-peer refers to transfers among downstream switch ports in the same partition. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. The switch does not support ACS P2P Egress Control among functions in a multi-function upstream port. Therefore, this field must never be set to 0x1 in an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
6	T	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x1 MSWSticky	**ACS Direct Translated P2P.** If set, indicates this function implements ACS Direct Translated Peer-to-Peer. For a downstream switch port, peer-to-peer refers to transfers among downstream switch ports in the same partition. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. This bit is ignored if the ACS Translation Blocking (B) bit is set to 0x1. In an upstream port, this field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function). Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
7	Reserved	RO	0x0	Reserved field.
15:8	ECVS	RWL	Upstream Port: 0x0 Down-stream Switch Port: 0x18 MSWSticky	**Egress Control Vector Size.** Indicates the number of applicable bits in the ACS Egress Control Vector register. The value of 0x18 indicates that egress control may be done with up to 24 ports. The value of this field is undefined if the ACS P2P Egress Control bit in this register is set to 0x0. This field must never be set to a value other than 0x0 in an upstream port. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

Notes

ACSCTL - ACS Control Register (0x326)

Bit Field	Field Name	Type	Default Value	Description
0	V	Upstream Port: RO Downstream Switch Port: RW	0x0	**ACS Source Validation Enable.** When set, this function performs ACS Source Validation. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
1	B	Upstream Port: RO Downstream Switch Port: RW	0x0	**ACS Translation Blocking Enable.** When set, this function performs ACS Translation Blocking. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
2	R	RW	0x0	**ACS P2P Request Redirect Enable.** When set, this function performs ACS Peer-to-Peer Request Redirect. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
3	C	RW	0x0	**ACS P2P Completion Redirect Enable.** When set, this function performs ACS Peer-to-Peer Completion Redirect. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
4	U	Upstream Port: RO Downstream Switch Port: RW	0x0	**ACS Upstream Forwarding Enable.** When set, this function performs ACS Upstream Forwarding. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	E	Upstream Port: RO Downstream Switch Port: RW	0x0	**ACS P2P Egress Control Enable.** When set, this function performs ACS Peer-to-Peer Egress Control. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
6	T	RW	0x0	**ACS Direct Translated P2P Enable.** When set, this function performs ACS Direct Translated Peer-to-Peer control. This bit is ignored if ACS Translation Blocking is enabled. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
15:7	Reserved	RO	0x0	Reserved field.

ACSECV - ACS Egress Control Vector (0x328)

Bit Field	Field Name	Type	Default Value	Description
23:0	ECV	See Description	0x0	**Egress Control Vector.** This field is used to configure ACS peer-to-peer egress control. The value in this field is only valid when ACS peer-to-peer egress control is enabled in the ACSCTL register. Each bit in this register corresponds to a port. Bit[0] corresponds to port 0, bit[1] corresponds to port 1, and so on. When a bit is set, peer-to-peer requests targeting the associated port are blocked or redirected. Refer to Section 6.12.3 of the PCI Express Base Specification for further details. The bit corresponding to this port number is read-only and hardwired to 0x0. For example, bit[0] is read-only for port 0, bit[1] is read-only for port 1, and so on. Depending on stack configuration, ports may be activated or de-activated (see section Stack Configuration on page 3-5). Bits in this register corresponding to de-activated ports are read-write but have no effect on the operation of the device.
31:24	Reserved	RO	0x0	Reserved field.

Notes

Multicast Extended Capability

MCCAPH - Multicast Extended Capability Header (0x330)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x12	**Capability ID.** The value of 0x12 indicates a multicast capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** Next capability pointer. The value of 0x0 terminates the list. The default value of this register depends on the port's operating mode. See section PCI-to-PCI Bridge Capability Structures on page 19-9 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

MCCAP - Multicast Capability (0x334)

Bit Field	Field Name	Type	Default Value	Description
5:0	MAXGROUP	RWL	0x3F SWSticky	**Max Multicast Groups.** This field indicates the maximum number of multicast groups supported by the switch partition. The number of supported groups is equal to the value in this field plus one.
14:6	Reserved	RO	0x0	Reserved field.
15	ECRCREG	RWL	0x1 SWSticky	**ECRC Regeneration Supported.** This bit is set to indicate that the switch supports multicast ECRC regeneration.

Notes

MCCTL- Multicast Control (0x336)

Bit Field	Field Name	Type	Default Value	Description
5:0	NUMGROUP	RW	0x0	**Number of Multicast Groups.** When the Multicast Enabler (MEN) bit is set, this field indicates the number of multicast groups that are enabled. The number of groups enabled is equal to the value in this field plus one. The behavior is undefined when the value in this field exceeds the value of the MAXGROUP field in the MCCAP register. This field must be set identically in all port functions in the partition associated with this port.
14:6	Reserved	RO	0x0	Reserved field.
15	MEN	RW	0x0	**Multicast Enable.** When this bit is set, multicast is enabled in the switch partition associated with this port. This field must be set identically in all port functions in the partition associated with this port.

MCBARL- Multicast Base Address Low (0x338)

Bit Field	Field Name	Type	Default Value	Description
5:0	INDEXPOS	RW	0x0	**Index Position.** When multicast is enabled, this field specifies the least significant bit of the multicast group number within a TLP address. The behavior is undefined when multicast is enabled and this field is less than 12. This field must be set identically in all port functions in the partition associated with this port.
11:6	Reserved	RO	0x0	Reserved field.
31:12	MCBARL	RW	0x0	**Multicast BAR Low.** This field specifies the lower 20-bits (i.e., bits 12 through 31) of the multicast BAR. The behavior is undefined if bits in this field corresponding to address bits that contain the multicast group number or those less than the multicast index position (i.e., INDEXPOS) are non-zero. This field must be set identically in all port functions in the partition associated with this port.

Notes

MCBARH- Multicast Base Address High (0x33C)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBARH	RW	0x0	**Multicast BAR High.** This field specifies the upper 32-bits (i.e., bits 32 through 63) of the multicast BAR. The behavior is undefined if bits in this field corresponding to address bits that contain the multicast group number or those less than the multicast index position (i.e., INDEX-POS) are non-zero. This field must be set identically in all port functions in the partition associated with this port.

MCRCVL- Multicast Receive Low (0x340)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCRCV	RW	0x0	**Multicast Receive.** Each bit in this field corresponds to one of the lower 32 multicast groups (e.g., bit 0 corresponds to multicast group 0, bit 1 corresponds to multicast group 1, and so on). When a bit is set in this field for an enabled multicast group, multicast TLPs associated with that multicast group that reach the virtual PCI bus of the partition and were not received on this port are forwarded out the port (i.e., the port associated with the PCI-to-PCI bridge in which this register resides). The value of bits greater than NUMGROUP in the MCCTL register is ignored.

MCRCVH- Multicast Receive High (0x344)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCRCV	RW	0x0	**Multicast Receive.** Each bit in this field corresponds to one of the upper 32 multicast groups (e.g., bit 0 corresponds to multicast group 32, bit 1 corresponds to multicast group 33, and so on). When a bit is set in this field for an enabled multicast group, multicast TLPs associated with that multicast group that reach the virtual PCI bus of the partition and were not received on this port are forwarded out the port (i.e., the port associated with the PCI-to-PCI bridge in which this register resides). The value of bits greater than NUMGROUP in the MCCTL register is ignored.

Notes

MCBLKALLL- Multicast Block All Low (0x348)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKALL	RW	0x0	**Multicast Block All.** Each bit in this field corresponds to one of the lower 32 multicast groups (e.g., bit 0 corresponds to multicast group 0, bit 1 corresponds to multicast group 1, and so on). When a bit is set in this field for an enabled multicast group, the PCI-to-PCI bridge associated with this register is blocked from forwarding multicast TLPs associated with that multicast group received on that port. This is an ingress port function performed on received TLPs. The value of bits greater than NUMGROUP in the MCCTL register is ignored.

MCBLKALLH- Multicast Block All High (0x34C)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKALL	RW	0x0	**Multicast Block All.** Each bit in this field corresponds to one of the upper 32 multicast groups (e.g., bit 0 corresponds to multicast group 32, bit 1 corresponds to multicast group 33, and so on). When a bit is set in this field for an enabled multicast group, the PCI-to-PCI bridge associated with this register is blocked from forwarding multicast TLPs associated with that multicast group received on that port. This is an ingress port function performed on received TLPs. The value of bits greater than NUMGROUP in the MCCTL register is ignored.

MCBLKUTL- Multicast Block Untranslated Low (0x350)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKUT	RW	0x0	**Multicast Block Untranslated.** Each bit in this field corresponds to one of the lower 32 multicast groups (e.g., bit 0 corresponds to multicast group 0, bit 1 corresponds to multicast group 1, and so on). When a bit is set in this field for an enabled multicast group, the function associated with this register is blocked from forwarding untranslated multicast TLPs associated with that multicast group received on that port. This is an ingress port function performed on received TLPs. The value of bits greater than NUMGROUP in the MCCTL register is ignored.

Notes

MCBLKUTH - Multicast Block Untranslated High (0x354)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKUT	RW	0x0	**Multicast Block Untranslated.** Each bit in this field corresponds to one of the upper 32 multicast groups (e.g., bit 0 corresponds to multicast group 32, bit 1 corresponds to multicast group 33, and so on). When a bit is set in this field for an enabled multicast group, the function associated with this register is blocked from forwarding untranslated multicast TLPs associated with that multicast group received on that port. This is an ingress port function performed on received TLPs. The value of bits greater than NUMGROUP in the MCCTL register is ignored.

MCOVRBARL- Multicast Overlay Base Address Low (0x358)

Bit Field	Field Name	Type	Default Value	Description
5:0	OVRSIZE	RW	0x0	**Overlay Size.** This field specifies the size in bytes of the overlay aperture as a power of 2. When the value in this field is less than six, the overlay mechanism is disabled.
31:6	MCOVRBARL	RW	0x0	**Multicast Overlay BAR Low.** This field specifies the lower 24-bits (i.e., bits 6 through 31) of the multicast overlay base address.

MCOVRBARH- Multicast Overlay Base Address High (0x35C)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCOVRBARH	RW	0x0	**Multicast Overlay BAR High.** This field specifies the upper 32-bits (i.e., bits 32 through 63) of the multicast overlay base address.

Notes

Proprietary Port Specific Registers

Notes

Port Control Register

PORTCTL - Port Control (0x400)

Bit Field	Field Name	Type	Default Value	Description
0	EWRRPA	RW	0x0 SWSticky	**Enable WRR Port Arbitration.** When this bit is set, port arbitration selection in the port's VC Capability structure is ignored and arbitration is done using a weighted round robin (WRR) algorithm controlled with proprietary count registers. When this bit is cleared, port arbitration is done using the port arbitration selection in the port's VC Capability structure (i.e., hardware fixed round robin). 0x0 - Hardware fixed round robin 0x1 - Weighted Round Robin Refer to section Proprietary Weighted Round Robin (WRR) Arbitration on page 4-8 for further details.
15:1	Reserved	RO	0x0	Reserved field.

Upstream PCI-to-PCI Bridge Interrupt and Signaling

P2PINTSTS - PCI-to-PCI Bridge Interrupt Status (0x404)

Bit Field	Field Name	Type	Default Value	Description
2:0	Reserved	RO	0x0	Reserved field.
3	SEVENT	RW1C	0x0	**Switch Event.** This bit is set in an upstream port whenever an unmasked switch event is generated to the partition (i.e., when the corresponding event bit in the SESTS register transitions from 0x0 to 0x1). Refer to section Switch Events on page 16-1 for details. This bit is read-only with a value of zero in downstream switch ports.
4	FMCI	RW1C	0x0	**Failover Mode Change Initiated** This bit is set in an upstream port whenever failover is enabled in the partition associated with this port (i.e., the FEN bit is set in the corresponding SWPARTxCTL register) and a failover mode change is initiated by the corresponding failover capability structure (i.e., the FMCI bit in the FCAPxSTS register transitions from 0x0 to 0x1). This bit is read-only with a value of zero in downstream switch ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	FMCC	RW1C	0x0	**Failover Mode Change Completed** This bit is set in an upstream port whenever failover is enabled in the partition associated with this port (i.e., the FEN bit is set in the corresponding SWPARTxCTL register) and a failover mode change is completed by the corresponding failover capability structure (i.e., the FMCC bit in the FCAPxSTS register transitions from 0x0 to 0x1). This bit is read-only with a value of zero in downstream switch ports.
6	Reserved	RO	0x0	Reserved field.
7	TMPSENSOR	RW1C	0x0	**Temperature Sensor Alarm.** This bit is set when a temperature sensor alarm is triggered (i.e., one of the temperature threshold bits in the TMPSTS register transitions from 0x0 to 0x1, and the corresponding bit is enabled in the TMPCTL register). This bit is read-only with a value of zero in downstream switch ports.
31:8	Reserved	RO	0x0	Reserved field.

P2PINTMSK - PCI-to-PCI Bridge Interrupt Mask (0x408)

Bit Field	Field Name	Type	Default Value	Description
2:0	Reserved	RO	0x0	Reserved field.
3	SEVENT	RW	0x1	**Switch Event.** When this bit is set in an upstream port, the corresponding bit in the P2PINTSTS register is masked from generating an interrupt. This bit is read-only with a value of zero in downstream switch ports.
4	FMCI	RW	0x1	**Failover Mode Change Initiated.** When this bit is set in an upstream port, the corresponding bit in the P2PINTSTS register is masked from generating an interrupt. This bit is read-only with a value of zero in downstream switch ports.
5	FMCC	RW	0x1	**Failover Mode Change Completed.** When this bit is set in an upstream port, the corresponding bit in the P2PINTSTS register is masked from generating an interrupt. This bit is read-only with a value of zero in downstream switch ports.
6	Reserved	RO	0x0	Reserved field.
7	TMPSENSOR	RW	0x1	**Temperature Sensor Alarm.** When this bit is set, the corresponding bit in the P2PINTSTS register is masked from generating an interrupt. This bit is read-only with a value of zero in downstream switch ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:8	Reserved	RO	0x0	Reserved field.

P2PSDATA - PCI-to-PCI Bridge Signal Data (0x410)

Bit Field	Field Name	Type	Default Value	Description
31:0	SDATA	RW	0x0 SWSticky	**Switch Signal Data.** This is a general 32-bit read write field that may be used in conjunction with switch signals. This field is read-only with a value of zero in downstream switch ports.

P2PGSIGNAL - PCI-to-PCI Bridge Global Signal (0x414)

Bit Field	Field Name	Type	Default Value	Description
0	GSIGNAL	RW	0x0	**Global Signal.** Writing a one to a bit in this field generates a switch signal to the partition associated with this port. This results in the bit corresponding to the partition being set in the Global Signal (GSIGNAL) field in the Switch Event Global Signal Status (SEGSIGSTS) register. This field always returns a value of zero when read. This bit is read-only with a value of zero in downstream switch ports.
31:1	Reserved	RO	0x0	Reserved field.

Port AER Mask Register

PAERMSK - Port AER Mask (0x424)

Bit Field	Field Name	Type	Default Value	Description
0	DLPERR	RW	0x0 Sticky	**Data Link Protocol Error Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
1	SDOENERR	RW	0x0 Sticky	**Surprise Down Error Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
2	POISONED	RW	0x0 Sticky	**Poisoned TLP Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
3	COMPTO	RW	0x0 Sticky	**Completion Timeout Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	CABORT	RW	0x0 Sticky	**Completer Abort Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
5	UECOMP	RW	0x0 Sticky	**Unexpected Completion Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
6	RCVOVR	RW	0x0 Sticky	**Receiver Overflow Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
7	MALFORMED	RW	0x0 Sticky	**Malformed TLP Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
8	ECRC	RW	0x0 Sticky	**ECRC Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
9	UR	RW	0x0 Sticky	**UR Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
10	ACSV	RW	0x0 Sticky	**ACS Violation Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
11	Reserved	RO	0x0	Reserved field.
12	MCBLKTLP	RW	0x0 Sticky	**MC Blocked TLP Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
13	RCVERR	RW	0x0 Sticky	**Receiver Error Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
14	BADTLP	RW	0x0 Sticky	**Bad TLP Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
15	BADDLLP	RW	0x0 Sticky	**Bad DLLP Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
16	RPLYROVR	RW	0x0 Sticky	**Replay Number Rollover Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
17	RPLYTO	RW	0x0 Sticky	**Replay Timer Timeout Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
18	ADVISORYNF	RW	0x0 Sticky	**Advisory Non-Fatal Error Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
19	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
20	HLO	RW	0x0 Sticky	**Header Log Overflow Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.
30:21	Reserved	RO	0x0	Reserved field.
31	IE	RW	0x0 Sticky	**Internal Error Mask.** When this bit is set, the corresponding bit in the internal, non-software visible PAERSTS register is masked.

Port Slot Control

PCIESCTLIV - PCI Express Slot Control Initial Value (0x430)

Bit Field	Field Name	Type	Default Value	Description
0	ABPE	RW	0x0 SWSticky	**Attention Button Pressed Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
1	PFDE	RW	0x0 SWSticky	**Power Fault Detected Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
2	MRLSCE	RW	0x0 SWSticky	**MRL Sensor Change Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	PDCE	RW	0x0 SWSticky	**Presence Detected Changed Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
4	CCIE	RW	0x0 SWSticky	**Command Complete Interrupt Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
5	HPIE	RW	0x0 SWSticky	**Hot Plug Interrupt Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
7:6	AIC	RW	0x3 SWSticky	**Attention Indicator Control.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
9:8	PIC	RW	0x1 SWSticky	**Power Indicator Control.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.

Notes

Bit Field	Field Name	Type	Default Value	Description
10	PCC	RW	0x0 SWSticky	**Power Controller Control.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
11	Reserved	RO	0x0	Reserved field.
12	DLLLASCE	RW	0x0 SWSticky	**Data Link Layer Link Active State Change Enable.** This field contains the initial value of the corresponding field in the PCI Express Slot Control (PCIESCTL) register when the corresponding slot or hot-plug capability is enabled. The intent of this field is to allow the initial value of the corresponding field in the PCIESCTL register to be controlled following a partition fundamental reset. Refer to the description of the corresponding field in the PCIESCTL register for further details.
15:13	Reserved	RO	0x0	Reserved field.

Internal Error Control and Status Registers

These registers control the enabling, logging, and signaling of internal errors associated with the port. Refer to section Internal Errors on page 4-16 for details.

IERRORCTL - Internal Error Reporting Control (0x480)

Bit Field	Field Name	Type	Default Value	Description
0	IERROREN	RW	0x1 SWSticky	Internal Error Reporting Enable When this bit is set, internal error reporting is enabled and reported through AER. Refer to section Internal Errors on page 4-16 for details.
31:1	Reserved	RO	0x0	Reserved field.

IERRORSTS0 - Internal Error Reporting Status 0 (0x484)

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW1C	0x0 SWSticky	**IFB Posted TLP Time-Out.** This bit is set when a posted TLP time-out is detected in the IFB.
1	IFBNPTLPTO	RW1C	0x0 SWSticky	**IFB Non-Posted TLP Time-Out.** This bit is set when a non-posted TLP time-out is detected in the IFB.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	IFBCPTLPTO	RW1C	0x0 SWSticky	**IFB Completion TLP Time-Out.** This bit is set when a completion time-out is detected in the IFB.
3	Reserved	RO	0x0	Reserved field.
4	EFBPTLPTO	RW1C	0x0 SWSticky	**EFB Posted TLP Time-Out.** This bit is set when a posted TLP time-out is detected in the EFB.
5	EFBNPTLPTO	RW1C	0x0 SWSticky	**EFB Non-Posted TLP Time-Out.** This bit is set when a non-posted TLP time-out is detected in the EFB.
6	EFBCPTLPTO	RW1C	0x0 SWSticky	**EFB Completion TLP Time-Out.** This bit is set when a completion time-out is detected in the EFB.
7	IFBDATSBE	RW1C	0x0 SWSticky	**IFB Data Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the IFB data RAM.
8	IFBDATDBE	RW1C	0x0 SWSticky	**IFB Data Double Bit Error.** This bit is set when a double bit ECC error is detected in the IFB data RAM.
9	IFBCTLSBE	RW1C	0x0 SWSticky	**IFB Control Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the IFB control RAM.
10	IFBCTLDBE	RW1C	0x0 SWSticky	**IFB Control Double Bit Error.** This bit is set when a double bit ECC error is detected in the IFB control RAM.
11	EFBDATSBE	RW1C	0x0 SWSticky	**EFB Data Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the EFB data RAM.
12	EFBDATDBE	RW1C	0x0 SWSticky	**EFB Data Double Bit Error.** This bit is set when a double bit ECC error is detected in the EFB data RAM.
13	EFBCTLSBE	RW1C	0x0 SWSticky	**EFB Control Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the EFB control RAM.
14	EFBCTLDBE	RW1C	0x0 SWSticky	**EFB Control Double Bit Error.** This bit is set when a double bit ECC error is detected in the EFB control RAM.
15	E2EPE	RW1C	0x0 SWSticky	**End-to-End Data Path Parity Error.** This bit is set when an end-to-end data path parity error is detected.
16	Reserved	RW1C	0x0 SWSticky	Reserved field.
17	RBCTLSBE	RW1C	0x0 SWSticky	**Replay Buffer Control Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the Replay Buffer's control RAM.

Notes

Bit Field	Field Name	Type	Default Value	Description
18	RBCTLDBE	RW1C	0x0 SWSticky	**Replay Buffer Control Double Bit Error.** This bit is set when a double bit ECC error is detected in the Replay Buffer's control RAM.
19	DIFBPTLPTO	RW1C	0x0 SWSticky	**DMA IFB Posted TLP Time-Out.** This bit is set when a posted TLP time-out is detected in the DMA IFB. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
20	DIFBNPTLPTO	RW1C	0x0 SWSticky	D**MA IFB Non-Posted TLP Time-Out.** This bit is set when a non-posted TLP time-out is detected in the DMA IFB. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
21	DIFBCPTLPTO	RW1C	0x0 SWSticky	**DMA IFB Completion TLP Time-Out.** This bit is set when a completion time-out is detected in the DMA IFB. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW1C	0x0 SWSticky	**DMA IFB Data Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the DMA IFB data RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
24	DIFBDATDBE	RW1C	0x0 SWSticky	**DMA IFB Data Double Bit Error.** This bit is set when a double bit ECC error is detected in the DMA IFB data RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
25	DIFBCTLSBE	RW1C	0x0 SWSticky	**DMA IFB Control Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the DMA IFB control RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
26	DIFBCTLDBE	RW1C	0x0 SWSticky	**DMA IFB Control Double Bit Error.** This bit is set when a double bit ECC error is detected in the DMA IFB control RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
27	DEFBSBE	RW1C	0x0 SWSticky	**DMA EFB Single Bit Error.** This bit is set when a single bit ECC error is detected and corrected in the DMA EFB RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.
28	DEFBDBE	RW1C	0x0 SWSticky	**DMA EFB Double Bit Error.** This bit is set when a double bit ECC error is detected in the DMA EFB RAM. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.

Notes

Bit Field	Field Name	Type	Default Value	Description
30:29	Reserved	RO	0x0	Reserved field.
31	DE2EPE	RW1C	0x0 SWSticky	**DMA End-to-End Data Path Parity Error.** This bit is set when an end-to-end data path parity error is detected by the DMA. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit is hardwired to 0x0.

IERRORSTS1 - Internal Error Reporting Status 1 (0x488)

Bit Field	Field Name	Type	Default Value	Description
0	P0AER	RW1C	0x0 SWSticky	**Port 0 AER Error.** This bit is at the time that port 0 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
1	Reserved	RO	0x0	Reserved field.
2	P2AER	RW1C	0x0 SWSticky	**Port 2 AER Error.** This bit is at the time that port 2 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
3	Reserved	RO	0x0	Reserved field.
4	P4AER	RW1C	0x0 SWSticky	**Port 4 AER Error.** This bit is at the time that port 4 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
5	Reserved	RO	0x0	Reserved field.
6	P6AER	RW1C	0x0 SWSticky	**Port 6 AER Error.** This bit is at the time that port 6 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
7	Reserved	RO	0x0	Reserved field.
8	P8AER	RW1C	0x0 SWSticky	**Port 8 AER Error.** This bit is at the time that port 8 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
11:9	Reserved	RO	0x0	Reserved field.
12	P12AER	RW1C	0x0 SWSticky	**Port 12 AER Error.** This bit is at the time that port 12 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
15:13	Reserved	RO	0x0	Reserved field.
16	P16AER	RW1C	0x0 SWSticky	**Port 16 AER Error.** This bit is at the time that port 16 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.

Notes

Bit Field	Field Name	Type	Default Value	Description
19:17	Reserved	RO	0x0	Reserved field.
20	P20AER	RW1C	0x0 SWSticky	**Port 20 AER Error.** This bit is at the time that port 20 detects an AER error in one of its functions and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
31:21	Reserved	RO	0x0	Reserved field.

IERRORSEV0 - Internal Error Reporting Severity 0 (0x48C)

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW	0x1 SWSticky	**IFB Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
1	IFBNPTLPTO	RW	0x0 SWSticky	**IFB Non-Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
2	IFBCPTLPTO	RW	0x0 SWSticky	**IFB Completion TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
3	Reserved	RO	0x0	Reserved field.
4	EFBPTLPTO	RW	0x1 SWSticky	**EFB Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
5	EFBNPTLPTO	RW	0x0 SWSticky	**EFB Non-Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
6	EFBCPTLPTO	RW	0x0 SWSticky	**EFB Completion TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
7	IFBDATSBE	RW	0x0 SWSticky	**IFB Data Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	IFBDATDBE	RW	0x1 SWSticky	**IFB Data Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
9	IFBCTLSBE	RW	0x0 SWSticky	**IFB Control Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
10	IFBCTLDBE	RW	0x1 SWSticky	**IFB Control Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
11	EFBDATSBE	RW	0x0 SWSticky	**EFB Data Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
12	EFBDATDBE	RW	0x1 SWSticky	**EFB Data Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
13	EFBCTLSBE	RW	0x0 SWSticky	**EFB Control Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
14	EFBCTLDBE	RW	0x1 SWSticky	**EFB Control Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
15	E2EPE	RW	0x1 SWSticky	**End-to-End Data Path Parity Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
16	ULD	RW	0x0 SWSticky	**Unreliable Link Detected.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
17	RBCTLSBE	RW	0x0 SWSticky	**Replay Buffer Control Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.

Notes

Bit Field	Field Name	Type	Default Value	Description
18	RBCTLDBE	RW	0x1 SWSticky	**Replay Buffer Control Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
19	DIFBPTLPTO	RW	0x1 SWSticky	**DMA IFB Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
20	DIFBNPTLPTO	RW	0x0 SWSticky	**DMA IFB Non-Posted TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
21	DIFBCPTLPTO	RW	0x0 SWSticky	**DMA IFB Completion TLP Time-Out.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW	0x0 SWSticky	**DMA IFB Data Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
24	DIFBDATDBE	RW	0x1 SWSticky	**DMA IFB Data Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
25	DIFBCTLSBE	RW	0x0 SWSticky	**DMA IFB Control Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
26	DIFBCTLDBE	RW	0x1 SWSticky	**DMA IFB Control Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
27	DEFBDATSBE	RW	0x0 SWSticky	**DMA EFB Data Single Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
28	DEFBDATDBE	RW	0x1 SWSticky	**DMA EFB Data Double Bit Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
30:29	Reserved	RO	0x0	Reserved field.
31	DE2EPE	RW	0x1 SWSticky	**DMA End-to-End Data Path Parity Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

IERRORSEV1 - Internal Error Reporting Severity 1 (0x490)

Bit Field	Field Name	Type	Default Value	Description
0	P0AER	RW	0x0 SWSticky	**Port 0 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.

Notes

Bit Field	Field Name	Type	Default Value	Description
1	Reserved	RO	0x0	Reserved field.
2	P2AER	RW	0x0 SWSticky	**Port 2 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
3	Reserved	RO	0x0	Reserved field.
4	P4AER	RW	0x0 SWSticky	**Port 4 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
5	Reserved	RO	0x0	Reserved field.
6	P6AER	RW	0x0 SWSticky	**Port 6 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
7	Reserved	RO	0x0	Reserved field.
8	P8AER	RW	0x0 SWSticky	**Port 8 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
11:9	Reserved	RO	0x0	Reserved field.
12	P12AER	RW	0x0 SWSticky	**Port 12 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
15:13	Reserved	RO	0x0	Reserved field.
16	P16AER	RW	0x0 SWSticky	**Port 16 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
19:17	Reserved	RO	0x0	Reserved field.
20	P20AER	RW	0x0 SWSticky	**Port 20 AER Error.** This bit controls how an error of the corresponding type is reported. When this bit is set, the error is reported as an uncorrectable internal error. When this bit is cleared, the error is reported as an correctable internal error.
31:21	Reserved	RO	0x0	Reserved field.

Notes

IERRORTST0 - Internal Error Reporting Test 0 (0x494)

This register can be used to emulate the occurrence of internal errors. Each bit in this register corresponds to an internal error. Writing a one to a bit in this register causes the port to log the corresponding internal error as if the error had actually occurred (e.g., the error is logged in the IERRORSTS0 register, logged by AER, etc.). Refer to section Internal Errors on page 4-16 for details.

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW	0x0	**IFB Posted TLP Time-Out.** This bit always returns a value of zero when read.
1	IFBNPTLPTO	RW	0x0	**IFB Non-Posted TLP Time-Out.** This bit always returns a value of zero when read.
2	IFBCPTLPTO	RW	0x0	**IFB Completion TLP Time-Out.** This bit always returns a value of zero when read.
3	Reserved	RO	0x0	Reserved field.
4	EFBPTLPTO	RW	0x0	**EFB Posted TLP Time-Out.** This bit always returns a value of zero when read.
5	EFBNPTLPTO	RW	0x0	**EFB Non-Posted TLP Time-Out.** This bit always returns a value of zero when read.
6	EFBCPTLPTO	RW	0x0	**EFB Completion TLP Time-Out.** This bit always returns a value of zero when read.
7	IFBDATSBE	RW	0x0	**IFB Data Single Bit Error.** This bit always returns a value of zero when read.
8	IFBDATDBE	RW	0x0	**IFB Data Double Bit Error.** This bit always returns a value of zero when read.
9	IFBCTLSBE	RW	0x0	**IFB Control Single Bit Error.** This bit always returns a value of zero when read.
10	IFBCTLDBE	RW	0x0	**IFB Control Double Bit Error.** This bit always returns a value of zero when read.
11	EFBDATSBE	RW	0x0	**EFB Data Single Bit Error.** This bit always returns a value of zero when read.
12	EFBDATDBE	RW	0x0	**EFB Data Double Bit Error.** This bit always returns a value of zero when read.
13	EFBCTLSBE	RW	0x0	**EFB Control Single Bit Error.** This bit always returns a value of zero when read.
14	EFBCTLDBE	RW	0x0	**EFB Control Double Bit Error.** This bit always returns a value of zero when read.
15	E2EPE	RW	0x0	**End-to-End Data Path Parity Error.** This bit always returns a value of zero when read.
16	ULD	RW	0x0	**Unreliable Link Detected.** This bit always returns a value of zero when read
17	RBCTLSBE	RW	0x0	**Replay Buffer Control Single Bit Error.** This bit always returns a value of zero when read.
18	RBCTLDBE	RW	0x0	**Replay Buffer Control Double Bit Error.** This bit always returns a value of zero when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
19	DIFBPTLPTO	RW	0x0	**DMA IFB Posted TLP Time-Out.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
20	DIFBNPTLPTO	RW	0x0	**DMA IFB Non-Posted TLP Time-Out.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
21	DIFBCPTLPTO	RW	0x0	**DMA IFB Completion TLP Time-Out.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW	0x0	**DMA IFB Data Single Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
24	DIFBDATDBE	RW	0x0	**DMA IFB Data Double Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
25	DIFBCTLSBE	RW	0x0	**DMA IFB Control Single Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
26	DIFBCTLDBE	RW	0x0	**DMA IFB Control Double Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
27	DEFBDATSBE	RW	0x0	**DMA EFB Data Single Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
28	DEFBDATDBE	RW	0x0	**DMA EFB Data Double Bit Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.
30:29	Reserved	RO	0x0	Reserved field.
31	DE2EPE	RW	0x0	**DMA End-to-End Data Path Parity Error.** This bit always returns a value of zero when read. This bit is only applicable for ports that contain a DMA function.

Notes

IERRORTST1 - Internal Error Reporting Test 1 (0x498)

This register can be used to emulate the occurrence of internal errors. Each bit in this register corresponds to an internal error. Writing a one to a bit in this register causes the port to log the corresponding internal error as if the error had actually occurred (e.g., the error is logged in the IERRORSTS1 register, logged by AER, etc.) Refer to section Internal Errors on page 4-16 for details.

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

P2PIERRORMSK0 - PCI-to-PCI Bridge Internal Error Reporting Mask 0 (0x4A0)

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW	0x0 SWSticky	**IFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	IFBNPTLPTO	RW	0x0 SWSticky	**IFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
2	IFBCPTLPTO	RW	0x0 SWSticky	**IFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3	Reserved	RO	0x0	Reserved field.
4	EFBPTLPTO	RW	0x0 SWSticky	**EFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	EFBNPTLPTO	RW	0x0 SWSticky	**EFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
6	EFBCPTLPTO	RW	0x0 SWSticky	**EFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
7	IFBDATSBE	RW	0x0 SWSticky	**IFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
8	IFBDATDBE	RW	0x0 SWSticky	**IFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
9	IFBCTLSBE	RW	0x0 SWSticky	**IFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
10	IFBCTLDBE	RW	0x0 SWSticky	**IFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11	EFBDATSBE	RW	0x0 SWSticky	**EFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
12	EFBDATDBE	RW	0x0 SWSticky	**EFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
13	EFBCTLSBE	RW	0x0 SWSticky	**EFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
14	EFBCTLDBE	RW	0x0 SWSticky	**EFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	E2EPE	RW	0x0 SWSticky	**End-to-End Data Path Parity Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
16	ULD	RW	0x0 SWSticky	**Unreliable Link Detected.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
17	RBCTLSBE	RW	0x0 SWSticky	**Replay Buffer Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
18	RBCTLDBE	RW	0x0 SWSticky	**Replay Buffer Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19	DIFBPTLPTO	RW	0x1 SWSticky	**DMA IFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
20	DIFBNPTLPTO	RW	0x1 SWSticky	**DMA IFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
21	DIFBCPTLPTO	RW	0x1 SWSticky	**DMA IFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW	0x1 SWSticky	**DMA IFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
24	DIFBDATDBE	RW	0x1 SWSticky	**DMA IFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
25	DIFBCTLSBE	RW	0x1 SWSticky	**DMA IFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
26	DIFBCTLDBE	RW	0x1 SWSticky	**DMA IFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
27	DEFBDATSBE	RW	0x1 SWSticky	**DMA EFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
28	DEFBDATDBE	RW	0x1 SWSticky	**DMA EFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
30:29	Reserved	RO	0x0	Reserved field.
31	DE2EPE	RW	0x1 SWSticky	**DMA End-to-End Data Path Parity Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

P2PIERRORMSK1 - PCI-to-PCI Bridge Internal Error Reporting Mask 1 (0x4A4)

Bit Field	Field Name	Type	Default Value	Description
0	P0AER	RW	0x1 SWSticky	**Port 0 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	Reserved	RO	0x0	Reserved field.
2	P2AER	RW	0x1 SWSticky	**Port 2 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3	Reserved	RO	0x0	Reserved field.
4	P4AER	RW	0x1 SWSticky	**Port 4 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	P6AER	RW	0x1 SWSticky	**Port 6 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
7	Reserved	RO	0x0	Reserved field.
8	P8AER	RW	0x1 SWSticky	**Port 8 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11:9	Reserved	RO	0x0	Reserved field.
12	P12AER	RW	0x1 SWSticky	**Port 12 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
15:13	Reserved	RO	0x0	Reserved field.
16	P16AER	RW	0x1 SWSticky	**Port 16 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19:17	Reserved	RO	0x0	Reserved field.
20	P20AER	RW	0x1 SWSticky	**Port 20 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the PCI-to-PCI bridge function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
31:21	Reserved	RO	0x0	Reserved field.

Physical Layer Control and Status Registers

This section describes the port's physical layer control and status registers. As described in section SerDes Numbering and Port Association on page 8-1, a port's association with a SerDes quad depends on the configuration of the corresponding stack. SerDes configuration and status registers listed in this section apply to the SerDes quad lane(s) associated with the port.

Also, a port's maximum supported link width also depends on the product option. Register fields which have per-lane control or status bits are only valid for lanes included within the port's maximum supported link width. For example, if a port's maximum link width is set to x2 (i.e., the port's MAXLNKWDTH field in the PCIELCAP register is set to 0x2), lane control and status bits for lanes 0 and 1 are valid, while lane control and status bits for lanes 2 and above are invalid and may take on undefined values.

Notes

SERDESCFG - SerDes Configuration (0x510)

Bit Field	Field Name	Type	Default Value	Description
7:0	RCVD_OVRD	RW	0x0 SWSticky	**Receiver Detect Override.** Each bit in this register corresponds to a lane associated with this port. Setting this bit causes the lane associated with this bit to indicate that a receiver has been detected on the line. This field is not valid when the port operates in SerDes Test Mode.
15:8	Reserved	RO	0x0	Reserved field.
16	LSE	RW	0x0 SWSticky	**Low-Swing Mode Enable.** When set, this bit enables Low-Swing mode operation at the SerDes Transmit logic for the lanes associated with the port. Please refer to section Low-Swing Transmitter Voltage Mode on page 8-12 for further details. 0x0 - Full-Swing Mode 0x1 - Low-Swing Mode
31:17	Reserved	RO	0x0	Reserved field.

LANESTS0 - Lane Status 0 (0x51C)

Bit Field	Field Name	Type	Default Value	Description
7:0	PDE	RW1C	0x0	**Phy Disparity Error.** Each bit in this field corresponds to a lane associated with the port. A bit is set when an 8B10B coding violation has resulted in a running disparity error in the received data stream. A bit can only be set when the LTSSM is in the L0 state.
15:8	Reserved	RO	0x0	Reserved field.
23:16	E8B10B	RW1C	0x0	**8B10B Error.** Each bit in this field corresponds to a lane associated with the port. A bit is set when an 8B10B decode error is detected in the received data stream. A bit can only be set when the LTSSM is in the L0, Configuration, Disabled, or Hot Reset states.
31:24	Reserved	RO	0x0	Reserved field.

Notes

LANESTS1 - Lane Status 1 (0x520)

Bit Field	Field Name	Type	Default Value	Description
7:0	UND	RW1C	0x0 Sticky	**Receiver Underflow Detected.** Each bit in this field corresponds to a lane associated with the port. A bit is set when the corresponding link receiver is unable to compensate for clock variance between link partners and has inserted one or more zero bytes into the stream. A bit can only be set when the LTSSM is in the L0 state.
15:8	Reserved	RO	0x0	Reserved field.
23:16	OVR	RW1C	0x0 Sticky	**Receiver Overflow Detected.** Each bit in this field corresponds to a lane associated with the port. A bit is set when the corresponding link receiver is unable to compensate for clock variance between link partners and has dropped one or more bytes. A bit can only be set when the LTSSM is in the L0 state.
31:24	Reserved	RO	0x0	Reserved field.

PHYLCFG0 - Phy Link Configuration 0 (0x530)

Bit Field	Field Name	Type	Default Value	Description
7:0	Reserved	RO	0x0	Reserved field.
8	G1CME	RW	0x0 SWSticky	**Gen 1 Compatibility Mode Enable.** When this bit is set, the PHY operates in Gen 1 Compatibility mode. In this mode, the PHY does not set training set bits not defined in the PCI Express 1.1 specification. Please refer to section Gen 1 Compatibility Mode on page 7-18 for further details.
10:9	Reserved	RO	0x0	Reserved field.
11	CLINKDIS	RW	0x0 SWSticky	**Disable Crosslink.** When this bit is set, crosslink link training is disabled and the device link trains as though crosslink were not implemented. Please refer to section Crosslink on page 7-17 for further details.
13:12	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	ILSCC	RW	Down-stream Switch Port: 0x0 Other: 0x1 MSWSticky	**Initial Link Speed Change Control.** This field determines whether a port automatically initiates a speed change to Gen 2 speed, if Gen 2 speed is permissible, after initial entry to L0 from Detect. 0x0 - (automatic) Automatically initiate speed change to Gen 2 speed, if permissible, after the first entry to L0 from Detect. 0x1 - (nochange) Do not automatically initiate a speed change to Gen 2 speed, stay in Gen 1 speed. Note that the initial value of this field depends on the port operating mode. Also, note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change. Finally, note that when cross-linking an upstream port with a link-partner's upstream port, neither port may automatically initiate a link speed change to Gen 2, thereby resulting in a Gen 1 link. By clearing this bit, the upstream port will initiate the link transition to Gen 2 speed.
15	DLWUC	RW	0x0 SWSticky	**Disable Link Width Upconfiguration Capability.** When this bit is set, the port does not advertise support for link width upconfiguration in the training sets it issues during the Configuration state. Otherwise, the port advertises support for link width upconfiguration during the Configuration state. After modifying this bit, it is recommended that the port's link be fully retrained by setting the FLRET bit in the PHYLSTATE0 register, to force the PHY LTSSM to transition to the Configuration state.
31:16	Reserved	RO	0x0	Reserved field.

PHYLSTATE0 - Phy Link State 0 (0x540)

Bit Field	Field Name	Type	Default Value	Description
30:0	Reserved	RO	0x0	Reserved field.
31	FLRET	RW	0x0	**Full Link Retrain.** Writing a one to this field initiates full link retraining by directing the PHY LTSSM into the DETECT state. This bit always returns zero when read. Writing of a one to this bit always results in the device returning a completion to the requester before the action specified by this bit takes effect. For an upstream port, writing of a one to this bit always results in the action specified by this bit to take effect after a programmed delay (see section Configuration Register Side-Effects on page 19-3). The device always returns a completion to the requester before the effect of this bit is applied.

Notes

PHYPRBS - Phy PRBS Seed (0x55C)

Bit Field	Field Name	Type	Default Value	Description
15:0	SEED	RW	0xFFFF SWSticky	**Phy PRBS Seed Value.** This field contains the PHY PRBS seed value used for crosslink operation. When the value in this register is modified, the PRBS counter associated with this seed is reset to the seed value and re-starts counting.
31:16	Reserved	RO	0x0	Reserved field.

TLCNTCFG - Transaction Layer Countables Configuration (0x690)

Bit Field	Field Name	Type	Default Value	Description
23:0	Reserved	RO	0x0	Reserved field.
31:24	BUS	RW	0x0	**Bus Number.** Per the PCI Express Base Specification, functions capture the bus number for all type 0 configuration write requests completed by the function. This field contains the last bus number value captured by any of the port functions. Software is allowed to modify this field only when the port operates in 'NT function' or 'NT and DMA function' mode. Modification of this field causes all functions of the port to operate using this bus number (e.g., completions generated by the function(s) use this bus number in the completer ID field of the completion TLP, requests generated by the function(s) use this bus number in the requester ID of the request TLP, etc.) Note that this field is always updated by hardware on completion of a type 0 configuration write request by any of the port functions.

L1ASPMRTC - L1 ASPM Rejection Timer Control (0x710)

Bit Field	Field Name	Type	Default Value	Description
13:0	MTL1ER	RW	0x947 SWSticky	**Minimum Time between L1 Entry Requests.** This field indicates the minimum time (in 250Mhz cycles) that the port waits between detecting consecutive L1 ASPM entry requests. An L1 ASPM entry request consists of a number of PM_L1_Active_State_Request DLLPs followed by an acknowledge (positive or negative) from the downstream switch port receiving the request. The actual time may be calculated by multiplying the value in this field by 4ns. For example, a setting of 0x947 (i.e., 2375 d) corresponds to a time of (2375 cycles * 4ns = 9.5us). Refer to section 5.4.1.2.1 of the PCI Express Base Specification 2.1 for further details on the L1 ASPM Entry rejection protocol.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:14	Reserved	RO	0x0	Reserved field
16	TSCTL	RW	0x1 SWSticky	**Timer Start Control.** Upon rejecting an L1 ASPM entry request from the link partner, the switch port counts an amount of time equal to the value in the MTL1ER field before detecting a new request. This field selects the criteria for starting the timer. 0x0 - Timer starts counting when the port has issued a PM_L1_Active_State_Nak TLP. Reception of PM_L1_Active_State_Request DLLPs from the link partner prior to the expiration of the timer result in the timer being re-started. 0x1 - Timer starts counting when the port has issued a PM_L1_Active_State_Nak TLP. Reception of PM_L1_Active_State_Request DLLPs from the link partner prior to the expiration of the timer are ignored. Once the timer has expired, reception of an incoming PM_L1_Active_State_Request DLLP will be treated as a new request.
31:17	Reserved	RO	0x0	Reserved field.

Request Metering

RMCTL - Requester Metering Control (0x880)

Bit Field	Field Name	Type	Default Value	Description
0	EN	RW	0x0 SWSticky	**Enable.** When this bit is set, request metering is enabled on the corresponding input port. Refer to section Request Metering on page 4-11 for further details.
4:1	OVRFACTOR	RW	0x2 SWSticky	**Overhead Factor.** This field contains the overhead factor used in computing the completion size estimate for memory read requests.
5	Reserved	RO	0x0	Reserved field.
15:6	CNSTLIMIT	RW	0x10 SWSticky	**Constant Limit.** This field is used to control the algorithm used to compute the completion size estimate for non-posted read requests when request metering is enabled. When the number of DWords required in completions to service a non-posted read request is less than or equal to the value in this field, then a constant completion size estimate is used. When the number of DWords required in completions to service a non-posted read request is greater than the value in this field, then the completion size estimate considers the number of individual completion TLPs.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:16	DVADJ	RW	0x0 SWSticky	**Decrement Value Adjustment.** This field contains the adjustment value used to determine the value by which the request metering counter is decremented each 250 MHz clock tick. The value in this field represents a sign-magnitude, fixed-point number with 4 integer bits and 11 fractional bits (i.e., a 1:4:11 format number). The sign bit (i.e., the most significant bit) is 0b0 for positive numbers and 0b1 for negative numbers.

RMCOUNT - Requester Metering Count (0x88C)

Bit Field	Field Name	Type	Default Value	Description
15:0	COUNT	RO	0x0	**Count.** This field contains the requester metering initial counter value for the last non-posted request The request metering counter is a 24-bit counter that represents a fixed point 0:13:11 number (i.e., an unsigned number with 13 integer bits and 11 fractional bits). The value in this field represents an unsigned number with 13 integer bits and 3 fractional bits. The least significant eight fractional bits of the initial counter value are always zero.
31:16	Reserved	RO	0x0	Reserved field.

WRR Port Arbitration Counts

VC0PARBCI0 - VC0 Port Arbiter Counter Initialization 0 (0x890)

Bit Field	Field Name	Type	Default Value	Description
7:0	P0IC	RW	See Description SWSticky	**Port 0 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 0. The initial value of this field is 0xFF in Port 0, and 0x0 in all other ports. In Port 0, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
15:8	Reserved	RO	0x0	Reserved field.
23:16	P2IC	RW	See Description SWSticky	**Port 2 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 2. The initial value of this field is 0xFF in Port 2, and 0x0 in all other ports. In Port 2, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:24	Reserved	RO	0x0	Reserved field.

VC0PARBCI1 - VC0 Port Arbiter Counter Initialization 1 (0x894)

Bit Field	Field Name	Type	Default Value	Description
7:0	P4IC	RW	See Description SWSticky	**Port 4 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 4. The initial value of this field is 0xFF in Port 4, and 0x0 in all other ports. In Port 4, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
15:8	Reserved	RO	0x0	Reserved field.
23:16	P6IC	RW	See Description SWSticky	**Port 6 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 6. The initial value of this field is 0xFF in Port 6, and 0x0 in all other ports. In Port 6, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
31:24	Reserved	RO	0x0	Reserved field.

VC0PARBCI2 - VC0 Port Arbiter Counter Initialization 2 (0x898)

Bit Field	Field Name	Type	Default Value	Description
7:0	P8IC	RW	See Description SWSticky	**Port 8 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 8. The initial value of this field is 0xFF in Port 8, and 0x0 in all other ports. In Port 8, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
31:8	Reserved	RO	0x0	Reserved field.

VC0PARBCI3 - VC0 Port Arbiter Counter Initialization 3 (0x89C)

Bit Field	Field Name	Type	Default Value	Description
7:0	P12IC	RW	See Description SWSticky	**Port 12 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 12. The initial value of this field is 0xFF in Port 12, and 0x0 in all other ports. In Port 12, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:8	Reserved	RO	0x0	Reserved field.

VC0PARBCI4 - VC0 Port Arbiter Counter Initialization 4 (0x8A0)

Bit Field	Field Name	Type	Default Value	Description
7:0	P16IC	RW	See Description SWSticky	**Port 16 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 16. The initial value of this field is 0xFF in Port 16, and 0x0 in all other ports. In Port 16, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
31:8	Reserved	RO	0x0	Reserved field.

VC0PARBCI5 - VC0 Port Arbiter Counter Initialization 5 (0x8A4)

Bit Field	Field Name	Type	Default Value	Description
7:0	P20IC	RW	See Description SWSticky	**Port 20 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to port 20. The initial value of this field is 0xFF in Port 20, and 0x0 in all other ports. In Port 20, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
31:8	Reserved	RO	0x0	Reserved field.

VC0PARBCI6 - VC0 Port Arbiter Counter Initialization 6 (0x8A8)

Bit Field	Field Name	Type	Default Value	Description
7:0	DMA0IC	RW	See Description SWSticky	**DMA Module 0 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to DMA module 0. The initial value of this field is 0xFF in Port 0 (i.e., where DMA module 0 is logically located), and 0x0 in all other ports. In Port 0, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.
15:8	DMA1IC	RW	See Description SWSticky	**DMA Module 1 Initial Count.** This field contains the initial value of the WRR port arbitration count corresponding to DMA module 1. The initial value of this field is 0xFF in Port 8 (i.e., where DMA module 1 is logically located), and 0x0 in all other ports. In Port 8, this field must never be set to 0x0. Refer to section Cut-Through Routing on page 4-9 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:16	Reserved	RO	0x0	Reserved field.

Non-Transparent Multicast Overlay

NTMCC - NT Multicast Control (0x900)

Bit Field	Field Name	Type	Default Value	Description
0	NTMCTEN	RW	0x0	**NT Multicast Transmit Enable.** This bit, when set, enables the transmission of NT multicast TLPs by the port. When cleared, the port does not transmit NT multicast TLPs. Refer to section Non-Transparent Multicast Operation on page 17-6 for details.
1	NTMCAOE	RW	0x0	**NT Multicast Address Overlay Enable.** This bit, when set, enables NT multicast NT Multicast address overlay. When cleared, NT multicast address overlay is not performed. This bit only has effect when the NTMCTEN bit in this register is set.
2	NTMCRIDOE	RW	0x0	**NT Multicast Requester ID Overlay Enable.** This bit, when set, enables NT multicast NT Multicast requester ID overlay. When cleared, NT multicast requester ID overlay is not performed. This bit only has effect when the NTMCTEN bit in this register is set.
31:3	Reserved	RO	0x0	Reserved field.

Notes

NTMCOVR[3:0]C - NT Multicast Overlay x Configuration

Bit Field	Field Name	Type	Default Value	Description
7:0	PART	RW	0x0	**Partition Association.** Each bit in this field corresponds to a switch partition (i.e., bit 0 corresponds to partition 0, bit 1 corresponds to partition 1, etc.) When an NT multicast TLP is received on a partition whose corresponding bit is set in this field and a group whose corresponding bit is set in the GROUP field in this register, the NT multicast TLP is subject to NT multicast requester ID overlay as determined by this register and NT multicast address overlay as determined by the corresponding NTMCOVRxBARL/H registers. Setting this field to a value of 0x0 causes no partitions to be associated with this NT multicast overlay control register, in effect disabling the overlay actions controlled by this register. Setting this field to a value of 0xFF causes all partitions to be associated with this NT multicast overlay control register. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.
11:8	GROUP	RW	0x0	**Group Association.** Each bit in this field corresponds to an NT Multicast Group (i.e., bit 0 corresponds to NT Multicast Group 0, bit 1 corresponds to NT Multicast Group 1, etc.) When an NT multicast TLP is received on a group whose corresponding bit is set in this field and a partition whose corresponding bit is set in the PART field in this register, the NT multicast TLP is subject to NT multicast requester ID overlay as determined by this register and NT multicast address overlay as determined by the corresponding NTMCOVRxBARL/H registers. Setting this field to a value of 0x0 causes no NT multicast groups to be associated with this NT multicast overlay control register, in effect disabling the overlay actions controlled by this register. Setting this field to a value of 0xF causes all NT multicast groups to be associated with this NT multicast overlay control register. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.
15:12	Reserved	RO	0x0	Reserved field.
31:16	OVRREQID	RW	0x0	**Overlay Requester ID.** The value of this field replaces the requester ID field of an NT multicast TLP transmitted by this port. This field may be divided into 3 sub-fields: overlay bus, overlay device, and overlay function number. Bits[15:8] in this field correspond to the overlay bus number. Bits[7:3] in this field correspond to the overlay device number. Bits[2:0] correspond to the overlay function number. Refer to section Non-Transparent Multicast Operation on page 17-6 for details.

Notes

NTMCOVR[3:0]BARL - NT Multicast Overlay x Base Address Low

Bit Field	Field Name	Type	Default Value	Description
5:0	OVRSIZE	RW	0x0	**Overlay Size.** This field specifies the size in bytes of the overlay aperture as a power of 2. The value in this field must be programmed to six or above. When the value in this field is less than six, the operation is undefined. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.
31:6	MCBARL	RW	0x0	**Multicast Overlay BAR Low.** This field specifies the lower 24-bits (i.e., bits 6 through 31) of the NT multicast overlay base address. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.

NTMCOVR[3:0]BARH - NT Multicast Overlay x Base Address High

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBARH	RW	0x0	**Multicast Overlay BAR High.** This field specifies the upper 32-bits (i.e., bits 32 through 63) of the NT multicast overlay base address. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.

AER Error Emulation

P2PUEEM - PCI-to-PCI Bridge Uncorrectable Error Emulation (0xD90)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x0 SWSticky	**Data Link Protocol Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
5	SDOENERR	RW	0x0 SWSticky	**Surprise Down Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 SWSticky	**Poisoned TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:13	Reserved	RO	0x0	Reserved field.
16	UECOMP	RW	0x0 SWSticky	**Unexpected Completion Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
17	RCVOVR	RW	0x0 SWSticky	**Receiver Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
18	MALFORMED	RW	0x0 SWSticky	**Malformed TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
19	ECRC	RW	0x0 SWSticky	**ECRC Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
20	UR	RW	0x0 SWSticky	**UR Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
21	ACSV	RW	0x0 SWSticky	**ACS Violation Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
22	UIE	RW	0x0 SWSticky	**Uncorrectable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
23	MCBLKTLP	RW	0x0 SWSticky	**MC Blocked TLP Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
30:24	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31	ADVISORYNF	RW	0x0 SWSticky	**Advisory Non-Fatal Error Trigger.** If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible (refer to the PCI Express Base Specification), an advisory non-fatal error is logged an reported in the PCI-to-PCI bridge function's AER capability structure, provided the error severity for the selected uncorrectable error is configured such that the error will be of type non-fatal. If this bit is set together with another error bit in this register for which an advisory non-fatal error is not possible, the operation is undefined. If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible, but the severity of the selected uncorrectable error is fatal, then this bit is ignored and the selected error is logged and reported as a fatal error.

P2PCEEM - PCI-to-PCI Bridge Correctable Error Emulation (0xD94)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 SWSticky	**Receiver Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 SWSticky	**Bad TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
7	BADDLLP	RW	0x0 SWSticky	**Bad DLLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
8	RPLYROVR	RW	0x0 SWSticky	**Replay Number Rollover Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 SWSticky	**Replay Timer Timeout Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
13	Reserved	RO	0x0	Reserved field.
14	CIE	RW	0x0 SWSticky	**Correctable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW	0x0 SWSticky	**Header Log Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
31:16	Reserved	RO	0x0	Reserved field.

Global Address Space Access Registers

GASAADDR - Global Address Space Access Address (0xFF8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
18:2	GADDR	RW	0x0	**Global Address.** This field selects the system address of the register to be accessed via the GASADATA register. The following restrictions apply regarding the programming of this register: 1) The value of this register must not be programmed to point to the address of the GASAADDR or GASADATA register in this or any other function. 2) The value of this register must not be programmed to point to the address of the Extended Configuration Address and Data registers (ECFGADDR and ECFGDATA) in this or any other function. 3) The value of this register must not be programmed to point to the NT Mapping Table Address and Data (NTMTBLADDR and NTMTBLDATA) registers in any NT function. Violations of these rules produce undefined results.
31:19	Reserved	RO	0x0	Reserved field.

GASADATA - Global Address Space Access Data (0xFFC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Data.** A read from this field will return the global space register value pointed to by the GASAADDR register. A write to this field will update the contents of the global space register pointed to by the GASAADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

Notes

NT Endpoint Registers

Notes

Type 0 Configuration Header Registers

VID - Vendor Identification (0x000)

Bit Field	Field Name	Type	Default Value	Description
15:0	VID	RO	0x111D	**Vendor Identification.** This field contains the 16-bit vendor ID value assigned to IDT. See section Vendor ID on page 1-1.

DID - Device Identification (0x002)

Bit Field	Field Name	Type	Default Value	Description
15:0	DID	RO	-	**Device Identification.** This field contains the 16-bit device ID assigned by IDT to this device. See section Device ID on page 1-1.

PCICMD - PCI Command (0x004)

Bit Field	Field Name	Type	Default Value	Description
0	IOAE	RW	0x0	**I/O Access Enable.** When this bit is cleared, the function does not respond to I/O accesses received on its primary bus (i.e., a TLP that targets the function's IO space is treated as an Unsupported Request). 0x0 - (disable) Disable I/O space. 0x1 - (enable) Enable I/O space.
1	MAE	RW	0x0	**Memory Access Enable.** When this bit is cleared, the function does not respond to memory and prefetchable memory space accesses received on its primary bus (i.e., a TLP that targets the function's memory or prefetchable memory space is treated as an Unsupported Request). 0x0 - (disable) Disable memory space. 0x1 - (enable) Enable memory space.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	BME	RW	0x0	**Bus Master Enable.** When this bit is cleared, inter-partition requests are not transmitted by the function. In addition, the function does not issue MSIs. All other requests or completions emitted by this function are not affected by this bit. 0x0 - (disable) Disable transmission of inter-partition requests, as well as MSI generated by the function. 0x1 - (enable) Enable transmission of inter-partition requests, as well as MSI generated by the function.
3	SSE	RO	0x0	**Special Cycle Enable.** Not applicable.
4	MWI	RO	0x0	**Memory Write Invalidate.** Not applicable.
5	VGAS	RO	0x0	**VGA Palette Snoop.** Not applicable.
6	PERRE	RW	0x0	**Parity Error Enable.** This bit controls the logging of poisoned TLPs in the Master Data Parity Error Detected (MDPED) field in the PCISTS register. When this bit is cleared, poisoned TLPs are not reported as master data parity errors in the PCISTS register.
7	ADSTEP	RO	0x0	**Address Data Stepping.** Not applicable.
8	SERRE	RW	0x0	**SERR Enable.** Non-fatal and fatal errors detected by the function are reported to the Root Complex when this bit is set or the bits in the PCI Express Device Control register are set. 0x0 - (disable) Disable non-fatal and fatal error reporting if also disabled in Device Control register. 0x1 - (enable) Enable non-fatal and fatal error reporting.
9	FB2B	RO	0x0	**Fast Back-to-Back Enable.** Not applicable.
10	INTXD	RW	0x0	**INTx Disable.** Controls the ability of the function to generate an INTx interrupt message. When this bit is set, any interrupts generated by this function are negated. This may result in a change in the resolved interrupt state of the function.
15:11	Reserved	RO	0x0	Reserved field.

PCISTS - PCI Status (0x006)

Bit Field	Field Name	Type	Default Value	Description
2:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	INTS	RO	0x0	**INTx Status.** This bit is set when an INTx interrupt is pending from the function.
4	CAPL	RO	0x1	**Capabilities List.** This bit is hardwired to one to indicate that this function implements an extended capability list item.
5	C66MHZ	RO	0x0	**66 MHz Capable.** Not applicable.
6	Reserved	RO	0x0	Reserved field.
7	FB2B	RO	0x0	**Fast Back-to-Back (FB2B).** Not applicable.
8	MDPED	RW1C	0x0	**Master Data Parity Error Detected.** This bit is set by the function when the PERRE bit in the PCICMD register is set and one of the following occurs: 1) The function receives a completion marked as 'poisoned'. 2) The function transmits a poisoned request.
10:9	DEVT	RO	0x0	**DEVSEL# TIming.** Not applicable.
11	STAS	RW1C	0x0	**Signaled Target Abort.** Not applicable since the NT function never issues completions with completer-abort status.
12	RTAS	RW1C	0x0	**Received Target Abort.** This bit is set when the NT function receives a completion with Completer Abort completion status. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a completion with Completer Abort completion status is received by this function.
13	RMAS	RW1C	0x0	**Received Master Abort.** This bit is set when the NT function receives a completion with Unsupported Request completion status. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a completion with Unsupported Request completion status is received by this function.
14	SSE	RW1C	0x0	**Signaled System Error.** This bit is set when the function sends an ERR_FATAL or ERR_NONFATAL message and the SERR Enable (SERRE) bit in the PCICMD register is set. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a fatal or non-fatal error is signaled.
15	DPE	RW1C	0x0	**Detected Parity Error.** This bit is set by the function whenever it receives a poisoned TLP regardless of the state of the PERRE bit in the PCI Command register.

Notes

RID - Revision Identification (0x008)

Bit Field	Field Name	Type	Default Value	Description
7:0	RID	RWL	- SWSticky	**Revision ID.** This field contains the revision identification number for the device. See section Revision ID on page 1-1.

CCODE - Class Code (0x009)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTF	RO	0x00	**Interface.** No standard interface defined.
15:8	SUB	RO	0x80	**Sub Class Code.** This value indicates that the device is classified as 'other'.
23:16	BASE	RO	0x06	**Base Class Code.** This value indicates that the device is a bridge.

CLS - Cache Line Size (0x00C)

Bit Field	Field Name	Type	Default Value	Description
7:0	CLS	RW	0x00	**Cache Line Size.** This field has no effect on the function's operation but may be read and written by software. This field is implemented for compatibility with legacy software.

LTIMER - Latency Time (0x00D)

r

Bit Field	Field Name	Type	Default Value	Description
7:0	LTIMER	RO	0x00	**Latency Timer.** Not applicable.

HDR - Header Type (0x00E)

Bit Field	Field Name	Type	Default Value	Description
7:0	HDR	RO	See Description	**Header Type.** This field indicates the configuration space header type for the NT function (type 0 header). The default value depends on the port's operating mode. If the port operating mode configures the port as a multi-function device, the value of this field is 0x80. Otherwise, the value of this field is 0x00.

Notes

BIST - Built-in Self Test Register (0x00F)

Bit Field	Field Name	Type	Default Value	Description
7:0	BIST	RO	0x0	**BIST.** This value indicates that the function does not implement BIST.

BAR0 - Base Address Register 0 (0x010)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP0 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. The value of this field is determined by the TYPE field in the BARSETUP0 register. When the MEMSI field indicates I/O space, this field is always zero. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP0 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. The value of the SIZE field in the BARSETUP0 register controls which bits in this field may be modified. Bits that cannot be modified are always zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR which in this case is BAR1. See the PCI and PCI Express Base Specifications for more information.

Notes

BAR1 - Base Address Register 1 (0x014)

When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. Otherwise, the BAR format below is used.

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP1 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. The value of this field is determined by the TYPE field in the BARSETUP1 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (reserved) reserved. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP1 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. See the PCI and PCI Express Base Specifications for more information. When the MEMSI field in the BARSETUP0 register is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BARSETUP0 register controls which bits in this field may be modified.Otherwise, the SIZE field in the BARSETUP1 register controls which bits in this field may be modified. See the SIZE field in the BARSETUP0 and BARSETUP1 registers for more information.

Notes

BAR2 - Base Address Register 2 (0x018)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP2 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. The value of this field is determined by the TYPE field in the BARSETUP2 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP2 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. The value of the SIZE field in the BARSETUP2 register controls which bits in this field may be modified. Bits that cannot be modified are always zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR which in this case is BAR3. See the PCI and PCI Express Base Specifications for more information.

Notes

BAR3 - Base Address Register 3 (0x01C)

When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. Otherwise, the BAR format below is used.

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP3 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. The value of this field is determined by the TYPE field in the BARSETUP3 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (reserved) reserved. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP3 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. See the PCI and PCI Express Base Specifications for more information. When the MEMSI field in the BARSETUP2 register is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BARSETUP2 register controls which bits in this field may be modified.Otherwise, the SIZE field in the BARSETUP3 register controls which bits in this field may be modified. See the SIZE field in the BARSETUP2 and BARSETUP3 registers for more information.

Notes

BAR4 - Base Address Register 4 (0x020)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP4 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. The value of this field is determined by the TYPE field in the BARSETUP4 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP4 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. The value of the SIZE field in the BARSETUP4 register controls which bits in this field may be modified. Bits that cannot be modified are always zero. When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR which in this case is BAR5. See the PCI and PCI Express Base Specifications for more information.

Notes

BAR5 - Base Address Register 5 (0x024)

When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. Otherwise, the BAR format below is used.

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. The value of this field is determined by the MEMSI field in the BARSETUP5 register. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. Since this is an odd-numbered BAR, it should only be configured with a 32-bit address format. The value of this field is determined by the TYPE field in the BARSETUP5 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (reserved) reserved. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. When the MEMSI field indicates I/O space, this field is always zero. The value of this field is determined by the PREF field in the BARSETUP5 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
31:4	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. See the PCI and PCI Express Base Specifications for more information. When the MEMSI field in the BARSETUP4 register is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BARSETUP4 register controls which bits in this field may be modified.Otherwise, the SIZE field in the BARSETUP5 register controls which bits in this field may be modified. See the SIZE field in the BARSETUP4 and BARSETUP5 registers for more information.

Notes

CCISPTR - CardBus CIS Pointer (0x028)

Bit Field	Field Name	Type	Default Value	Description
31:0	CCISPTR	RO	0x0	**CardBus CIS Pointer.** Not applicable.

SUBVID - Subsystem Vendor ID Pointer (0x02C)

Bit Field	Field Name	Type	Default Value	Description
15:0	SUBVID	RWL	0x0 SWSticky	**Subsystem Vendor ID.** This field identifies the vendor of the subsystem. This field must be loaded with the subsystem vendor ID prior to system software accessing PCI configuration space (e.g., via EEPROM). Refer to the PCI 3.0 specification, Section 6.2.4 for further information.

SUBID - Subsystem ID Pointer (0x02E)

Bit Field	Field Name	Type	Default Value	Description
15:0	SUBID	RWL	0x0 SWSticky	**Subsystem ID.** This field identifies the subsystem. This field must be loaded with the subsystem ID prior to system software accessing PCI configuration space (e.g., via EEPROM). Refer to the PCI 3.0 specification, Section 6.2.4 for further information.

EROMBASE - Expansion ROM Base (0x030)

Bit Field	Field Name	Type	Default Value	Description
31:0	EROMBASE	RO	0x0	**Expansion ROM Base Address.** The function does not implement an expansion ROM. Thus, this field is hardwired to zero.

CAPPTR - Capabilities Pointer (0x034)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPPTR	RWL	0x40 SWSticky	**Capabilities Pointer.** This field specifies a pointer to the head of the capabilities structure.

Notes

INTRLINE - Interrupt Line (0x03C)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRLINE	RW	0x0	**Interrupt Line.** This register communicates interrupt line routing information. Values in this register are programmed by system software and are system architecture specific. The function does not use the value in this register.

INTRPIN - Interrupt PIN (0x03D)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRPIN	RWL	0x0 SWSticky	**Interrupt Pin.** The value in this register indicates the INTx message (e.g., INTA, INTB, etc.) used by this function. This field has RWL type to allow system designers to change the INTx generated by this function, as shown below. 0x0 - (none) This function does not generate INTx interrupts. 0x1 - (INTA) This function generates INTA interrupts. 0x2 - (INTB) This function generates INTB interrupts. 0x3 - (INTC) This function generates INTC interrupts. 0x4 - (INTD) This function generates INTD interrupts. Programming this field to 0x0 in effect disables INTx interrupt generation.

MINGNT - Minimum Grant (0x03E)

Bit Field	Field Name	Type	Default Value	Description
7:0	MINGNT	RO	0x0	**Minimum Grant.** Not applicable.

MAXLAT - Maximum Latency (0x03F)

Bit Field	Field Name	Type	Default Value	Description
7:0	MAXLAT	RO	0x0	**Maximum Latency.** Not applicable.

Notes

PCI Express Capability Structure

PCIECAP - PCI Express Capability (0x040)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x10	**Capability ID.** The value of 0x10 identifies this capability as a PCI Express capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details.
19:16	VER	RWL	0x2 SWSticky	**PCI Express Capability Version.** This field indicates the PCI-SIG defined PCI Express capability structure version number.
23:20	TYPE	RO	0x0	**Port Type.** This field indicates that the function is a PCI Express Endpoint function.
24	SLOT	RO	0x0	**Slot Implemented.** Not applicable.
29:25	IMN	RO	0x0	**Interrupt Message Number.** The function is allocated only one MSI. Therefore, this field is set to zero.
31:30	Reserved	RO	0x0	Reserved field.

PCIEDCAP - PCI Express Device Capabilities (0x044)

Bit Field	Field Name	Type	Default Value	Description
2:0	MPAYLOAD	RWL	HWINIT (See description) MSWSticky	**Maximum Payload Size Supported.** This field indicates the maximum payload size that the device can support for TLPs. The default value of this field is automatically set by the hardware based on the port's maximum link width as determined by the stack's configuration. If a port has a maximum link width of x1, the default value of this field is 0x3. Otherwise, the default value of this field is 0x4. 0x0 - (s128) 128 bytes max payload size 0x1 - (s256) 256 bytes max payload size 0x2 - (s512) 512 bytes max payload size 0x3 - (s1024) 1024 bytes max payload size 0x4 - (s2048) 2048 bytes max payload size 0x5 - Not supported 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)

Notes

Bit Field	Field Name	Type	Default Value	Description
4:3	PFS	RO	0x0	**Phantom Functions Supported.** This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with the TLP's tag identifier. The value is hardwired to 0x0 to indicate that no function number bits are used for phantom functions.
5	ETAG	RWL	0x1 SWSticky	**Extended Tag Field Support.** This field indicates the maximum supported size of the Tag field as a requester. 0x0 - 5-bit Tag field supported 0x1 - 8-bit Tag field supported
8:6	E0AL	RWL	0x7 SWSticky	**Endpoint L0s Acceptable Latency.** This field indicates the acceptable total latency that this function can withstand due to transition from the L0s state to the L0 state. The value defaults to 0x7 indicating that this function places no limit on the L0s to L0 latency.
11:9	E1AL	RWL	0x7 SWSticky	**Endpoint L1 Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L1 state to the L0 state. The value defaults to 0x7 indicating that this function places no limit on the L1 to L0 latency.
12	ABP	RO	0x0	**Attention Button Present.** In PCI Express 1.0a when set, this bit indicates that an Attention Button is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
13	AIP	RO	0x0	**Attention Indicator Present.** In PCI Express 1.0a when set, this bit indicates that an Attention Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
14	PIP	RO	0x0	**Power Indicator Present.** In PCI Express 1.0a when set, this bit indicates that a Power Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
15	RBERR	RO	0x1	**Role Based Error Reporting.** This bit is set to indicate that this function supports role-based error reporting as defined in the PCI Express Base Specification Rev. 2.1.
17:16	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
25:18	CSPLV	RO	0x0	**Captured Slot Power Limit Value.** This field in combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. The value of this field is set by a Set_Slot_Power_Limit Message received by the port.[1]
27:26	CSPLS	RO	0x0	**Captured Slot Power Limit Scale.** This field specifies the scale used for the Slot Power Limit Value. The value of this field is set by a Set_Slot_Power_Limit Message received by the port. 0 - (v1) 1.0x 1 - (v1p1) 0.1x 2 - (v0p01) 0.01x 3 - (v0p001x) 0.001x
28	FLR	RO	0x0	**Function Level Reset Capability.** This function does not support function-level-reset. Therefore, this field is hardwired to 0x0.
31:29	Reserved	RO	0x0	Reserved field.

1. NOTE: Set_Slot_Power_Limit messages received by a port implicitly target all functions in the port.

PCIEDCTL - PCI Express Device Control (0x048)

Bit Field	Field Name	Type	Default Value	Description
0	CEREN	RW	0x0	**Correctable Error Reporting Enable.** This bit controls reporting of correctable errors by this function.
1	NFEREN	RW	0x0	**Non-Fatal Error Reporting Enable.** This bit controls reporting of non-fatal errors by this function.
2	FEREN	RW	0x0	**Fatal Error Reporting Enable.** This bit controls reporting of fatal errors by this function.
3	URREN	RW	0x0	**Unsupported Request Reporting Enable.** This bit controls reporting of unsupported requests by this function.
4	ERO	RW	0x1	**Enable Relaxed Ordering.** This bit may be set or cleared by software, but it has no effect on the hardware. See section TLP Attribute Processing on page 14-14 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
7:5	MPS	RW	0x0	**Max Payload Size.** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as large as the set value. As a transmitter, the function must not generate TLPs exceeding the set value. This field should be set to a value less than that advertised by the Maximum Payload Size Supported (MPAYLOAD) field in the PCI Express Device Capabilities (PCIEDCAP) register. Setting this field to a value larger than that advertised in the MPAYLOAD field produces undefined results. Programming of this field is subject to the restrictions outlined in section Maximum Payload Size on page 10-2 and section Maximum Payload Size on page 14-22. 0x0 - (s128) 128 bytes max payload size 0x1 - (s256) 256 bytes max payload size 0x2 - (s512) 512 bytes max payload size 0x3 - (s1024) 1024 bytes max payload size 0x4 - (s2048) 2048 bytes max payload size 0x5 - (s4096) 4096 bytes max payload size 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
8	ETFEN	RW	0x0	**Extended Tag Field Enable.** When this bit is set, Request TLPs generated by the function use an 8-bit tag (i.e., allows up to 256 outstanding requests). Else, Request TLPs generated by the function use a 5-bit tag (i.e., allows up to 32 outstanding requests). Note that the NT function does not modify the Tag of TLPs that cross the NT bridge. Also, the NT function only generates non-posted requests when issuing a punch-through request. In this case, only one request at a time is generated, with the Tag set to 0x0. As a result, this field has no functional effect on the NT function. Software must set this field appropriately based on knowledge of the tags issued by PCI Express agents that communicate across the NTB. For example, if PCI Express agents that communicate across the NTB use 8-bit tags, this field must be set accordingly in the NT function.
9	PFEN	RO	0x0	**Phantom Function Enable.** This function does not support phantom function numbers. Therefore, this field is hardwired to zero.
10	AUXPMEN	RO	0x0	**Auxiliary Power PM Enable.** The switch does not implement this capability.
11	ENS	RW	0x1	**Enable No Snoop.** This bit may be set or cleared by software, but it has no effect on the hardware. See section No Snoop Processing on page 14-14 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
14:12	MRRS	RW	0x2	**Maximum Read Request Size.** The NT function passes transactions through the NTB with the size unmodified. Therefore, this field has no functional effect on the behavior of the NTB. The user must ensure that no translated TLPs emitted by this NT function exceed the value programmed in this field (i.e., TLPs received by an NT function in another partition and emitted by this NT function). 0x0 - (s128) 128 bytes max read request size 0x1 - (s256) 256 bytes max read request size 0x2 - (s512) 512 bytes max read request size 0x3 - (s1024) 1024 bytes max read request size 0x4 - (s2048) 2048 bytes max read request size 0x5 - (s4096) 4096 bytes max read request size 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
15	IFLR	RO	0x0	**Initiate Function Level Reset.** This function does not support function-level-reset. Therefore this field is hardwired to 0x0.

PCIEDSTS - PCI Express Device Status (0x04A)

Bit Field	Field Name	Type	Default Value	Description
0	CED	RW1C	0x0	**Correctable Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
1	NFED	RW1C	0x0	**Non-Fatal Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
2	FED	RW1C	0x0	**Fatal Error Detected.** This bit indicates the status of Fatal errors detected by this function. Errors are logged in this registers regardless of whether error reporting is enabled or not.
3	URD	RW1C	0x0	**Unsupported Request Detected.** This bit indicates that the function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not.
4	AUXPD	RO	0x0	**Aux Power Detected.** Devices that require AUX power, set this bit when AUX power is detected.This device does not require AUX power, hence the value is hardwired to zero.
5	TP	RO	0x0	**Transactions Pending.** The NT function does not keep track of transactions it issues. Therefore, this field is hardwired to zero.
15:6	Reserved	RO	0x0	Reserved field.

Notes

PCIELCAP - PCI Express Link Capabilities (0x04C)

Bit Field	Field Name	Type	Default Value	Description
3:0	MAXLNKSPD	RWL	0x2 SWSticky	**Maximum Link Speed.** This field indicates the supported link speeds of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved Note: This device advertises support for 5 GT/s regardless of the setting of this field. Modifying this field has no effect on the hardware.
9:4	MAXLNK-WDTH	RWL	HWINIT (See description) MSWSticky	**Maximum Link Width.** This field indicates the maximum link width of the given PCI Express link. This field may be overridden to allow the link width to be forced to a smaller value. When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode). Violating this rule produces undefined results. Setting this field to an invalid or reserved value is allowed, and results in the port operating at it's default value. The default value of this field is automatically set by the hardware as described in section Port Maximum Link Width on page 7-2. 0 - reserved 1 - (x1) x1 link width 2 - (x2) x2 link width 4 - (x4) x4 link width 8 - (x8) x8 link width 12 - (x12) x12 link width 16 - (x16) x16 link width 32 - (x32) x32 link width others - reserved Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any stack configuration change.
11:10	ASPMS	RWL	0x3 SWSticky	**Active State Power Management (ASPM) Support.** This default value of this field is 0x3 to indicate that L0s and L1 are supported. This field may be overridden to allow user control over the ASPM capabilities of this port (L0s and/or L1). When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
14:12	L0SEL	RWL	0x6 SWSticky	**L0s Exit Latency.** This field indicates the L0s exit latency for the given PCI Express link. Transitioning from L0s to L0 always requires approximately 2.04us. Thus, default value indicates an L0s exit latency between 2us and 4us. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).

Notes

Bit Field	Field Name	Type	Default Value	Description
17:15	L1EL	RWL	0x2 SWSticky	**L1 Exit Latency.** This field indicates the L1 exit latency for the given PCI Express link. Transitioning from L1 to L0 always requires approximately 2.3 uS. Therefore, a value 2 µs to less than 4 µs is reported with a default value of 0x2. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
18	CPM	RWL	0x0 SWSticky	**Clock Power Management.** This bit indicates if the component tolerates removal of the reference clock via the "CLKREQ#" mechanism. The switch does not support the removal of reference clocks.
19	SDERR	RO	0x0	**Surprise Down Error Reporting.** Not applicable to upstream ports.
20	DLLLA	RO	0x0	**Data Link Layer Link Active Reporting.** Not applicable to upstream ports.
21	LBN	RO	0x0	**Link Bandwidth Notification Capability.** Not applicable to upstream ports.
23:22	Reserved	RO	0x0	Reserved field.
31:24	PORTNUM	RO	Port 0: 0x0 Port 2: 0x2 Port 4: 0x4 Port 6: 0x6 Port 8: 0x8 Port 12: 0xC Port 16: 0x10 Port 20: 0x14	**Port Number.** This field indicates the PCI Express port number for the corresponding link.

PCIELCTL - PCI Express Link Control (0x050)

Bit Field	Field Name	Type	Default Value	Description
1:0	ASPM	RW	0x0	**Active State Power Management (ASPM) Control.** This field controls the level of ASPM supported by the link. The initial value corresponds to disabled. 0x0 - (disabled) disabled 0x1 - (l0s) L0s enable entry 0x2 - (l1) L1 enable entry 0x3 - (l0sl1) L0s and L1 enable entry Note that "L0s enable entry" corresponds to the transmitter entering L0s (the receiver supports this function and is not affected by this setting). When a port operates in a multi-function mode, only capabilities enabled in all functions of the port are enabled for the port as a whole (e.g., L0s is enabled for the port when all functions of the port have L0s enabled in this field). It is recommended, though not required, that software program the same value in this field for all functions of the port.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	Reserved	RO	0x0	
3	RCB	RO	0x0	**Read Completion Boundary.** The NT function passes transactions through the NTB with the size unmodified. Therefore, this field has no functional effect on the behavior of the NTB.
4	LDIS	RO	0x0	**Link Disable.** Not applicable.
5	LRET	RWL	0x0	**Link Retrain.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. Writing a one to this field initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. This field always returns zero when read. It is permitted to set this bit while simultaneously modifying other fields in this register. When this bit is set and the LTSSM is already in the Recovery or Configuration states, all modifications that affect link retraining are applied in the subsequent retraining. Else, if the LTSSM is not in the Recovery or Configuration states, modifications that affect link retraining are applied immediately. For compliance with the PCI Express Base Specification, this bit has no effect on the upstream port when the REGUNLOCK bit is cleared in the SWCTL register. In this mode the field is hardwired to zero. When the REGUNLOCK bit is set, writing a one to the LRET bit initiates link retraining on the upstream port with a programmed delay (see section NT Function Registers on page 19-14 for details). The switch always returns a completion for the request that set this bit, before the effect of this bit is applied.
6	CCLK	RW	0x0	**Common Clock Configuration.** When set, this bit indicates that this port and the port at the opposite end of the link are operating with a distributed common reference clock. When a port operates in a multi-function mode, software must set this bit identically for all functions of the port. Otherwise, the port assumes that it is not operating with a distributed common reference clock. After modifying this bit in both components of the link, software must trigger a link retrain by setting the link retrain bit in the upstream component's Link Control register. In the switch, the L0s and L1 exit latencies do not change among common and non-common clock configurations.
7	ESYNC	RW	0x0	**Extended Sync.** When set this bit forces transmission of additional ordered sets when exiting the L0s state and when in the recovery state. When a port operates in a multi-function mode, the effect of this bit is applied when this bit is set in any of the port's functions.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	CLKP-WRMGT	RO	0x0	**Enable Clock Power Management.** The device does not support this feature.
9	HAWD	RO	0x0	**Hardware Autonomous Width Disable.** Device ports do not have a hardware autonomous mechanism to change link width, except due to link reliability issues. Therefore, this bit is not applicable and is hardwired to zero.
10	LBWINTEN	RO	0x0	**Link Bandwidth Management Interrupt Enable.** Not applicable.
11	LABWINTEN	RO	0x0	**Link Autonomous Bandwidth Interrupt Enable.** Not applicable.
15:12	Reserved	RO	0x0	Reserved field.

PCIELSTS - PCI Express Link Status (0x052)

Bit Field	Field Name	Type	Default Value	Description
3:0	CLS	RO	0x1	**Current Link Speed.** This field indicates the current link speed of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved
9:4	NLW	RO	HWINIT	**Negotiated Link Width.** This field indicates the negotiated width of the link. 00 0001b - x1 00 0010b - x2 00 0100b - x4 00 1000b - x8 00 1100b - x12 01 0000b - x16 10 0000b - x32 When the MAXLNKWDTH field in the PCIELCAP register selects a width not supported by the port, the value of this field corresponds to the setting of the MAXLNKWDTH field, regardless of the actual negotiated link width. When the MAXLNKWDTH field in the PCIELCAP register selects a width supported by the port, but the link is unable to train, the value in this field is set to 0x0. When the port operates in a multi-function mode, the above rules are based on the MAXLNKWDTH field for function 0 of the port. Note that software must ensure that all functions of the port have identical MAXLNKWDTH field values.
10	Reserved	RO	0x0	Reserved field.
11	LTRAIN	RO	0x0	**Link Training.** Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
12	SCLK	RWL	HWINIT SWSticky	**Slot Clock Configuration.** When set, this bit indicates that the port uses the same physical reference clock used by its link partner (i.e., common-clock configuration). The initial value of this field depends on the port's clocking mode. Refer to Table 2.4 for further details. When the port operates in a multi-function mode, this field reports the same value for all functions of the port.
13	DLLLA	RO	0x0	**Data Link Layer Link Active.** Not applicable.
14	LBWSTS	RO	0x0	**Link Bandwidth Management Status.** Not applicable.
15	LABWSTS	RO	0x0	**Link Autonomous Bandwidth Status.** Not applicable.

PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTRS	RWL	0xF	**Completion Timeout Ranges Supported.** The default value indicates support for all completion timeout ranges specified in the PCI Express Base Specification Rev 2.1.
4	CTDS	RWL	0x1	**Completion Timeout Disable Supported.** The default value indicates support for completion timeout disable.
5	ARIFS	RO	0x0	**ARI Forwarding Supported.** Not applicable.
6	ATOPRS	RO	0x0	**AtomicOp Routing Supported.** Not applicable.
7	ATOPC32S	RO	0x0	**32-bit AtomicOp Completer Supported.** Not supported.
8	ATOPC64S	RO	0x0	**64-bit AtomicOp Completer Supported.** Not supported.
9	CASC128S	RO	0x0	**128-bit CAS Completer Supported.** Not supported.
10	NROEP	RO	0x1	**No RO-enabled PR-PR Passing.** Not applicable.
11	LTRMS	RO	0x0	**LTR Mechanism Supported.** The switch does not support the Latency Tolerance Reporting mechanism.
13:12	TPHCS	RO	0x0	**TPH Completer Supported.** Not supported.
19:14	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
20	EFMTFS	RO	0x0	**Extended Fmt Field Supported.** The switch does not support the 3-bit definition of the FMT field in TLPs.
21	E2ETPS	RO	0x0	**End-to-End TLP Prefix Supported.** The switch does not support End-to-End TLP Prefixes.
31:22	Reserved	RO	0x0	Reserved field.

PCIEDCTL2 - PCI Express Device Control 2 (0x068)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTV	RW	0x0	**Completion Timeout Value.** The NT function does not track non-posted requests that it transmits (i.e., requests that crossed the NTB). As a result, the NT function does not implement a completion timeout mechanism. The value programmed in this field has no effect on the NT function. The system must not rely on the NT function detecting a completion timeout, regardless of the value programmed in this field. It is recommended that completion timeout be disabled in the NT function, by setting the Completion Timeout Disable (CTD) bit in this register.
4	CTD	RW	0x0	**Completion Timeout Disable.** The NT function does not track non-posted requests that it transmits (i.e., requests that crossed the NTB). As a result, the NT function does not implement a completion timeout mechanism. The value programmed in this field has no effect on the NT function. Since the NT function never detects completion timeout, it is recommended that software set this bit.
5	ARIFEN	RO	0x0	**ARI Forwarding Enable.** Not applicable.
6	ATOPRE	RO	0x0	**AtomicOp Requester Enable.** Not supported.
7	ATOPEB	RO	0x0	**AtomicOp Egress Blocking.** Not applicable.
8	IDORE	RO	0x0	**IDO Request Enable.** Not supported.
9	IDOCE	RO	0x0	**IDO Completion Enable.** Not supported.
10	LTRME	RO	0x0	**LTR Mechanism Enable.** Not supported.
14:11	Reserved	RO	0x0	Reserved field.
15	E2ETLPPB	RO	0x0	**End-to-End TLP Prefix Blocking.** Not supported.

Notes

PCIEDSTS2 - PCI Express Device Status 2 (0x06A)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

PCIELCTL2 - PCI Express Link Control 2 (0x070)

Bit Field	Field Name	Type	Default Value	Description
3:0	TLS	If NT function is function 0 of the port: RW Otherwise: RO	If NT function is function 0 of the port: 0x2 Sticky Otherwise: 0x0	**Target Link Speed.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable, this field is used to set the target compliance mode speed when software is using the ECOMP bit in this register to force a link into compliance mode. The switch supports 2.5 GT/s and 5.0 GT/s operation. Setting this field to an unsupported value produces undefined results. 0x1 - (gen1) 2.5 GT/s 0x2 - (gen2) 5.0 GT/s others - reserved
4	ECOMP	If NT function is function 0 of the port: RW Otherwise: RO	0x0 Sticky	**Enter Compliance.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable, software is permitted to force a link into compliance mode at the speed indicated by the TLS field by setting this bit in both components on a link and then initiating a hot reset on the link.
5	HASD	RO	0x0	**Hardware Autonomous Speed Disable.** Switch ports do not have an autonomous mechanism to regulate link speed, except due to link reliability issues. Therefore, this bit is not applicable. Note that this bit does not affect link speed changes triggered by software setting the target link speed and link-retrain bits. Refer to section Software Management of Link Speed on page 7-8 for further details.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	SDE	RWL	0x0 SWSticky	**Selectable De-emphasis.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable, this bit selects the de-emphasis preference advertised via training sets (the actual de-emphasis on the link is selected by the link partner). 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB This bit has no effect when the link operates at 2.5 GT/s, or when the link operates in low-swing mode. After modifying this field, it is recommended that the link be fully retrained by setting the FLRET bit in the PHYLSTATE0 register.
9:7	TM	If NT function is function 0 of the port: RW Otherwise: RO	0x0 Sticky	**Transmit Margin.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable: This field controls the value of the non de-emphasized voltage level at the transmitter pins. This field is reset to 0x0 on entry to the LTSSM Polling.Configuration substate. 0x0 - Normal operating range 0x1 - 900 mV for full swing and 500 mV for low-swing 0x2 - 700 mV for full swing and 400 mV for low-swing 0x3 - 500 mV for full swing and 300 mV for low-swing 0x4 - 300 mV for full swing and 200 mv for low-swing 0x5 - 200 mV for full swing and 100 mv for low-swing 0x6 - 0x7 - Reserved This register is intended for debug and compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value. When this field is set to “Normal Operating Range”, the SerDes transmitter drive level is selected via the SerDes Transmitter Control registers (S[x]TXLCTL0 and S[x]TXLCTL1). Refer to section SerDes Transmitter Controls on page 8-3. When this field is modified, the newly selected value is not applied until the PHY LTSSM transitions through the states in which it is allowed to modify the transmit margin setting on the line (i.e., Recovery.RcvrLock). Therefore, after modifying this field, it is recommended that the link be retrained by setting the LRET bit in the PCIELCTL register.

Notes

Bit Field	Field Name	Type	Default Value	Description
10	EMC	If NT function is function 0 of the port: RW Otherwise: RO	0x0 Sticky	**Enter Modified Compliance.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable: When this bit is set to 1b, the port transmits the modified compliance pattern if the LTSSM enters Polling.Compliance state. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.
11	CSOS	If NT function is function 0 of the port: RW Otherwise: RO	0x0 Sticky	**Compliance SOS.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable: When set to 1b, the LTSSM is required to send SOS periodically in between the compliance and modified compliance patterns.
12	CDE	If NT function is function 0 of the port: RW Otherwise: RO	0x0 Sticky	**Compliance De-emphasis.** This field is only applicable for port operating modes in which the NT function is function 0 of the port. When applicable: This bit selects the de-emphasis value in the Polling.Compliance state when this state was entered as a result of setting the Enter Compliance (ECOMP) bit in this register. 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB This bit is intended for debug, compliance testing purposes. System firmware and software is allowed to modify this bit only during debug or compliance testing.
15:13	Reserved	RO	0x0	Reserved field.

PCIELSTS2 - PCI Express Link Status 2 (0x072)

Bit Field	Field Name	Type	Default Value	Description
0	CDE	RO	0x0	**Current De-emphasis.** The value of this bit indicates the current de-emphasis level when the link operates in 5.0 GT/s. 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB The value of this bit in undefined when the link operates at 2.5 GT/s.
15:1	Reserved	RO	0x0	Reserved field.

Notes

PCI Power Management Capability Structure

PMCAP - PCI Power Management Capabilities (0x0C0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 identifies this capability as a PCI power management capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details.
18:16	VER	RO	0x3	**Power Management Capability Version.** Complies with version the PCI Bus Power Management Interface Specification, Revision 1.2.
19	PMECLK	RO	0x0	**PME Clock.** Does not apply to PCI Express.
20	Reserved	RO	0x0	Reserved field.
21	DEVSP	RWL	0x0 SWSticky	**Device Specific Initialization.** The value of zero indicates that no device specific initialization is required.
24:22	AUXI	RO	0x0	**AUX Current.** The device does not use auxiliary current.
25	D1	RO	0x0	**D1 Support.** This field indicates that this function does not support D1.
26	D2	RO	0x0	**D2 Support.** This field indicates that this function does not support D2.
31:27	PME	RO	0x0	**PME Support.** This field indicates the power states in which the function may generate a PME.

PMCSR - PCI Power Management Control and Status (0x0C4)

Bit Field	Field Name	Type	Default Value	Description
1:0	PSTATE	RW	0x0	**Power State.** This field is used to determine the current power state of the function and to set a new power state. 0x0 - (d0) D0 state 0x1 - (d1) D1 state (not supported by the switch and reserved) 0x2 - (d2) D2 state (not supported by the switch and reserved) 0x3 - (d3) $D3_{hot}$ state
2	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	NOSOFTRST	RWL	0x1 SWSticky	**No Soft Reset.** This bit indicates if the configuration context is preserved by the function when the device transitions from a D3hot to D0 power management state. 0x0 - (reset) State reset 0x1 - (preserved) State preserved
7:4	Reserved	RO	0x0	Reserved field.
8	PMEE	RO	0x0	**PME Enable.** Not applicable since this function does not support generation of PME events.
12:9	DSEL	RO	0x0	**Data Select.** The optional data register is not implemented.
14:13	DSCALE	RO	0x0	**Data Scale.** The optional data register is not implemented.
15	PMES	RW1C	0x0 Sticky	**PME Status.** Since this function never generates a PME, this bit will never be set.
21:16	Reserved	RO	0x0	Reserved field.
22	B2B3	RO	0x0	**B2/B3 Support.** Does not apply to PCI Express.
23	BPCCE	RO	0x0	**Bus Power/Clock Control Enable.** Does not apply to PCI Express.
31:24	DATA	RO	0x0	**Data.** This optional field is not implemented.

Message Signaled Interrupt Capability Structure

MSICAP - Message Signaled Interrupt Capability and Control (0x0D0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x5	**Capability ID.** The value of 0x5 identifies this capability as a MSI capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. This field is set to 0x0 indicating that it is the last capability. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
16	EN	RW	0x0	**Enable.** This bit enables MSI. 0x0 - (disable) disabled 0x1 - (enable) enabled
19:17	MMC	RO	0x0	**Multiple Message Capable.** This field contains the number of requested messages.
22:20	MME	RW	0x0	**Multiple Message Enable.** Hardwired to one message.
23	A64	RO	0x1	**64-bit Address Capable.** The function is capable of generating messages using a 64-bit address.
31:24	Reserved	RO	0x0	Reserved field.

MSIADDR - Message Signaled Interrupt Address (0x0D4)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	ADDR	RW	0x0	**Message Address.** This field specifies the lower portion of the DWORD address of the MSI memory write transaction. Refer to section Interrupts on page 14-20 for restrictions on the programming of this field.

MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8)

Bit Field	Field Name	Type	Default Value	Description
31:0	UADDR	RW	0x0	**Upper Message Address.** This field specifies the upper portion of the DWORD address of the MSI memory write transaction. If the contents of this field are non-zero, then 64-bit address is used in the MSI memory write transaction. If the contents of this field are zero, then the 32-bit address specified in the MSIADDR register is used. Refer to section Interrupts on page 14-20 for restrictions on the programming of this field.

MSIMDATA - Message Signaled Interrupt Message Data (0x0DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	MDATA	RW	0x0	**Message Data.** This field contains the lower 16-bits of data that are written when a MSI is signaled.
31:16	Reserved	RO	0x0	Reserved field.

Notes

Subsystem ID and Subsystem Vendor ID

SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0xD	**Capability ID.** The value of 0xD identifies this capability as a SSID/SSVID capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details.
31:16	Reserved	RO	0x0	Reserved field.

SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4)

Bit Field	Field Name	Type	Default Value	Description
15:0	SSVID	RWL	0x0 SWSticky	**Subsystem Vendor ID.** This field identifies the manufacturer of the add-in card or subsystem. SSVID values are assigned by the PCI-SIG to insure uniqueness.
31:16	SSID	RWL	0x0 SWSticky	**Subsystem ID.** This field identifies the add-in card or subsystem. SSID values are assigned by the vendor.

Extended Configuration Space Access Registers

ECFGADDR - Extended Configuration Space Access Address (0x0F8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
7:2	REG	RW	0x0	**Register Number.** This field selects the configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 2.1. The following restrictions apply when programming this register: 1) The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). 2) The value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). 3) The value in this register must not be programmed to point to the NT Mapping Table Address and Data registers (NTMTBLADDR and NTMTBLDATA) in this function. Violation of these rules produces undefined results.
11:8	EREG	RW	0x0	**Extended Register Number.** This field selects the extended configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 2.1. The following restrictions apply when programming this register: 1) The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). 2) The value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). 3) The value in this register must not be programmed to point to the NT Mapping Table Address and Data registers (NTMTBLADDR and NTMTBLDATA) in this function. Violation of these rules produces undefined results.
31:12	Reserved	RO	0x0	Reserved field.

ECFGDATA - Extended Configuration Space Access Data (0x0FC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Configuration Data.** A read from this field will return the configuration space register value pointed to by the ECFGADDR register. A write to this field will update the contents of the configuration space register pointed to by the ECFGADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

Notes

Advanced Error Reporting (AER) Extended Capability

AERCAP - AER Capabilities (0x100)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 indicates an Advanced Error Reporting capability structure.
19:16	CAPVER	RO	0x2	**Capability Version.** The value of 0x2 indicates compatibility with the PCI Express Base Specification Rev 2.1.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details.

AERUES - AER Uncorrectable Error Status (0x104)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW1C	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW1C	0x0 Sticky	**Data Link Protocol Error Status.** This bit is set when a data link layer protocol error is detected.
5	SDOENERR	RO	0x0	**Surprise Down Error Status.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW1C	0x0 Sticky	**Poisoned TLP Status.** This bit is set when a poisoned TLP is detected.
13	FCPERR	RO	0x0	**Flow Control Protocol Error Status.** Not applicable. The switch does not support flow control protocol error checking.
14	COMPTO	RW1C	0x0 Sticky	**Completion Timeout Status.** The NT function does not track completion timeout for non-posted requests that is transmits (i.e., requests that crossed the NTB). As a result, this bit is never set.
15	CABORT	RW1C	0x0 Sticky	**Completer Abort Status.** This bit is never set as this function never responds to a non-posted request with a completer abort. Note that this bit is not set when the NT function emits a completion with completer abort status that crossed the NTB.
16	UECOMP	RW1C	0x0 Sticky	**Unexpected Completion Status.** This bit is set when an unexpected completion is detected.

Notes

Bit Field	Field Name	Type	Default Value	Description
17	RCVOVR	RW1C	0x0 Sticky	**Receiver Overflow Status.** This bit is set when a receiver overflow is detected.
18	MALFORMED	RW1C	0x0 Sticky	**Malformed TLP Status**. This bit is set when a malformed TLP is detected.
19	ECRC	RW1C	0x0 Sticky	**ECRC Status.** This bit is set when an ECRC error is detected.
20	UR	RW1C	0x0 Sticky	**UR Status.** This bit is set when an unsupported request is detected.
21	ACSV	RW1C	0x0 Sticky	**ACS Violation Status.** This bit is set when an ACS violation is detected by this function.
22	UIE	RW1C	0x0 Sticky	**Uncorrectable Internal Error Status.** This bit is set when an uncorrectable internal error associated with the this function is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RW1C	0x0 Sticky	**MC Blocked TLP Status.** Not applicable (i.e., the NT function does not check the MC_Block_All register when emitting a translated TLP, per the usage restriction rules in section NT Function Registers on page 19-14).
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Status.** Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Status.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERUEM - AER Uncorrectable Error Mask (0x108)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	DLPERR	RW	0x0 Sticky	**Data Link Protocol Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
5	SDOENERR	RO	0x0	**Surprise Down Error Mask.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
13	FCPERR	RO	0x0	**Flow Control Protocol Error Mask.** Not applicable.
14	COMPTO	RW	0x0 Sticky	**Completion Timeout Mask.** This function does not track non-posted requests it transmits (i.e., requests that crossed the NTB). Therefore, this bit has no effect when set.
15	CABORT	RW	0x0 Sticky	**Completer Abort Mask.** This field has no functional effect, as the Completer Abort bit in the AERUES register is never set.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
17	RCVOVR	RW	0x0 Sticky	**Receiver Overflow Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
18	MALFORMED	RW	0x0 Sticky	**Malformed TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
19	ECRC	RW	0x0 Sticky	**ECRC Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
20	UR	RW	0x0 Sticky	**UR Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
21	ACSV	RW	0x0 Sticky	**ACS Violation Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
22	UIE	RW	0x0 Sticky	**Uncorrectable Internal Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RW	0x0 Sticky	**MC Blocked TLP Mask.** Not applicable.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Mask.** Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Mask.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERUESV - AER Uncorrectable Error Severity (0x10C)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x1 Sticky	**Data Link Protocol Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
5	SDOENERR	RO	0x1	**Surprise Down Error Severity.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Status Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
13	FCPERR	RO	0x1	**Flow Control Protocol Error Severity.** Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	COMPTO	RW	0x0 Sticky	**Completion Timeout Severity.** This function does not track non-posted requests it transmits (i.e., requests that crossed the NTB). Therefore, this bit has no effect when set.
15	CABORT	RW	0x0 Sticky	**Completer Abort Severity.** This field has no functional effect, as the Completer Abort bit in the AERUES register is never set.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
17	RCVOVR	RW	0x1 Sticky	**Receiver Overflow Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
18	MALFORMED	RW	0x1 Sticky	**Malformed TLP Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
19	ECRC	RW	0x0 Sticky	**ECRC Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
20	UR	RW	0x0 Sticky	**UR Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
21	ACSV	RW	0x0 Sticky	**ACS Violation Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
22	UIE	RW	0x0 Sticky	**Uncorrectable Internal Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of one. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RW	0x0 Sticky	**MC Blocked TLP Severity.** Not Applicable.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Severity.** Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Severity.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERCES - AER Correctable Error Status (0x110)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW1C	0x0 Sticky	**Receiver Error Status.** This bit is set when the Physical Layer detects a receiver error.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW1C	0x0 Sticky	**Bad TLP Status.** This bit is set when a bad TLP is detected.
7	BADDLLP	RW1C	0x0 Sticky	**Bad DLLP Status.** This bit is set when a bad DLLP is detected.
8	RPLYROVR	RW1C	0x0 Sticky	Replay Number Rollover Status. This bit is set when a replay number rollover has occurred indicating that the data link layer has abandoned replays and has requested that the link be retrained.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW1C	0x0 Sticky	**Replay Timer timeout Status.** This bit is set when the replay timer in the data link layer times out.
13	ADVISORYNF	RW1C	0x0 Sticky	**Advisory Non-Fatal Error Status.** This bit is set when an advisory non-fatal error is detected as described in Section 6.2.3.2.4 of the PCI Express Base Specification 2.1.
14	CIE	RW1C	0x0 Sticky	**Correctable Internal Error Status.** This bit is set whenever an correctable internal error associated with the port is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW1C	0x0 Sticky	**Header Log Overflow Status.** This bit is set when an error that requires packet-header logging occurs but the packet header cannot be logged by the function's AER Header Log registers (AERHL[1:4]DW). A packet's header cannot be logged in the AER Header Log registers when an error occurs while the First Error Pointer (FEPTR field in the AERCTL register) is valid. The First Error Pointer is valid when it points to a set bit in the AERUES register (i.e., indicating the occurrence of a prior uncorrectable error which has not been cleared by software). When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
31:16	Reserved	RO	0x0	Reserved field.

AERCEM - AER Correctable Error Mask (0x114)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 Sticky	**Receiver Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 Sticky	**Bad TLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
7	BADDLLP	RW	0x0 Sticky	**Bad DLLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	RPLYROVR	RW	0x0 Sticky	**Replay Number Rollover Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 Sticky	**Replay Timer timeout Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
13	ADVISORYNF	RW	0x1 Sticky	**Advisory Non-Fatal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
14	CIE	RW	0x1 Sticky	**Correctable Internal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW	0x1 Sticky	**Header Log Overflow Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
31:16	Reserved	RO	0x0	Reserved field.

AERCTL - AER Control (0x118)

Bit Field	Field Name	Type	Default Value	Description
4:0	FEPTR	RO	0x0 Sticky	**First Error Pointer.** This field contains a pointer to the bit in the AERUES register that resulted in the first reported error. This field is valid only when it points to a set bit in the AERUES register.
5	ECRCGC	RWL	0x1 SWSticky	**ECRC Generation Capable.** This bit indicates if the function is capable of generating ECRC.
6	ECRCGE	RW	0x0 Sticky	**ECRC Generation Enable.** When this bit is set, ECRC generation is enabled for the function.
7	ECRCCC	RWL	0x1 SWSticky	**ECRC Check Capable.** This bit indicates if the function is capable of checking ECRC.
8	ECRCCE	RW	0x0 Sticky	**ECRC Check Enable.** When this bit is set, ECRC checking is enabled for the function.
9	MHRC	RO	0x0	**Multiple Header Recording Capable.** The switch ports do not support recording of multiple packet headers.
10	MHRE	RO	0x0	**Multiple Header Recording Enable.** The switch ports do not support recording of multiple packet headers. As a result, this bit is hardwired to 0x0.
31:11	Reserved	RO	0x0	Reserved field.

Notes

AERHL1DW - AER Header Log 1st Doubleword (0x11C)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 1st doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL2DW - AER Header Log 2nd Doubleword (0x120)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 2nd doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL3DW - AER Header Log 3rd Doubleword (0x124)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 3rd doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL4DW - AER Header Log 4th Doubleword (0x128)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 4th doubleword of the TLP header that resulted in the first reported uncorrectable error.

Device Serial Number Extended Capability

SNUMCAP - Serial Number Capabilities (0x180)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x3	**Capability ID.** The value of 0x3 indicates a device serial number capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification, Rev 2.1.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

SNUMLDW - Serial Number Lower Doubleword (0x184)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 SWSticky	**Lower 32-bits of Device Serial Number.** This field contains the lower 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device. When a port operates in a multi-function mode, this field must be programmed identically in all functions that implement this capability.

SNUMUDW - Serial Number Upper Doubleword (0x188)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 SWSticky	**Upper 32-bits of Device Serial Number.** This field contains the upper 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device. When a port operates in a multi-function mode, this field must be programmed identically in all functions that implement this capability.

PCI Express Virtual Channel Capability

The Virtual Channel (VC) capability structure defined in this section is only applicable for port operating modes in which the NT function is function 0 of the port. For port operating modes in which the PCI-to-PCI bridge function is function 0 of the port, this capability structure must not be linked into the extended capabilities list in the NT function and the registers in this capability structure are considered 'reserved'[1] (i.e., must not be programmed). In this case, the NT function uses the PCI-to-PCI bridge function's VC Capability Structure for architected TC/VC mapping. Refer to section Arbitration on page 4-6 for details.

1. Reading from a reserved address returns and undefined value. Writes to a reserved address complete successfully but produce undefined behavior on the register.

Notes

PCIEVCECAP - PCI Express VC Extended Capability Header (0x200)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x2	**Capability ID.** The value of 0x2 indicates a Virtual Channel Capability Structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base specification, Rev 2.1.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

PVCCAP1- Port VC Capability 1 (0x204)

Bit Field	Field Name	Type	Default Value	Description
2:0	EVCCNT	RO	0x0	**Extended VC Count.** A value 0x0 indicates that only the default VC (VC0) is implemented.
3	Reserved	RO	0x0	Reserved field.
6:4	LPEVCCNT	RO	0x0	**Low Priority Extended VC Count.** Not applicable (only the default VC0 is implemented).
7	Reserved	RO	0x0	Reserved field.
9:8	REFCLK	RO	0x0	**Reference Clock.** Not supported (i.e., Time-based WRR Port Arbitration is not implemented).
11:10	PATBLSIZ	RWL	0x0 SWSticky	**Port Arbitration Table Entry Size.** This field is not applicable to endpoint functions (e.g., the NT function), but may be overridden by software (e.g., EEPROM) to program non-transparent-bridge inter-partition transfer arbitration using Weighted Round Robin (WRR). Refer to section Arbitration on page 4-6 for further details on inter-partition port arbitration. This field indicates the size of the port arbitration table associated with the function. ***If modified, it must be set to 0x3 to indicate a table with 8-bit entries.*** 0x0 - (bit1) Port arbitration table is 1-bit 0x1 - (bit2) Port arbitration table is 2-bits 0x2 - (bit4) Port arbitration table is 4-bits 0x3 - (bit8) Port arbitration table is 8-bits
31:12	Reserved	RO	0x0	Reserved field.

Notes

PVCCAP2- Port VC Capability 2 (0x208)

Bit Field	Field Name	Type	Default Value	Description
7:0	VCARBCAP	RO	0x0	**VC Arbitration Capability.** Not applicable (only the default VC0 is implemented).
23:8	Reserved	RO	0x0	Reserved field.
31:24	VCATBLOFF	RO	0x0	**VC Arbitration Table Offset.** Not applicable.

PVCCTL - Port VC Control (0x20C)

Bit Field	Field Name	Type	Default Value	Description
0	LVCAT	RO	0x0	**Load VC Arbitration Table.** Not applicable.
3:1	VCARBSEL	RW	0x0	**VC Arbitration Select.** Not applicable (only the default VC0 is implemented). This field has RW type for compliance with the PCI Express Base Specification.
15:4	Reserved	RO	0x0	Reserved field.

PVCSTS - Port VC Status (0x20E)

Bit Field	Field Name	Type	Default Value	Description
0	VCATS	RO	0x0	**VC Arbitration Table Status.** Not applicable.
15:1	Reserved	RO	0x0	Reserved field.

VCR0CAP- VC Resource 0 Capability (0x210)

Bit Field	Field Name	Type	Default Value	Description
7:0	PARBC	R0	0x0	**Port Arbitration Capability.** Not applicable.
14:8	Reserved	RO	0x0	Reserved field.
15	RJST	RO	0x0	**Reject Snoop Transactions.** No supported for switch ports.
22:16	MAXTS	RO	0x0	**Maximum Time Slots.** Since this VC does not support time-based WRR, this field is not valid.
23	Reserved	RO	0x0	Reserved field.
31:24	PATBLOFF	R0	0x0	**Port Arbitration Table Offset.** Not applicable.

Notes

VCR0CTL- VC Resource 0 Control (0x214)

Bit Field	Field Name	Type	Default Value	Description
7:0	TCVCMAP	bit 0: RO bits 1 through 7: RW	0xFF	**TC/VC Map.** This field indicates the TCs that are mapped to the VC resource. Each bit corresponds to a TC. When a bit is set, the corresponding TC is mapped to the VC.
15:8	Reserved	RO	0x0	Reserved field.
16	LPAT	RO	0x0	**Load Port Arbitration Table.** Not applicable.
19:17	PARBSEL	RO	0x0 SWSticky	**Port Arbitration Select.** Not applicable.
23:20	Reserved	RO	0x0	Reserved field.
26:24	VCID	RO	0x0	**VC ID.** This field assigns a VC ID to the VC resource. For VC0, this field is always hardwired to zero.
30:27	Reserved	RO	0x0	Reserved field.
31	VCEN	RO	0x1	**VC Enable.** This field, when set, enables a virtual channel. For VC0, this field is hardwired to 0x1 (enabled).

VCR0STS - VC Resource 0 Status (0x218)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.
16	PATS	RO	0x0	**Port Arbitration Table Status.** Not applicable.
17	VCNEG	RO	0x0	**VC Negotiation Pending.** This bit is not applicable for VC0 and is therefore hardwired to 0x0.
31:18	Reserved	RO	0x0	Reserved field.

Notes

ACS Extended Capability

ACSECAPH - ACS Extended Capability Header (0x320)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0xD	**Capability ID.** The value of 0xD indicates an ACS extended capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

ACSCAP - ACS Capability (0x324)

Bit Field	Field Name	Type	Default Value	Description
0	V	RO	0x0	**ACS Source Validation.** Not applicable to multi-function upstream ports.
1	B	RO	0x0	**ACS Translation Blocking.** Not applicable to multi-function upstream ports.
2	R	RWL	0x0 SWSticky	**ACS P2P Request Redirect.** If set, indicates that this function implements ACS Peer-to-Peer Request Redirect. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. This field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function). Refer to section Access Control Services (ACS) on page 14-23 for details.
3	C	RWL	0x0 SWSticky	**ACS P2P Completion Redirect.** If set, indicates that this function implements ACS Peer-to-Peer Completion Redirect. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. This field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function). Refer to section Access Control Services (ACS) on page 14-23 for details.
4	U	RO	0x0	**ACS Upstream Forwarding.** Not applicable to multi-function upstream ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	E	RO	0x0	**ACS P2P Egress Control.** The switch does not support ACS P2P Egress Control among functions in a multi-function upstream port.
6	T	RWL	0x0 SWSticky	**ACS Direct Translated P2P.** If set, indicates that this function implements ACS Direct Translated Peer-to-Peer. For a multi-function upstream port, peer-to-peer refers to transfers among functions in the port. This field can only be set to 0x1 when the port's operating mode is a multi-function mode (i.e., upstream switch port with NT function and upstream switch port with NT and DMA function). Refer to section Access Control Services (ACS) on page 14-23 for details.
15:7	Reserved	RO	0x0	Reserved field.

ACSCTL - ACS Control (0x326)

Bit Field	Field Name	Type	Default Value	Description
0	V	RO	0x0	**ACS Source Validation Enable.** Not applicable to multi-function upstream ports.
1	B	RO	0x0	**ACS Translation Blocking Enable.** Not applicable to multi-function upstream ports.
2	R	RW	0x0	**ACS P2P Request Redirect Enable.** When set, this function performs ACS Peer-to-Peer Request Redirect for function-to-function transfers. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
3	C	RW	0x0	**ACS P2P Completion Redirect Enable.** When set, this function performs ACS Peer-to-Peer Completion Redirect for function-to-function transfers. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
4	U	RO	0x0	**ACS Upstream Forwarding Enable.** Not applicable to multi-function upstream ports.
5	E	RO	0x0	**ACS P2P Egress Control Enable.** The switch does not support ACS P2P Egress Control among functions in a multi-function upstream port.
6	T	RW	0x0	**ACS Direct Translated P2P Enable.** When set, this function performs ACS Direct Translated Peer-to-Peer control for function-to-function transfers. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
15:7	Reserved	RO	0x0	Reserved field.

Notes

Multicast Extended Capability

MCCAPH - Multicast Extended Capability Header (0x330)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x12	**Capability ID.** The value of 0x12 indicates a multicast capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section NT Function Capability Structures on page 19-22 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

MCCAP - Multicast Capability (0x334)

Bit Field	Field Name	Type	Default Value	Description
5:0	MAXGROUP	RWL	0x3 SWSticky	**Max Multicast Groups.** This field indicates the maximum number of multicast groups supported by the NT function. The number of supported groups is equal to the value in this field plus one.
14:6	Reserved	RO	0x0	Reserved field.
15	ECRCREG	RO	0x0	**ECRC Regeneration Supported.** Not applicable to endpoint functions.

MCCTL- Multicast Control (0x336)

Bit Field	Field Name	Type	Default Value	Description
5:0	NUMGROUP	RW	0x0	**Number of Multicast Groups.** When the Multicast Enabler (MEN) bit is set, this field indicates the number of multicast groups that are enabled. The number of groups enabled is equal to the value in this field plus one. The behavior is undefined when the value in this field exceeds the value of the MAXGROUP field in the MCCAP register. This field must be set identically in all port functions in the partition associated with this port.
14:6	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	MEN	RW	0x0	**Multicast Enable.** When this bit is set, multicast is enabled in the switch partition associated with this function. This field must be set identically in all port functions in the partition associated with this port.

MCBARL- Multicast Base Address Low (0x338)

Bit Field	Field Name	Type	Default Value	Description
5:0	INDEXPOS	RW	0x0	**Index Position.** When multicast is enabled, this field specifies the least significant bit of the multicast group number within a TLP address. The behavior is undefined when multicast is enabled and this field is less than 12. This field must be set identically in all port functions in the partition associated with this port.
11:6	Reserved	RO	0x0	Reserved field.
31:12	MCBARL	RW	0x0	**Multicast BAR Low.** This field specifies the lower 20-bits (i.e., bits 12 through 31) of the multicast BAR. The behavior is undefined if bits in this field corresponding to address bits that contain the multicast group number or those less than the multicast index position (i.e., INDEXPOS) are non-zero. This field must be set identically in all port functions in the partition associated with this port.

MCBARH- Multicast Base Address High (0x33C)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBARH	RW	0x0	**Multicast BAR High.** This field specifies the upper 32-bits (i.e., bits 32 through 63) of the multicast BAR. The behavior is undefined if bits in this field corresponding to address bits that contain the multicast group number or those less than the multicast index position (i.e., INDEXPOS) are non-zero. This field must be set identically in all port functions in the partition associated with this port.

Notes

MCRCVL- Multicast Receive Low (0x340)

Bit Field	Field Name	Type	Default Value	Description
3:0	MCRCV	RW	0x0	**Multicast Receive.** Each bit in this field corresponds to one of the lower 32 multicast groups (e.g., bit 0 corresponds to multicast group 0, bit 1 corresponds to multicast group 1, and so on). When a bit is set in this field for an enabled multicast group, multicast TLPs associated with that multicast group are accepted and processed by this function. Otherwise, the TLP is ignored by this function. The value of bits greater than NUMGROUP in the MCCTL register is ignored. Note that the NT function supports a maximum of 4 groups.
31:4	Reserved	RO	0x0	Reserved field.

MCRCVH- Multicast Receive High (0x344)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCRCV	RO	0x0	**Multicast Receive.** Not applicable as the NT function supports a maximum of 4 groups.

MCBLKALLL- Multicast Block All Low (0x348)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKALL	RW	0x0	**Multicast Block All.** Not applicable (i.e., the NT function does not check the MC_Block_All register when emitting a translated TLP, per the usage restriction rules in section Usage Restrictions on page 17-6). This register has no functional effect, but remains read-writable for compatibility with the PCI Express Base Specification.

MCBLKALLH- Multicast Block All High (0x34C)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKALL	RW	0x0	**Multicast Block All.** Not applicable (i.e., the NT function does not check the MC_Block_All register when emitting a translated TLP, per the usage restriction rules in section Usage Restrictions on page 17-6). This register has no functional effect, but remains read-writable for compatibility with the PCI Express Base Specification.

Notes

MCBLKUTL- Multicast Block Untranslated Low (0x350)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKUT	RO	0x0	**Multicast Block Untranslated.** Not applicable (the NT function does not implement Address Translation Services (ATS)).

MCBLKUTH - Multicast Block Untranslated High (0x354)

Bit Field	Field Name	Type	Default Value	Description
31:0	MCBLKUT	RO	0x0	**Multicast Block Untranslated.** Not applicable (the NT function does not implement Address Translation Services (ATS)).

NT Registers

NT Control & Status

NTCTL - NT Endpoint Control (0x400)

Bit Field	Field Name	Type	Default Value	Description
0	IDPROTDIS	RW	0x0 SWSticky	**ID Protection Check Disable.** When this bit is set, ID protection checking performed by the NT endpoint for posted requests (i.e., the NT lookup) is disabled and all posted requests regardless of requester ID are allowed to map through the NT endpoint. 0x0 - (enable) ID protection check enable 0x1 - (disable) ID protection check disable
1	CPEN	RW	0x0	**Completion Enable.** When this bit is cleared, the NT endpoint does not emit completion TLPs that have crossed the NTB. When this bit is set, the NT endpoint does emit completion TLPs that have crossed the NTB. This bit has no effect on completions generated by the NT endpoint itself (e.g., in response to configuration requests). 0x0 - (disable) Disable emission of translated completions 0x1 - (Enable) Enable emission of translated completions Note that this bit must be set in the NT function of a source partition prior to sending non-posted requests across the NTB, to allow the completions generated in the destination partition to be emitted back into the source partition.
2	RNS	RW	0x0 SWSticky	**Request No Snoop Processing.** When the IDPROTDIS bit in this register is set, this bit controls No Snoop processing on posted request TLPs received by the NT endpoint. No Snoop processing is described in section No Snoop Processing on page 14-14. When the IDPROTDIS bit in this register is cleared, this bit has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	ATP	RW	0x0 SWSticky	**Address Type Processing.** When the IDPROTDIS bit in this register is set, this bit controls Address Type processing on posted request TLPs received by the NT endpoint. Address Type processing is described in section Address Type Processing on page 14-15. When the IDPROTDIS bit in this register is cleared, this bit has no effect.
31:4	Reserved	RO	0x0	Reserved field.

NT Interrupt and Signaling

NTINTSTS - NT Endpoint Interrupt Status (0x404)

Bit Field	Field Name	Type	Default Value	Description
0	MSG	RO	0x0	**Message Interrupt.** This bit is set whenever an unmasked bit is set in the Message Status (MSGSTS) register.
1	DBELL	RO	0x0	**Doorbell Interrupt.** This bit is set whenever an unmasked bit is set in the Inbound Doorbell Status (INDBELLSTS) register.
2	Reserved	RO	0x0	Reserved field.
3	SEVENT	RW1C	0x0	**Switch Event.** This bit is set whenever an unmasked switch event is generated to the partition with which the NT endpoint is associated (i.e., when the corresponding event bit in the SESTS register transitions from 0x0 to 0x1). Refer to section Switch Events on page 16-1 for details.
4	FMCI	RW1C	0x0	**Failover Mode Change Initiated.** This bit is set in an upstream port whenever failover is enabled in the partition associated with this port (i.e., the FEN bit is set in the corresponding SWPARTxCTL register) and a failover mode change is initiated by the corresponding failover capability structure (i.e., the FMCI bit in the FCAPxSTS register transitions from 0x0 to 0x1). This bit is read-only with a value of zero in downstream switch ports.
5	FMCC	RW1C	0x0	**Failover Mode Change Completed.** This bit is set in an upstream port whenever failover is enabled in the partition associated with this port (i.e., the FEN bit is set in the corresponding SWPARTxCTL register) and a failover mode change is completed by the corresponding failover capability structure (i.e., the FMCC bit in the FCAPxSTS register transitions from 0x0 to 0x1). This bit is read-only with a value of zero in downstream switch ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	Reserved	RO	0x0	Reserved field.
7	TMPSENSOR	RW1C	0x0	**Temperature Sensor Alarm.** This bit is set when a temperature sensor alarm is triggered (i.e., one of the temperature threshold bits in the TMPSTS register transitions from 0x0 to 0x1, and the corresponding bit is enabled in the TMPCTL register). This bit is read-only with a value of zero in downstream switch ports.
31:8	Reserved	RO	0x0	Reserved field.

NTINTMSK - NT Endpoint Interrupt Mask (0x408)

Bit Field	Field Name	Type	Default Value	Description
0	MSG	RW	0x1	**Message Interrupt.** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
1	DBELL	RW	0x1	**Doorbell Interrupt.** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
2	Reserved	RO	0x0	Reserved field.
3	SEVENT	RW	0x1	**Switch Event.** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
4	FMCI	RW	0x1	**Failover Mode Change Initiated** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
5	FMCC	RW	0x1	**Failover Mode Change Completed** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
6	Reserved	RO	0x0	Reserved field.
7	TMPSENSOR	RW	0x1	**Temperature Sensor Alarm** When this bit is set, the corresponding bit in the NTINTSTS register is masked from generating an interrupt.
31:8	Reserved	RO	0x0	Reserved field.

NTSDATA - NT Endpoint Signal Data (0x40C)

Bit Field	Field Name	Type	Default Value	Description
31:0	SDATA	RW	0x0 SWSticky	**Switch Signal Data.** This is a general 32-bit read write field that may be used in conjunction with switch signals.

Notes

NTGSIGNAL - NT Endpoint Global Signal (0x410)

Bit Field	Field Name	Type	Default Value	Description
0	GSIGNAL	RW	0x0	**Global Signal** Writing a one to a bit in this field generates a switch signal to the partition associated with this NT function. This results in the bit corresponding to the partition being set in the Global Signal (GSIGNAL) field in the Switch Event Global Signal Status (SEGSIGSTS) register. This field always returns a value of zero when read.
31:1	Reserved	RO	0x0	Reserved field.

Internal Error Reporting Masks

NTIERRORMSK0 - Internal Error Reporting Mask 0 (0x414)

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW	0x0 SWSticky	**IFB Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	IFBNPTLPTO	RW	0x0 SWSticky	**IFB Non-Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
2	IFBCPTLPTO	RW	0x0 SWSticky	**IFB Completion TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3				Reserved field.
4	EFBPTLPTO	RW	0x0 SWSticky	**EFB Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	EFBNPTLPTO	RW	0x0 SWSticky	**EFB Non-Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	EFBCPTLPTO	RW	0x0 SWSticky	**EFB Completion TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
7	IFBDATSBE	RW	0x0 SWSticky	**IFB Data Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
8	IFBDATDBE	RW	0x0 SWSticky	**IFB Data Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
9	IFBCTLSBE	RW	0x0 SWSticky	**IFB Control Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
10	IFBCTLDBE	RW	0x0 SWSticky	**IFB Control Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11	EFBDATSBE	RW	0x0 SWSticky	**EFB Data Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
12	EFBDATDBE	RW	0x0 SWSticky	**EFB Data Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
13	EFBCTLSBE	RW	0x0 SWSticky	**EFB Control Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	EFBCTLDBE	RW	0x0 SWSticky	**EFB Control Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
15	E2EPE	RW	0x0 SWSticky	**End-to-End Data Path Parity Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
16	ULD	RW	0x0 SWSticky	**Unreliable Link Detected** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
17	RBCTLSBE	RW	0x0 SWSticky	**Replay Buffer Control Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
18	RBCTLDBE	RW	0x0 SWSticky	**Replay Buffer Control Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19	DIFBPTLPTO	RW	0x1 SWSticky	**DMA IFB Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
20	DIFBNPTLPTO	RW	0x1 SWSticky	**DMA IFB Non-Posted TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
21	DIFBCPTLPTO	RW	0x1 SWSticky	**DMA IFB Completion TLP Time-Out** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW	0x1 SWSticky	**DMA IFB Data Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
24	DIFBDATDBE	RW	0x1 SWSticky	**DMA IFB Data Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
25	DIFBCTLSBE	RW	0x1 SWSticky	**DMA IFB Control Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
26	DIFBCTLDBE	RW	0x1 SWSticky	**DMA IFB Control Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

Notes

Bit Field	Field Name	Type	Default Value	Description
27	DEFBDATSBE	RW	0x1 SWSticky	**DMA EFB Data Single Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
28	DEFBDATDBE	RW	0x1 SWSticky	**DMA EFB Data Double Bit Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.
30:29	Reserved	RW	0x1 SWSticky	This field is reserved but remains read-write in the hardware. Modifying this field has no effect other than changing the value of the field.
31	DE2EPE	RW	0x1 SWSticky	**DMA End-to-End Data Path Parity Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register. This bit is only applicable for ports that contain a DMA function. When not applicable, this bit remains read-write but has no effect.

NTIERRORMSK1 - Internal Error Reporting Mask 1 (0x418)

Bit Field	Field Name	Type	Default Value	Description
0	P0AER	RW	0x1 SWSticky	**Port 0 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	Reserved	RO	0x0	Reserved field.
2	P2AER	RW	0x1 SWSticky	**Port 2 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	P4AER	RW	0x1 SWSticky	**Port 4 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	Reserved	RO	0x0	Reserved field.
6	P6AER	RW	0x1 SWSticky	**Port 6 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
7	Reserved	RO	0x0	Reserved field.
8	P8AER	RW	0x1 SWSticky	**Port 8 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11:9	Reserved	RO	0x0	Reserved field.
12	P12AER	RW	0x1 SWSticky	**Port 12 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
15:13	Reserved	RO	0x0	Reserved field.
16	P16AER	RW	0x1 SWSticky	**Port 16 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19:17	Reserved	RO	0x0	Reserved field.
20	P20AER	RW	0x1 SWSticky	**Port 20 AER Error** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the NT function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
31:21	Reserved	RO	0x0	Reserved field.

Notes

Doorbell Registers

OUTDBELLSET - NT Outbound Doorbell Set (0x420)

Bit Field	Field Name	Type	Default Value	Description
31:0	OUTDBELL-SET	RW	0x0	**Outbound Doorbell Set.** Each bit in this field corresponds to one of the 32 outbound doorbells associated with the NT endpoint. Writing a one to a bit in this field sets the corresponding bit in this field and initiates an outbound doorbell request. Writing a zero to a bit in this field has no effect. Reading this field returns 0x0. Refer to section Doorbell Registers on page 14-16 for details.

INDBELLSTS - NT Inbound Doorbell Status (0x428)

Bit Field	Field Name	Type	Default Value	Description
31:0	INDBELLSTS	RW1C	0x0	**Inbound Doorbell Status.** Each bit in this field corresponds to one of the 32 inbound doorbells associated with the NT endpoint. A bit in this field is set when the corresponding inbound partition doorbell bit active. Refer to section Doorbell Registers on page 14-16 for details.

INDBELLMSK - NT Inbound Doorbell Mask (0x42C)

Bit Field	Field Name	Type	Default Value	Description
31:0	INDBELLMSK	RW	0x0	**Inbound Doorbell Mask.** Each bit in this field corresponds to one of the 32 inbound doorbells associated with the NT endpoint. When a bit in this field is set, the corresponding bit in the NT inbound doorbell status register is masked from generating an NT interrupt. Refer to section Doorbell Registers on page 14-16 for details.

Notes

Message Registers

OUTMSG[3:0] - Outbound Message[3:0] (0x430-43C)

Bit Field	Field Name	Type	Default Value	Description
31:0	OUTMSG	RW	0x0	**Outbound Message.** Writing a value to this field updates the value in the Inbound Message (INMSG) field of the Inbound Message Register (INMSGx) selected by Partition (PART) and Register (REG) fields in the Partition x Message Control (SWPxMSGCTLy) register if the INMSGx register is not full. See section Message Registers on page 14-17 for a description of the message registers. Reading this field returns the previous value written.

INMSG[3:0] - Inbound Message [3:0] (0x440-44C)

Bit Field	Field Name	Type	Default Value	Description
31:0	INMSG	RO	0x0	**Inbound Message.** This read only field contains the value written by an agent to an Outbound Message (OUTMSG) register that is mapped to this register. See section Message Registers on page 14-17 for a description of the message registers.

INMSGSRC[3:0] - Inbound Message Source [3:0] (0x450-45C)

Bit Field	Field Name	Type	Default Value	Description
3:0	SRC	RO	0x0	**Inbound Message Source Partition.** This read only field contains the partition number of the agent that caused the Inbound Message (INMSG) field in the Inbound Message (INMSG) register to be updated. Valid values are 0 to 7.
31:4	Reserved	RO	0x0	Reserved field.

MSGSTS - Message Status (0x460)

Bit Field	Field Name	Type	Default Value	Description
0	OUTMSGSTS0	RW1C	0x0	**Outbound Message 0 Status.** This bit is set when a write to the OUTMSG0 register fails. See section Message Registers on page 14-17 for a description of the message registers.
1	OUTMSGSTS1	RW1C	0x0	**Outbound Message 1 Status.** This bit is set when a write to the OUTMSG1 register fails. See section Message Registers on page 14-17 for a description of the message registers.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	OUTMSGSTS2	RW1C	0x0	**Outbound Message 2 Status.** This bit is set when a write to the OUTMSG2 register fails. See section Message Registers on page 14-17 for a description of the message registers. Valid values are 0 to 7.
3	OUTMSGSTS3	RW1C	0x0	**Outbound Message 3 Status.** This bit is set when a write to the OUTMSG3 register fails. See section Message Registers on page 14-17 for a description of the message registers.
15:4	Reserved	RO	0x0	Reserved field.
16	INMSGSTS0	RW1C	0x0	**Inbound Message 0 Status.** This bit is set when the INMSG0 register is updated. While this bit is set, the INMSG0 register is full and subsequent updates fail. See section Message Registers on page 14-17 for a description of the message registers.
17	INMSGSTS1	RW1C	0x0	**Inbound Message 1 Status.** This bit is set when the INMSG1 register is updated. While this bit is set, the INMSG1 register is full and subsequent updates fail. See section Message Registers on page 14-17 for a description of the message registers.
18	INMSGSTS2	RW1C	0x0	**Inbound Message 2 Status.** This bit is set when the INMSG2 register is updated. While this bit is set, the INMSG2 register is full and subsequent updates fail. See section Message Registers on page 14-17 for a description of the message registers.
19	INMSGSTS3	RW1C	0x0	**Inbound Message 3 Status.** This bit is set when the INMSG3 register is updated. While this bit is set, the INMSG3 register is full and subsequent updates fail. See section Message Registers on page 14-17 for a description of the message registers.
31:20	Reserved	RO	0x0	Reserved field.

MSGSTSMSK - Message Status Mask (0x464)

Bit Field	Field Name	Type	Default Value	Description
0	OUTMSGSTS0	RW	0x0	**Outbound Message 0 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
1	OUTMSGSTS1	RW	0x0	**Outbound Message 1 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
2	OUTMSGSTS2	RW	0x0	**Outbound Message 2 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	OUTMSGSTS3	RW	0x0	**Outbound Message 3 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
15:4	Reserved	RO	0x0	Reserved field.
16	INMSGSTS0	RW	0x0	**Inbound Message 0 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
17	INMSGSTS1	RW	0x0	**Inbound Message 1 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
18	INMSGSTS2	RW	0x0	**Inbound Message 2 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
19	INMSGSTS3	RW	0x0	**Inbound Message 3 Mask.** When this bit is set, assertion of the corresponding bit in the MSGSTS register is masked from generating an interrupt.
31:20	Reserved	RO	0x0	Reserved field.

BAR Configuration

BARSETUP0 - BAR 0 Setup (0x470)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 - (memory) memory space 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.

Notes

Bit Field	Field Name	Type	Default Value	Description
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. Setting the SIZE field to a value greater than 31 when the MEMSI and TYPE fields in this register select 32-bit memory space results in bits greater than 32 being ignored (i.e., only the TYPE field can enable 64-bit addressing).
10	MODE	RW	0x0 SWSticky	**BAR Mode.** This field selects the operating mode of the BAR. 0x0 - (window) address window. 0x1 - (cfgspace) mapped configuration space. When this field is set to 0x1, the SIZE field in this register must be set to 0xC (i.e., BAR size is 4 KB).
12:11	ATRAN	RO	0x0	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. This field is read-only with a value of zero since BAR 0 only supports direct address translation. 0x0 - (direct) direct address translation others - reserved
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT0 - BAR 0 Limit Address (0x474)

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	0x0	Reserved field.
31:10	LADDR	RW	0x3F_FFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the BAR is configured to operate as a 64-bit address window, this field acts as the lower bits of the LADDR field while the upper bits are provided by the BARLIMIT1 register. When the MODE field in the BARSETUP0 register is set to 0x1 (i.e., the BAR is mapped to configuration space), this field must be set to a value equal to or greater than 0x3 (i.e., limit address is >= to the BAR base address + 4 KB).

BARLTBASE0 - BAR 0 Lower Translated Base Address (0x478)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

Notes

BARUTBASE0 - BAR 0 Upper Translated Base Address (0x47C)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARSETUP1 - BAR 1 Setup (0x480)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (memory) memory space 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space 0x1 - (reserved) reserved 0x2 - (addr64) 64-bit addressing 0x3 - (reserved) reserved

Notes

Bit Field	Field Name	Type	Default Value	Description
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains RW but has no functional effect on the operation of the device. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 32 being ignored (i.e., odd BARs only support 32-bit addressing).
10	MODE	RO	0x0	**BAR Mode.** This field selects the operating mode of the BAR. This field is read-only with a value of zero since BAR 1 only supports address window mode operation. 0x0 - (window) address window. 0x1 - reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
12:11	ATRAN	RO	0x0	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. This field is read-only with a value of zero since BAR 1 only supports direct address translation. 0x0 - (direct) direct address translation others - reserved
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT1 - BAR 1 Limit Address (0x484)

When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write, default to the value 0xFFFF_FFFF, and act as the upper bits of the LADDR field in the BARLIMIT0 register.

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	See Description	**Reserved.** When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write SWSticky and act as the upper bits of the LADDR field in the BARLIMIT0 register.
31:10	LADDR	RW	0x3F_FFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, these bits act as the upper bits of the LADDR field in the BARLIMIT0 register.

Notes

BARLTBASE1 - BAR 1 Lower Translated Base Address (0x488)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARUTBASE1 - BAR 1 Upper Translated Base Address (0x48C)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARSETUP2 - BAR 2 Setup (0x490)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 - (memory) memory space. 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. When the BAR is configured to operate as an address window with lookup table address translation, valid values for the SIZE field are 14 through 37 (values greater than 31 require a 64-bit BAR). Setting the SIZE field outside this range produces undefined results. Setting the SIZE field to a value greater than 31 when then MEMSI and TYPE fields in this register select 32-bit memory space, results in bits greater than 31 being ignored (i.e., only the TYPE field can enable 64-bit addressing).
10	MODE	RO	0x0	**BAR Mode.** This field selects the operating mode of the BAR. This field is read-only with a value of zero since BAR 2 only supports address window mode operation. 0x0 - (window) address window. 0x1 - reserved.
12:11	ATRAN	RW	0x0 SWSticky	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. 0x0 - (direct) direct address translation 0x1 - (lookup12) 12-entry lookup table address translation 0x2 - (lookup24) 24-entry lookup table address translation 0x3 - reserved

Notes

Bit Field	Field Name	Type	Default Value	Description
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT2 - BAR 2 Limit Address (0x494)

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	0x0	Reserved field.
31:10	LADDR	RW	0xFFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the BAR is configured to operate as a 64-bit address window, this field acts as the lower bits of the LADDR field while the upper bits are provided by the BARLIMIT1 register.

BARLTBASE2 - BAR 2 Lower Translated Base Address (0x498)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

Notes

BARUTBASE2 - BAR 2 Upper Translated Base Address (0x49C)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARSETUP3 - BAR 3 Setup (0x4A0)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (memory) memory space 0x1 - reserved.
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.

Notes

Bit Field	Field Name	Type	Default Value	Description
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains RW but has no functional effect on the operation of the device. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 31 being ignored (i.e., odd BARs only support 32-bit addressing).
10	MODE	RO	0x0	**BAR Mode.** This field selects the operating mode of the BAR. This field is read-only with a value of zero since BAR 3 only supports address window mode operation. 0x0 - (window) address window. 0x1 - reserved.
12:11	ATRAN	RO	0x0	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. This field is read-only with a value of zero since BAR 3 only supports direct address translation. 0x0 - (direct) direct address translation others - reserved
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR3 takes on the function of the upper 32-bits of the BADDR field in BAR2. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT3 - BAR 3 Limit Address (0x4A4)

When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write, default to the value 0xFFFF_FFFF, and act as the upper bits of the LADDR field in the BARLIMIT2 register.

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	See Description	**Reserved.** When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write SWSticky and act as the upper bits of the LADDR field in the BARLIMIT3 register.
31:10	LADDR	RW	0x3F_FFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the MEMSI field in BARSETUP2 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, these bits act as the upper bits of the LADDR field in the BARLIMIT2 register.

BARLTBASE3 - BAR 3 Lower Translated Base Address (0x4A8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARUTBASE3 - BAR 3 Upper Translated Base Address (0x4AC)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARSETUP4 - BAR 4 Setup (0x4B0)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 - (memory) memory space. 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.

Notes

Bit Field	Field Name	Type	Default Value	Description
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR or BAR pair when 64-bit addressing is selected. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space and 2^{63} bytes for 64-bit address space. When the BAR is configured to operate as an address window with lookup table address translation, valid values for the SIZE field are 14 through 37 (values greater than 31 require a 64-bit BAR). Setting the SIZE field outside this range produces undefined results. Setting the SIZE field to a value greater than 31 when then MEMSI and TYPE fields in this register select 32-bit memory space, results in bits greater than 31 being ignored (i.e., only the TYPE field can enable 64-bit addressing).
10	MODE	RO	0x0	**BAR Mode.** This field selects the operating mode of the BAR. This field is read-only with a value of zero since BAR 4 only supports address window mode operation. 0x0 - (window) address window. 0x1 - reserved.
12:11	ATRAN	RW	0x0 SWSticky	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. BAR 4 only supports direct address translation and 16-entry lookup table address translation. 0x0 - (direct) direct address translation 0x1 - (lookup16) 16-entry lookup table address translation Others - reserved
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT4 - BAR 4 Limit Address (0x4B4)

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	0x0	Reserved field.
31:10	LADDR	RW	0x3F_FFFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the BAR is configured to operate as a 64-bit address window, this field acts as the lower bits of the LADDR field while the upper bits are provided by the BARLIMIT5 register.

BARLTBASE4 - BAR 4 Lower Translated Base Address (0x4B8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

Notes

BARUTBASE4 - BAR 4 Upper Translated Base Address (0x4BC)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARSETUP5 - BAR 5 Setup (0x4C0)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (memory) memory space 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
9:4	SIZE	RW	0x0 SWSticky	**Address Space Size.** This field selects the size, in address bits, of the address space for the corresponding BAR. When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains RW but has no functional effect on the operation of the device. Assuming the size field is set to a valid value, the size of the address space requested by the BADDR field in the corresponding BAR is equal to 2^{SIZE}. Bits in the BAR BADDR field correspond to PCI Express address bits. For example, bit 0 or the BAR BADDR field corresponds to PCI Express Address bit 4. Setting this SIZE field to a non-zero value allows bits in the BAR BADDR field that correspond to PCI Express address bits greater than or equal to the SIZE field to be modified. Corresponding bits less than the SIZE field and greater than or equal to four always return a value of zero when read and cannot be modified. Setting the SIZE field to a value less than four results in all bits in the corresponding BAR BADDR field to take on a read-only zero value that effectively disables the BAR. The smallest memory size that may be requested by PCI Express is 128 (i.e., SIZE equal to 7) and the largest is 2^{31} bytes for 32-bit address space. Setting the SIZE field to a value greater than 31 results in bits greater than 31 being ignored (i.e., odd BARs only support 32-bit addressing).
10	MODE	RO	0x0	**BAR Mode.** This field selects the operating mode of the BAR. This field is read-only with a value of zero since BAR 5 only supports address window mode operation. 0x0 - (window) address window. 0x1 - reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
12:11	ATRAN	RO	0x0	**Address Translation.** When the BAR is configured to operate as an address window, this field specifies the type of address translation that is used. This field is read-only with a value of zero since BAR 5 only supports direct address translation. 0x0 - (direct) direct address translation others - reserved
15:13	TPART	RW	0x0 SWSticky	**Translated Partition.** When the BAR is configured to operate as an address window with direct address translation, this field specifies the translated partition number.
30:16	Reserved	RO	0x0	Reserved field.
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR5 takes on the function of the upper 32-bits of the BADDR field in BAR4. In this mode, this field remains RW but has no functional effect on the operation of the device. 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

BARLIMIT5 - BAR 5 Limit Address (0x4C4)

When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write, default to the value 0xFFFF_FFFF, and act as the upper bits of the LADDR field in the BARLIMIT4 register.

Bit Field	Field Name	Type	Default Value	Description
9:0	Reserved	RO	See Description	**Reserved.** When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, all 32-bits of this register become read-write SWSticky and act as the upper bits of the LADDR field in the BARLIMIT4 register.
31:10	LADDR	RW	0x3F_FFFF SWSticky	**Limit Address.** When the BAR is configured to operate as an address window, this field specifies the limit address associated with the BAR. When the MEMSI field in BARSETUP4 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, these bits act as the upper bits of the LADDR field in the BARLIMIT4 register.

Notes

BARLTBASE5 - BAR 5 Lower Translated Base Address (0x4C8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 2 through 31 of the translated base address. The corresponding BAR upper translated base address register contains the upper 32-bits of the address. Since the translated base address must be DWord aligned, the bottom two bits of the address are always zero. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

BARUTBASE5 - BAR 5 Upper Translated Base Address (0x4CC)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	0x0 SWSticky	**Translated Base Address.** When the BAR is configured for direct address translation, this field specifies the translated base address. The translated base address is 64-bits. This field contains bits 32 through 63 of the translated base address. The corresponding BAR lower translated base address register contains the lower bits of the address. Refer to section Non Transparent Operation Restrictions on page 14-40 for restrictions on programming this field.

Mapping Table

NTMTBLADDR - NT Mapping Table Address (0x4D0)

Bit Field	Field Name	Type	Default Value	Description
6:0	ADDR	RW	0x0	**NT Mapping Table Address** This field specifies the partition NT Mapping table entry accessed by the NTMTBLDATA register. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.
31:7	Reserved	RO	0x0	Reserved field.

Notes

NTMTBLSTS - NT Mapping Table Status (0x4D4)

Bit Field	Field Name	Type	Default Value	Description
0	ERR	RW1C	0x0 Sticky	**NT Mapping Table Access Error** This bit is set if an invalid partition NT Mapping Table entry is accessed or when an NT Mapping Table protection violation occurs.
31:1	Reserved	RO	0x0	Reserved field.

NTMTBLDATA - NT Mapping Table Data (0x4D8)

Bit Field	Field Name	Type	Default Value	Description
0	VALID	RW	- SWSticky	**Valid** Reading this field returns the VALID field of the NT Mapping table entry specified by the partition NT Mapping table address in the NTMTBLADDR register. Writing to this field updates the VALID field of the NT Mapping table entry specified by the partition NT Mapping table address. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.
3:1	FUNC	RW	- SWSticky	**Function** Reading this field returns the FUNC field of the NT Mapping table entry specified by the partition NT Mapping table address in the NTMTBLADDR register. Writing to this field updates the FUNC field of the NT Mapping table entry specified by the partition NT Mapping table address. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.
8:4	DEV	RW	- SWSticky	**Device** Reading this field returns the DEV field of the NT Mapping table entry specified by the partition NT Mapping table address in the NTMTBLADDR register. Writing to this field updates the DEV field of the NT Mapping table entry specified by the partition NT Mapping table address. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.
16:9	BUS	RW	- SWSticky	**Bus** Reading this field returns the BUS field of the NT Mapping table entry specified by the partition NT Mapping table address in the NTMTBLADDR register. Writing to this field updates the BUS field of the NT Mapping table entry specified by the partition NT Mapping table address. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.

Notes

Bit Field	Field Name	Type	Default Value	Description
19:17	PART	RW	- SWSticky	**Partition** Reading this field returns the PART field of the NT Mapping table entry specified by the partition NT Mapping table address in the NTMTBLADDR register. Writing to this field updates the PART field of the NT Mapping table entry specified by the partition NT Mapping table address. The actual physical NT Mapping table entry accessed is determined as described in section NT Mapping Table on page 14-9.
28:20	Reserved	RO	0x0	Reserved field.
29	ATP	RW	- SWSticky	**Address Type Processing.** This field specifies the processing of the address type (AT) field on request TLPs. 0x0 - (untranslated) AT field set to untranslated in TLP emitted by NT function 0x1 - (translated) AT field set to translated in TLP emitted by NT function Refer to section Address Type Processing on page 14-15.
30	CNS	RW	- SWSticky	**Completion No Snoop Processing** This field specifies the processing performed on the no snoop attribute in the header of completion TLPs. 0x0 - (nochange) no change 0x1 - (invert) invert no snoop attribute Refer to section No Snoop Processing on page 14-14.
31	RNS	RW	- SWSticky	**Request No Snoop Processing** This field specifies the processing performed on the no snoop attribute in the header of request TLPs. 0x0 - (nochange) no change 0x1 - (invert) invert no snoop attribute Refer to section No Snoop Processing on page 14-14.

REQIDCAP - Requester ID Capture (0x4DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	REQID	RO	-	**Requester ID Capture** When read, this field returns the requester ID of the PCI Express agent that issued the read request to this register. This register may be used by software as an aid in programming the NT Mapping Table. Refer to section Requester ID Capture Register on page 14-14 for details.
31:16	Reserved	RO	0x0	Reserved field.

Notes

Lookup Table

LUTOFFSET - Lookup Table Offset (0x4E0)

Bit Field	Field Name	Type	Default Value	Description
4:0	INDEX	RW	0x0 SWSticky	**Lookup Table Index.** This field selects the index of the lookup table accessed when the lookup table data registers (i.e., LUTLDATA, LUTMDATA and LUTUDATA) are read or written. Selecting a value that is outside the range supported by a lookup table configuration produces undefined results. For example, when BAR 2 is configured for 24 entry lookup table and the BAR field in this register selects BAR 2, the INDEX field must only be set to values 0 to 23. On the other hand, if BAR 2 is configured for 12 entry lookup table and the BAR field in this register selects BAR 2, the INDEX field must only be set to values 0 to 11. Similarly, if BAR 4 is configured for 12 entry lookup table and the BAR field in this register selects BAR 4, the INDEX field must only be set to values 0 to 11.
7:5	Reserved	RO	0x0	Reserved field.
10:8	BAR	RW	0x0 SWSticky	**Lookup Table BAR Select.** This field selects the BAR of the lookup table accessed when the lookup table data registers (i.e., LUTLDATA, LUTMDATA and LUTUDATA) are read or written. Selecting a reserved value produces undefined results. 0x0 - reserved 0x1 - reserved 0x2 - (bar2) BAR 2 0x3 - reserved 0x4 - (bar4) BAR 4 0x5 - reserved 0x6 - reserved 0x7 - reserved
31:11	Reserved	RO	0x0	Reserved field.

LUTLDATA - Lookup Table Lower Data (0x4E4)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	TADDR	RW	- SWSticky	**Translated Base Address.** This field contains bits 31 through 2 of the translated base address field associated with the lookup table entry selected by the BAR and INDEX fields of the Lookup Table Offset (LUTOFFSET) register. The value read from this field corresponds to the value of the corresponding lookup entry table field. The value written to this field updates the corresponding table entry field

Notes

LUTMDATA - Lookup Table Middle Data (0x4E8)

Bit Field	Field Name	Type	Default Value	Description
31:0	TADDR	RW	- SWSticky	**Translated Base Address.** This field contains bits 63 through 32 of the translated base address field associated with the lookup table entry selected by the BAR and INDEX fields of the Lookup Table Offset (LUTOFFSET) register. The value read from this field corresponds to the value of the corresponding lookup entry table field. The value written to this field updates the corresponding table entry field

LUTUDATA - Lookup Table Upper Data (0x4EC)

Bit Field	Field Name	Type	Default Value	Description
3:0	PART	RW	- SWSticky	**Partition.** This field contains the partition field of the lookup table entry selected by the BAR and INDEX fields of the Lookup Table Offset (LUTOFFSET) register. The value read from this field corresponds to the value of the corresponding lookup entry table field. The value written to this field updates the corresponding table entry field. pdates the corresponding table entry field. 0x0 - Partition 0 0x1 - Partition 1 0x2 - Partition 2 0x3 - Partition 3 0x4 - Partition 4 0x5 - Partition 5 0x6 - Partition 6 0x7 - Partition 7 Others - reserved
30:4	Reserved	RO	0x0	Reserved field.
31	VALID	RW	0x0 SWSticky	**Valid.** This field contains the valid field of the lookup table entry selected by the BAR and INDEX fields of the Lookup Table Offset (LUTOFFSET) register. The value read from this field corresponds to the value of the corresponding lookup entry table field. The value written to this field updates the corresponding table entry field

AER Error Emulation

NTUEEM - NT Endpoint Uncorrectable Error Emulation (0x4F0)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	DLPERR	RW	0x0 SWSticky	**Data Link Protocol Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
11:5	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 SWSticky	**Poisoned TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
15:13	Reserved	RO	0x0	Reserved field.
16	UECOMP	RW	0x0 SWSticky	**Unexpected Completion Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
17	RCVOVR	RW	0x0 SWSticky	**Receiver Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
18	MALFORMED	RW	0x0 SWSticky	**Malformed TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
19	ECRC	RW	0x0 SWSticky	**ECRC Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
20	UR	RW	0x0 SWSticky	**UR Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
21	ACSV	RW	0x0 SWSticky	**ACS Violation Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
22	UIE	RW	0x0 SWSticky	**Uncorrectable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
23	MCBLKTLP	RW	0x0 SWSticky	**MC Blocked TLP Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERUES register. This bit always returns 0x0 when read.
30:24	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31	ADVISORYNF	RW	0x0 SWSticky	**Advisory Non-Fatal Error Trigger.** If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible (refer to the PCI Express Base Specification), an advisory non-fatal error is logged an reported in the NT function's AER capability structure, provided the error severity for the selected uncorrectable error is configured such that the error will be of type non-fatal. If this bit is set together with another error bit in this register for which an advisory non-fatal error is not possible, the operation is undefined. If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible, but the severity of the selected uncorrectable error is fatal, then this bit is ignored and the selected error is logged and reported as a fatal error.

NTCEEM - NT Endpoint Correctable Error Emulation (0x4F4)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 SWSticky	**Receiver Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 SWSticky	**Bad TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
7	BADDLLP	RW	0x0 SWSticky	**Bad DLLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
8	RPLYROVR	RW	0x0 SWSticky	**Replay Number Rollover Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 SWSticky	**Replay Timer Timeout Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
13	Reserved	RO	0x0	Reserved field.
14	CIE	RW	0x0 SWSticky	**Correctable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW	0x0 SWSticky	**Header Log Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the AERCES register. This bit always returns 0x0 when read.
31:16	Reserved	RO	0x0	Reserved field.

Punch-Through Configuration Registers

Refer to section Punch-Through Configuration Requests on page 14-18 for a detailed description of punch-through configuration requests.

PTCCTL0 - Punch-Through Configuration Control 0 (0x510)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
7:2	REG	RW	0x0	**Register Number** This field selects the configuration register number (as defined by Section 7.2.2 of the PCI Express Base Specification) in the punch-through configuration request
11:8	EREG	RW	0x0	**Extended Register Number** This field selects the extended configuration register number (as defined by Section 7.2.2 of the PCI Express Base Specification) in the punch-through configuration request.
15:12	Reserved	RO	0x0	Reserved field.
18:16	FUNC	RW	0x0	**Function Number** This field selects the function number (as defined by Section 7.2.2 of the PCI Express Base Specification) in the punch-through configuration request completer ID field.
23:19	DEV	RW	0x0	**Device Number** This field selects the device number (as defined by Section 7.2.2 of the PCI Express Base Specification) in the configuration request completer ID field.
31:24	BUS	RW	0x0	**Bus Number** This field selects the bus number (as defined by Section 7.2.2 of the PCI Express Base Specification) in the configuration request completer ID field.

Notes

PTCCTL1 - Punch-Through Configuration Control 1 (0x514)

Bit Field	Field Name	Type	Default Value	Description
0	CFGTYPE	RW	0x0	**Configuration Access Type** This field selects the type of configuration access generated using the punch-through mechanism. 0x0 - (type0) type 0 configuration access 0x1 - (type1) type 1 configuration access
1	OP	RW	0x0	**Operation** This field selects the type of configuration operation to be performed when the PTCDATA register is written. 0x0 - (read) configuration read 0x1 - (write) configuration write
31:2	Reserved	RO	0x0	Reserved field.

PTCDATA - Punch-Through Data (0x518)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Configuration Data.** A write to this field will generate a configuration read or write transaction, as selected by the OP field in the PTCCFG1 register, on the NT endpoint's link. The byte enables in the generated transaction match those of the write to this register. When a configuration write operation is selected, the value written to this field is the value used in the configuration write transaction. When a configuration read operation is selected, the value written to this field is ignored and the value returned by the read may be read from this field when the DONE bit is set in the PTCSTS register. Status for the generated transaction is reported in the PTCSTS register.

PTCSTS - Punch-Through Status (0x51C)

Bit Field	Field Name	Type	Default Value	Description
0	BUSY	RO	0x0	**Punch-Through Configuration Interface Busy** This bit is set when a punch-through configuration transaction is in progress. 0x0 - (idle) configuration transaction interface is idle 0x1 - (busy) configuration transaction in progress

Notes

Bit Field	Field Name	Type	Default Value	Description
1	DONE	RW1C	0x0	**Punch-Through Configuration Transaction Completed.** This bit is set when a punch-through configuration transaction has completed and the STATUS field is valid. Writing a one to this bit clears the status bit or aborts a punch-through operation in progress. 0x0 - (notdone) configuration transaction interface is idle or transaction in flight. 0x1 - (done) configuration transaction completed
4:2	STATUS	RO	0x0	**Punch-Through Configuration Transaction Status.** This field contains the completion status of the last punch-through configuration transaction and is valid only when the DONE bit in this register is set. 0x0 - (sc) successful completion 0x1 - (ur) unsupported request 0x2 - (crs) configuration request retry 0x3 - (ra) requester abort 0x4 - (ca) completer abort others - reserved
5	PTABORT	RO	0x0	**Punch-Through Abort Status.** This bit is set if the last punch-through configuration transaction was aborted (i.e., the STATUS field in this register is set to requester abort). This bit will remain set until the next punch-through configuration transaction is initiated.
31:6	Reserved	RO	0x0	Reserved field.

NT Multicast

NTMCG[3:0]PA - NT Multicast Group x Port Association (0x600-60C)

Bit Field	Field Name	Type	Default Value	Description
23:0	PORTVEC	RW	0x0	**Port Vector.** Each bit in this field corresponds to a port (i.e., bit 0 corresponds to port 0, bit 1 corresponds to port 1, and so on). When a bit is set, the corresponding port transmits a multicast TLP when the TLP is received by this NT function and is associated with group x. For example, setting bit[0] in the NTMCG[2] register causes TLPs received by this NT function that are associated with group 2 to be NT multicasted to port 0. Operation is undefined when a bit is set corresponding to a port that is not enabled in an operational mode (i.e., the port is unattached or disabled) or does not exist in the device. Refer to section Non-Transparent Multicast Operation on page 17-6 for details on programming this field.
31:24	Reserved	RO	0x0	Reserved field.

Notes

Global Address Space Access Registers

GASAADDR - Global Address Space Access Address (0xFF8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
18:2	GADDR	RW	0x0	**Global Address.** This field selects the system address of the register to be accessed via the GASADATA register. The value of this register must not be programmed to point to the address of the GASAADDR or GASADATA register in this or any other port. Similarly, the value of this register must not be programmed to point to the address of the Extended Configuration Address (ECFGADDR) or Extended Configuration Data registers (ECFGDATA) in this or any other function. Violations of these rules produce undefined results.
31:19	Reserved	RO	0x0	Reserved field.

GASADATA - Global Address Space Access Data (0xFFC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Data.** A read from this field will return the global space register value pointed to by the GASAADDR register. A write to this field will update the contents of the global space register pointed to by the GASAADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

DMA Function Registers

Notes

Type 0 Configuration Header Registers

VID - Vendor Identification (0x000)

Bit Field	Field Name	Type	Default Value	Description
15:0	VID	RO	0x111D	**Vendor Identification.** This field contains the 16-bit vendor ID value assigned to IDT. See section Vendor ID on page 1-1.

DID - Device Identification (0x002)

Bit Field	Field Name	Type	Default Value	Description
15:0	DID	RO	-	**Device Identification.** This field contains the 16-bit device ID assigned by IDT to this device. See section Device ID on page 1-1.

PCICMD - PCI Command (0x004)

Bit Field	Field Name	Type	Default Value	Description
0	IOAE	RO	0x0	**I/O Access Enable.** The DMA function does not implement Base Address Register (BAR) apertures in I/O space. All I/O accesses received by the DMA function are treated as Unsupported Requests. As a result, this bit is hardwired to read-only zero.
1	MAE	RW	0x0	**Memory Access Enable.** When this bit is cleared, the function does not respond to memory and prefetchable memory space accesses received on its primary bus (i.e., a TLP that targets the function's memory or prefetchable memory space is treated as an Unsupported Request). 0x0 - (disable) Disable memory space. 0x1 - (enable) Enable memory space.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	BME	RW	0x0	**Bus Master Enable.** When this bit is set, the DMA function is allowed to issue memory requests. When this bit is cleared, the DMA function does not transmit memory requests. Note that the DMA function never issues I/O requests. Also, note that clearing this bit inhibits the DMA function from issuing MSI requests. Completions or messages generated by the function (i.e., a completion corresponding to a configuration request) are not affected by this bit. 0x0 - (disable) Disable transmission of memory requests. 0x1 - (enable) Enable transmission of memory requests.
3	SSE	RO	0x0	**Special Cycle Enable.** Not applicable.
4	MWI	RO	0x0	**Memory Write Invalidate.** Not applicable.
5	VGAS	RO	0x0	**VGA Palette Snoop.** Not applicable.
6	PERRE	RW	0x0	**Parity Error Enable.** This bit controls the logging of poisoned TLPs in the Master Data Parity Error Detected (MDPED) field in the PCI Status (PCISTS) register. When this bit is cleared, poisoned TLPs are not reported as master data parity errors in the PCISTS register.
7	ADSTEP	RO	0x0	**Address Data Stepping.** Not applicable.
8	SERRE	RW	0x0	**SERR Enable.** Non-fatal and fatal errors detected by the function are reported to the Root Complex when this bit is set or the bits in the PCI Express Device Control register are set. 0x0 - (disable) Disable non-fatal and fatal error reporting if also disabled in Device Control register. 0x1 - (enable) Enable non-fatal and fatal error reporting.
9	FB2B	RO	0x0	**Fast Back-to-Back Enable.** Not applicable.
10	INTXD	RW	0x0	**INTx Disable.** Controls the ability of the function to generate an INTx interrupt message. When this bit is set, INTx interrupts generated by this function are negated. This may result in a change in the resolved interrupt state of the function.
15:11	Reserved	RO	0x0	Reserved field.

Notes

PCISTS - PCI Status (0x006)

Bit Field	Field Name	Type	Default Value	Description
2:0	Reserved	RO	0x0	Reserved field.
3	INTS	RO	0x0	**INTx Status.** This bit is set when an INTx interrupt is pending from the function.
4	CAPL	RO	0x1	**Capabilities List.** This bit is hardwired to one to indicate that this function implements an extended capability list item.
5	C66MHZ	RO	0x0	**66 MHz Capable.** Not applicable.
6	Reserved	RO	0x0	Reserved field.
7	FB2B	RO	0x0	**Fast Back-to-Back (FB2B).** Not applicable.
8	MDPED	RW1C	0x0	**Master Data Parity Error Detected.** This bit is set by the function when the PERRE bit in the PCICMD register is set and one of the following occurs: 1) The function receives a completion marked as 'poisoned'. 2) The function transmits a poisoned request.
10:9	DEVT	RO	0x0	**DEVSEL# TIming.** Not applicable.
11	STAS	RO	0x0	**Signaled Target Abort.** Not applicable since the DMA function never issues completions with completer-abort status.
12	RTAS	RW1C	0x0	**Received Target Abort.** This bit is set when the DMA function receives a completion with Completer Abort completion status. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a completion with Completer Abort completion status is received by this function.
13	RMAS	RW1C	0x0	**Received Master Abort.** This bit is set when the DMA function receives a completion with Unsupported Request completion status. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a completion with Unsupported Request completion status is received by this function.
14	SSE	RW1C	0x0	**Signaled System Error.** This bit is set when the function sends an ERR_FATAL or ERR_NONFATAL message and the SERR Enable (SERRE) bit in the PCICMD register is set. 0x0 - (noerror) no error. 0x1 - (error) This bit is set when a fatal or non-fatal error is signaled.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	DPE	RW1C	0x0	**Detected Parity Error.** This bit is set by the function whenever it receives a poisoned TLP regardless of the state of the PERRE bit in the PCI Command register.

RID - Revision Identification (0x008)

Bit Field	Field Name	Type	Default Value	Description
7:0	RID	RWL	- SWSticky	**Revision ID.** This field contains the revision identification number for the device. See section Revision ID on page 1-1.

CCODE - Class Code (0x009)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTF	RWL	0x00 SWSticky	**Interface.** No standard interface defined.
15:8	SUB	RWL	0x80 SWSticky	**Sub Class Code.** This value indicates that the device is classified as 'other'.
23:16	BASE	RWL	0x08 SWSticky	**Base Class Code.** This value indicates that the device is a generic system peripheral.

CLS - Cache Line Size (0x00C)

Bit Field	Field Name	Type	Default Value	Description
7:0	CLS	RW	0x00	**Cache Line Size.** This field has no effect on the function's operation but may be read and written by software. This field is implemented for compatibility with legacy software.

LTIMER - Latency Timer (0x00D)

Bit Field	Field Name	Type	Default Value	Description
7:0	LTIMER	RO	0x00	**Latency Timer.** Not applicable.

Notes

HDR - Header Type (0x00E)

Bit Field	Field Name	Type	Default Value	Description
7:0	HDR	RO	0x80	**Header Type.** This field indicates the configuration space header type for the DMA function (type 0 header). Since the DMA function always co-exists with another function in the port, this field has a value of 0x80.

BIST - Built-in Self Test Register (0x00F)

Bit Field	Field Name	Type	Default Value	Description
7:0	BIST	RO	0x0	**BIST.** This value indicates that the function does not implement BIST.

BAR0 - Base Address Register 0 (0x010)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RO	0x0	**Memory Space Indicator.** This bit determines if the base address register maps into memory space or I/O space. This bit is always 0x0 since the DMA function does not support I/O space. 0x0 - (memory) memory space. 0x1 - (io) I/O space.
2:1	TYPE	RO	0x0	**Address Type.** When the MEMSI field indicates memory space, this field specifies if a 32-bit or 64-bit address format is used. The value of this field is determined by the TYPE field in the BARSETUP0 register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.
3	PREF	RO	0x0	**Prefetchable.** If the MEMSI field selects memory, this field indicates if the memory is prefetchable. The value of this field is determined by the PREF field in the BARSETUP0 register. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
11:4	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:12	BADDR	RW	0x0	**Base Address.** This field specifies the address bits to be used by the function in decoding and accepting transactions. The BAR aperture for this BAR is always 4 KB (i.e., bits [11:4] in this register are hardwired to 0x0). When the MEMSI indicates memory and the TYPE field indicates 64-bit addressing, the upper bits of the address of the BADDR field are contained in the next consecutive odd numbered BAR (i.e., BAR1). See the PCI and PCI Express specifications for more information.

BAR1 - Base Address Register 1 (0x014)

When the MEMSI field in BARSETUP0 is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, BAR1 takes on the function of the upper 32-bits of the BADDR field in BAR0. Otherwise, BAR1 takes on a value of 0x0 and all bits are read-only.

Bit Field	Field Name	Type	Default Value	Description
31:0	BADDR	See Description	0x0	**Base Address.** When the MEMSI field in the BARSETUP0 register is set to memory space (i.e., zero) and the TYPE field is set to 64-bit addressing, the SIZE field in the BARSETUP0 register controls which bits in this field may be modified. Otherwise, this field is read-only with a default value of 0x0.

BAR2 - Base Address Register 2 (0x018)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Not supported.

BAR3 - Base Address Register 3 (0x01C)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Not supported.

BAR4 - Base Address Register 4 (0x020)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Not supported.

Notes

BAR5 - Base Address Register 5 (0x024)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Not supported.

CCISPTR - CardBus CIS Pointer (0x028)

Bit Field	Field Name	Type	Default Value	Description
31:0	CCISPTR	RO	0x0	**CardBus CIS Pointer.** Not applicable.

SUBVID - Subsystem Vendor ID Pointer (0x02C)

r

Bit Field	Field Name	Type	Default Value	Description
15:0	SUBVID	RWL	0x0 SWSticky	**Subsystem Vendor ID.** This field identifies the vendor of the subsystem. This field must be loaded with the subsystem vendor ID prior to system software accessing PCI configuration space (e.g., via EEPROM). Refer to the PCI 3.0 specification, Section 6.2.4 for further information.

SUBID - Subsystem ID Pointer (0x02E)

Bit Field	Field Name	Type	Default Value	Description
15:0	SUBID	RWL	0x0 SWSticky	**Subsystem ID.** This field identifies the subsystem. This field must be loaded with the subsystem ID prior to system software accessing PCI configuration space (e.g., via EEPROM). Refer to the PCI 3.0 specification, Section 6.2.4 for further information.

EROMBASE - Expansion ROM Base (0x030)

Bit Field	Field Name	Type	Default Value	Description
31:0	EROMBASE	RO	0x0	**Expansion ROM Base Address.** The function does not implement an expansion ROM. Thus, this field is hardwired to zero.

Notes

CAPPTR - Capabilities Pointer (0x034)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPPTR	RWL	0x40 SWSticky	**Capabilities Pointer.** This field specifies a pointer to the head of the capabilities structure.

INTRLINE - Interrupt Line (0x03C)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRLINE	RW	0x0	**Interrupt Line.** This register communicates interrupt line routing information. Values in this register are programmed by system software and are system architecture specific. The function does not use the value in this register.

INTRPIN - Interrupt PIN (0x03D)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRPIN	RWL	0x0 SWSticky	**Interrupt Pin.** The value in this register indicates the INTx message (e.g., INTA, INTB, etc.) used by this function. This field has RWL type to allow system designers to change the INTx messages generated by this function, as shown below. 0x0 - (none) This function does not generate any INTx interrupts. 0x1 - (INTA) This function generates INTA interrupts. 0x2 - (INTB) This function generates INTB interrupts. 0x3 - (INTC) This function generates INTC interrupts. 0x4 - (INTD) This function generates INTD interrupts. Programming this field to 0x0 in effect disables interrupt generation.

MINGNT - Minimum Grant (0x03E)

Bit Field	Field Name	Type	Default Value	Description
7:0	MINGNT	RO	0x0	**Minimum Grant.** Not applicable.

Notes

MAXLAT - Maximum Latency (0x03F)

Bit Field	Field Name	Type	Default Value	Description
7:0	MAXLAT	RO	0x0	**Maximum Latency.** Not applicable.

PCI Express Capability Structure

PCIECAP - PCI Express Capability (0x040)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x10	**Capability ID.** The value of 0x10 identifies this capability as a PCI Express capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section DMA Function Registers on page 19-23 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
19:16	VER	RWL	0x2 SWSticky	**PCI Express Capability Version.** This field indicates the PCI-SIG defined PCI Express capability structure version number.
23:20	TYPE	RO	0x0	**Port Type.** This field indicates that the function is a PCI Express Endpoint function.
24	SLOT	RO	0x0	**Slot Implemented.** Not applicable.
29:25	IMN	RO	0x0	**Interrupt Message Number.** The function is allocated only one MSI. Therefore, this field is set to zero.
31:30	Reserved	RO	0x0	Reserved field.

Notes

PCIEDCAP - PCI Express Device Capabilities (0x044)

Bit Field	Field Name	Type	Default Value	Description
2:0	MPAYLOAD	RWL	HWINIT (See description) MSWSticky	**Maximum Payload Size Supported.** This field indicates the maximum payload size that the device can support for TLPs. The default value of this field is automatically set by the hardware based on the port's maximum link width as determined by the stack's configuration. If a port has a maximum link width of x1, the default value of this field is 0x3. Otherwise, the default value of this field is 0x4. 0x0 - (s128) 128 bytes max payload size 0x1 - (s256) 256 bytes max payload size 0x2 - (s512) 512 bytes max payload size 0x3 - (s1024) 1024 bytes max payload size 0x4 - (s2048) 2048 bytes max payload size 0x5 - Not supported 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
4:3	PFS	RO	0x0	**Phantom Functions Supported.** This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with the TLP's tag identifier. The value is hardwired to 0x0 to indicate that no function number bits are used for phantom functions.
5	ETAG	RWL	0x1 SWSticky	**Extended Tag Field Support.** This field indicates the maximum supported size of the Tag field as a requester. 0x0 -5-bit Tag field supported 0x1 -8-bit Tag field supported
8:6	E0AL	RWL	0x7 SWSticky	**Endpoint L0s Acceptable Latency.** This field indicates the acceptable total latency that this function can withstand due to transition from the L0s state to the L0 state. The value is hardwired to 0x7 indicating that this function places no limit on the L0s to L0 latency.
11:9	E1AL	RWL	0x7 SWSticky	**Endpoint L1 Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L1 state to the L0 state. The value is hardwired to 0x7 indicating that this function places no limit on the L1 to L0 latency.
12	ABP	RO	0x0	**Attention Button Present.** In PCI Express 1.0a when set, this bit indicates that an Attention Button is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
13	AIP	RO	0x0	**Attention Indicator Present.** In PCI Express 1.0a when set, this bit indicates that an Attention Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	PIP	RO	0x0	**Power Indicator Present**. In PCI Express 1.0a when set, this bit indicates that a Power Indicator is implemented on the card/module. The value of this field is undefined in the PCI Express Base Specification Rev. 2.1.
15	RBERR	RO	0x1	**Role Based Error Reporting.** This bit is set to indicate that this function supports role-based error reporting as defined in the PCI Express Base Specification Rev. 2.1.
17:16	Reserved	RO	0x0	Reserved field.
25:18	CSPLV	RO	0x0	**Captured Slot Power Limit Value.** This field in combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. The value of this field is set by a Set_Slot_Power_Limit Message received by the port.[1]
27:26	CSPLS	RO	0x0	**Captured Slot Power Limit Scale.** This field specifies the scale used for the Slot Power Limit Value. The value of this field is set by a Set_Slot_Power_Limit Message received by the port. 0 - (v1) 1.0x 1 - (v1p1) 0.1x 2 - (v0p01) 0.01x 3 - (v0p001x) 0.001x
28	FLR	RO	0x0	**Function Level Reset Capability.** This function does not support function-level-reset. Therefore, this field is hardwired to 0x0.
31:29	Reserved	RO	0x0	Reserved field.

1. NOTE: Set_Slot_Power_Limit messages received by a port implicitly target all functions in the port.

PCIEDCTL - PCI Express Device Control (0x048)

Bit Field	Field Name	Type	Default Value	Description
0	CEREN	RW	0x0	**Correctable Error Reporting Enable.** This bit controls reporting of correctable errors by this function.
1	NFEREN	RW	0x0	**Non-Fatal Error Reporting Enable.** This bit controls reporting of non-fatal errors by this function.
2	FEREN	RW	0x0	**Fatal Error Reporting Enable.** This bit controls reporting of fatal errors by this function.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	URREN	RW	0x0	**Unsupported Request Reporting Enable.** This bit controls reporting of unsupported requests by this function.
4	ERO	RW	0x1	**Enable Relaxed Ordering.** When this bit is set, the DMA function is permitted to set the relaxed-ordering bit in the attributes field of the transactions it initiates (refer to section TLP Attribute and Traffic Class Control on page 15-20). When this bit is cleared, the DMA function does not set the relaxed-ordering bit for the transactions it initiates. This bit overrides the relaxed-ordering controls described in section TLP Attribute and Traffic Class Control on page 15-20.
7:5	MPS	RW	0x0	**Max Payload Size.** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as large as the set value. As a transmitter, the function must not generate TLPs exceeding the set value. This field should be set to a value less than that advertised by the Maximum Payload Size Supported (MPAYLOAD) field in the PCI Express Device Capabilities (PCIEDCAP) register. Setting this field to a value larger than that advertised in the MPAYLOAD field produces undefined results. Programming of this field is subject to the restrictions outlined in section Maximum Payload Size on page 10-2 and section Maximum Payload Size on page 14-22. 0x0 - (s128) 128 bytes max payload size 0x1 - (s256) 256 bytes max payload size 0x2 - (s512) 512 bytes max payload size 0x3 - (s1024) 1024 bytes max payload size 0x4 - (s2048) 2048 bytes max payload size 0x5 - reserved (treated as 128 bytes) 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
8	ETFEN	RW	0x0	**Extended Tag Field Enable.** When this bit is set, Request TLPs generated by the function use an 8-bit tag (i.e., allows up to 256 outstanding requests). Else, Request TLPs generated by the function use a 5-bit tag (i.e., allows up to 32 outstanding requests).
9	PFEN	RO	0x0	**Phantom Function Enable.** This function does not support phantom function numbers. Therefore, this field is hardwired to zero.
10	AUXPMEN	RO	0x0	**Auxiliary Power PM Enable.** The device does not implement this capability.

Notes

Bit Field	Field Name	Type	Default Value	Description
11	ENS	RW	0x1	**Enable No Snoop.** When this bit is set, the DMA function is permitted to set the No Snoop bit in the attributes field of the transactions it initiates (refer to section TLP Attribute and Traffic Class Control on page 15-20). When this bit is cleared, the DMA function does not set the No Snoop bit for the transactions it initiates. This bit overrides the No Snoop controls described in section TLP Attribute and Traffic Class Control on page 15-20.
14:12	MRRS	RW	0x2	**Maximum Read Request Size.** This field sets the maximum read request size for the DMA function as a requester. 0x0 - (s128) 128 bytes max read request size 0x1 - (s256) 256 bytes max read request size 0x2 - (s512) 512 bytes max read request size 0x3 - (s1024) 1024 bytes max read request size 0x4 - (s2048) 2048 bytes max read request size 0x5 - (s4096) 4096 bytes max read request size 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)
15	IFLR	RO	0x0	**Initiate Function Level Reset.** This function does not support function-level-reset. Therefore this field is hardwired to 0x0.

PCIEDSTS - PCI Express Device Status (0x04A)

Bit Field	Field Name	Type	Default Value	Description
0	CED	RW1C	0x0	**Correctable Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
1	NFED	RW1C	0x0	**Non-Fatal Error Detected.** This bit indicates the status of correctable errors detected by this function. Errors are logged in this register regardless of whether error reporting is enabled or not.
2	FED	RW1C	0x0	**Fatal Error Detected.** This bit indicates the status of Fatal errors detected by this function. Errors are logged in this registers regardless of whether error reporting is enabled or not.
3	URD	RW1C	0x0	**Unsupported Request Detected.** This bit indicates that the function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not.
4	AUXPD	RO	0x0	**Aux Power Detected.** Devices that require AUX power, set this bit when AUX power is detected.This device does not require AUX power, hence the value is hardwired to zero.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	TP	RO	0x0	**Transactions Pending.** This bit is set when the DMA function has issued non-posted requests that have not been completed. This bit is cleared when all outstanding non-posted requests have been completed or terminated via the completion timeout mechanism.
15:6	Reserved	RO	0x0	Reserved field.

PCIELCAP - PCI Express Link Capabilities (0x04C)

Bit Field	Field Name	Type	Default Value	Description
3:0	MAXLNKSPD	RWL	0x2 SWSticky	**Maximum Link Speed.** This field indicates the supported link speeds of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved Note: This device advertises support for 5 GT/s regardless of the setting of this field.Modifying this field has no effect on the hardware.
9:4	MAXLNK-WDTH	RWL	HWINIT (See description) MSWSticky	**Maximum Link Width.** This field indicates the maximum link width of the given PCI Express link. This field may be overridden to allow the link width to be forced to a smaller value. When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode). Violating this rule produces undefined results. Setting this field to an invalid or reserved value is allowed, and results in the port operating at it's default value. The default value of this field is automatically set by the hardware as described in section Port Maximum Link Width on page 7-2. 0 - reserved 1 - (x1) x1 link width 2 - (x2) x2 link width 4 - (x4) x4 link width 8 - (x8) x8 link width 12 - (x12) x12 link width 16 - (x16) x16 link width 32- (x32) x32 link width others - reserved Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any stack configuration change.

Notes

Bit Field	Field Name	Type	Default Value	Description
11:10	ASPMS	RWL	0x3 SWSticky	**Active State Power Management (ASPM) Support.** This default value of this field is 0x3 to indicate that L0s and L1 are supported. This field may be overridden to allow user control over the ASPM capabilities of this port (L0s and/or L1). When modifying this field, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
14:12	L0SEL	RWL	0x6 SWSticky	**L0s Exit Latency.** This field indicates the L0s exit latency for the given PCI Express link. Transitioning from L0s to L0 always requires approximately 2.04µs. Thus, default value indicates an L0s exit latency between 2µs and 4µs. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
17:15	L1EL	RWL	0x2 SWSticky	**L1 Exit Latency.** This field indicates the L1 exit latency for the given PCI Express link. Transitioning from L1 to L0 always requires approximately 2.3 µS. Therefore, a value 2 µs to less than 4 µs is reported with a default value of 0x2. If this field is modified, the user must ensure that all functions of the port have identical values in this field (i.e., when the port operates in a multi-function mode).
18	CPM	RWL	0x0 SWSticky	**Clock Power Management.** This bit indicates if the component tolerates removal of the reference clock via the "CLKREQ#" mechanism. The device does not support the removal of reference clocks.
19	SDERR	RO	0x0	**Surprise Down Error Reporting.** Not applicable to upstream ports.
20	DLLLA	RO	0x0	**Data Link Layer Link Active Reporting.** Not applicable to upstream ports.
21	LBN	RO	0x0	**Link Bandwidth Notification Capability.** Not applicable to upstream ports.
23:22	Reserved	RO	0x0	Reserved field.
31:24	PORTNUM	RO	Port 0: 0x0 Port 2: 0x2 Port 4: 0x4 Port 6: 0x6 Port 8: 0x8 Port 12: 0xC Port 16: 0x10 Port 20: 0x14	**Port Number.** This field indicates the PCI Express port number for the corresponding link.

Notes

PCIELCTL - PCI Express Link Control (0x050)

Bit Field	Field Name	Type	Default Value	Description
1:0	ASPM	RW	0x0	**Active State Power Management (ASPM) Control.** This field controls the level of ASPM supported by the link. The initial value corresponds to disabled. 0x0 - (disabled) disabled 0x1 - (l0s) L0s enable entry 0x2 - (l1) L1 enable entry 0x3 - (l0sl1) L0s and L1 enable entry Note that "L0s enable entry" corresponds to the transmitter entering L0s (the receiver supports this function and is not affected by this setting). When a port operates in a multi-function mode, only capabilities enabled in all functions of the port are enabled for the port as a whole (e.g., L0s is enabled for the port when all functions of the port have L0s enabled in this field). It is recommended, though not required, that software program the same value in this field for all functions of the port.
2	Reserved	RO	0x0	Reserved field.
3	RCB	RO	0x0	**Read Completion Boundary.** The DMA function assumes a read-completion-boundary of 64 bytes and does not support re-programming of this field.
4	LDIS	RO	0x0	**Link Disable.** Not applicable.
5	LRET	RO	0x0	**Link Retrain.** Not applicable.
6	CCLK	RW	0x0	**Common Clock Configuration.** When set, this bit indicates that this port and the port at the opposite end of the link are operating with a distributed common reference clock. When a port operates in a multi-function mode, software must set this bit identically for all functions of the port. Otherwise, the port assumes that it is not operating with a distributed common reference clock. After modifying this bit in both components of the link, software must trigger a link retrain by setting the link retrain bit in the upstream component's Link Control register. In the switch, the L0s and L1 exit latencies do not change among common and non-common clock configurations.
7	ESYNC	RW	0x0	**Extended Sync.** When set this bit forces transmission of additional ordered sets when exiting the L0s state and when in the recovery state. When a port operates in a multi-function mode, the effect of this bit is applied when this bit is set in any of the port's functions.
8	CLKPWRMGT	RO	0x0	**Enable Clock Power Management.** The device does not support this feature.

Notes

Bit Field	Field Name	Type	Default Value	Description
9	HAWD	RO	0x0	**Hardware Autonomous Width Disable.** Not applicable.
10	LBWINTEN	RO	0x0	**Link Bandwidth Management Interrupt Enable.** Not applicable.
11	LABWINTEN	RO	0x0	**Link Autonomous Bandwidth Interrupt Enable.** Not applicable.
15:12	Reserved	RO	0x0	Reserved field.

PCIELSTS - PCI Express Link Status (0x052)

Bit Field	Field Name	Type	Default Value	Description
3:0	CLS	RO	0x1	**Current Link Speed.** This field indicates the current link speed of the port. 1 - (gen1) 2.5 GT/s 2 - (gen2) 5 GT/s others - reserved
9:4	NLW	RO	HWINIT	**Negotiated Link Width.** This field indicates the negotiated width of the link. 00 0001b - x1 00 0010b - x2 00 0100b - x4 00 1000b - x8 00 1100b - x12 01 0000b - x16 10 0000b - x32 When the MAXLNKWDTH field in the PCIELCAP register selects a width not supported by the port, the value of this field corresponds to the setting of the MAXLNKWDTH field, regardless of the actual negotiated link width. When the MAXLNKWDTH field in the PCIELCAP register selects a width supported by the port, but the link is unable to train, the value in this field is set to 0x0. When the port operates in a multi-function mode, the above rules are based on the MAXLNKWDTH field for function 0 of the port. Note that software must ensure that all functions of the port have identical MAXLNKWDTH field values.
10	Reserved	RO	0x0	Reserved field.
11	LTRAIN	RO	0x0	**Link Training.** Not applicable.
12	SCLK	RWL	HWINIT SWSticky	**Slot Clock Configuration.** When set, this bit indicates that the port uses the same physical reference clock used by its link partner (i.e., common-clock configuration). The initial value of this field depends on the port's clocking mode. Refer to Table 2.4 for further details. When the port operates in a multi-function mode, this field reports the same value for all functions of the port.

Notes

Bit Field	Field Name	Type	Default Value	Description
13	DLLLA	RO	0x0	**Data Link Layer Link Active.** Not applicable.
14	LBWSTS	RO	0x0	**Link Bandwidth Management Status.** Not applicable.
15	LABWSTS	RO	0x0	**Link Autonomous Bandwidth Status.** Not applicable.

PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTRS	RWL	0xF	**Completion Timeout Ranges Supported.** The DMA function supports completion timeout ranges A, B, C, D as described below. Range A: 50 µs to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4 s Range D: 4 s to 64 s
4	CTDS	RWL	0x1	**Completion Timeout Disable Supported.** The DMA function supports completion timeout disabling. Note: In the device, completion timeout disabling is strongly discouraged, as it can result in DMA malfunction in cases in which outstanding DMA requests are not fully completed (e.g., due to an ECRC error in a completion TLP).
5	ARIFS	RO	0x0	**ARI Forwarding Supported.** Not applicable.
6	ATOPRS	RO	0x0	**AtomicOp Routing Supported.** Not applicable.
7	ATOPC32S	RO	0x0	**32-bit AtomicOp Completer Supported.** Not supported.
8	ATOPC64S	RO	0x0	**64-bit AtomicOp Completer Supported.** Not supported.
9	CASC128S	RO	0x0	**128-bit CAS Completer Supported.** Not supported.
10	NROEP	RO	0x1	**No RO-enabled PR-PR Passing.** Not applicable.
11	LTRMS	RO	0x0	**LTR Mechanism Supported.** The switch does not support the Latency Tolerance Reporting mechanism.
13:12	TPHCS	RO	0x0	**TPH Completer Supported.** Not supported.
19:14	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
20	EFMTFS	RO	0x0	**Extended Fmt Field Supported.** The switch does not support the 3-bit definition of the FMT field in TLPs.
21	E2ETPS	RO	0x0	**End-to-End TLP Prefix Supported.** The switch does not support End-to-End TLP Prefixes.
31:22	Reserved	RO	0x0	Reserved field.

PCIEDCTL2 - PCI Express Device Control 2 (0x068)

Bit Field	Field Name	Type	Default Value	Description
3:0	CTV	RW	0x0	**Completion Timeout Value.** This field selects the completion timeout value used by the DMA function. nction. 0x0 - 50 µs 0x1 - 100 µs 0x2 - 10 ms 0x5 - 55 ms 0x6 - 210 ms 0x9 - 900 ms 0xA - 3.5 s 0xD - 13 s 0xE - 34 s others - reserved Software is permitted to change the value in this field at any time. For requests already pending when the completion timeout value is changed, hardware is permitted to use either the new or the old value for the outstanding requests, and is permitted to base the start time for each request either on when this value was changed or on when each request was issued.
4	CTD	RW	0x0	**Completion Timeout Disable.** When this bit is set, completion timeout checking is disabled in the DMA function. Note: In the switch, completion timeout disabling is strongly discouraged, as it can result in DMA malfunction in cases in which outstanding DMA requests are not fully completed (e.g., due to an ECRC error in a completion TLP).
5	ARIFEN	RW	0x0	**ARI Forwarding Enable.** Not applicable.
6	ATOPRE	RO	0x0	**AtomicOp Requester Enable.** Not supported.
7	ATOPEB	RO	0x0	**AtomicOp Egress Blocking.** Not applicable.
8	IDORE	RO	0x0	**IDO Request Enable.** Not supported.

Notes

Bit Field	Field Name	Type	Default Value	Description
9	IDOCE	RO	0x0	**IDO Completion Enable.** Not supported.
10	LTRME	RO	0x0	**LTR Mechanism Enable.** Not supported.
14:11	Reserved	RO	0x0	Reserved field.
15	E2ETLPPB	RO	0x0	**End-to-End TLP Prefix Blocking.** Not supported.

PCIEDSTS2 - PCI Express Device Status 2 (0x06A)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

PCIELCTL2 - PCI Express Link Control 2 (0x070)

Bit Field	Field Name	Type	Default Value	Description
3:0	TLS	RO	0x0	**Target Link Speed.** Not applicable (function 0 of the port controls this functionality).
4	ECOMP	RO	0x0	**Enter Compliance.** Not applicable (function 0 of the port controls this functionality).
5	HASD	RO	0x0	**Hardware Autonomous Speed Disable.** Not applicable (function 0 of the port controls this functionality).
6	SDE	RO	0x0	**Selectable De-emphasis.** Not applicable (function 0 of the port controls this functionality).
9:7	TM	RO	0x0	**Transmit Margin.** Not applicable (function 0 of the port controls this functionality).
10	EMC	RO	0x0	**Enter Modified Compliance.** Not applicable (function 0 of the port controls this functionality).

Notes

Bit Field	Field Name	Type	Default Value	Description
11	CSOS	RO	0x0	**Compliance SOS.** Not applicable (function 0 of the port controls this functionality).
12	CDE	RO	0x0	**Compliance De-emphasis.** Not applicable (function 0 of the port controls this functionality).
15:13	Reserved	RO	0x0	Reserved field.

PCIELSTS2 - PCI Express Link Status 2 (0x072)

Bit Field	Field Name	Type	Default Value	Description
0	CDE	RO	0x0	**Current De-emphasis.** The value of this bit indicates the current de-emphasis level when the link operates in 5.0 GT/s. 0x0 - De-emphasis level = -6.0 dB 0x1 - De-emphasis level = -3.5 dB The value of this bit in undefined when the link operates at 2.5 GT/s.
15:1	Reserved	RO	0x0	Reserved field.

PCI Power Management Capability Structure

PMCAP - PCI Power Management Capabilities (0x0C0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 identifies this capability as a PCI power management capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section DMA Function Registers on page 19-23 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
18:16	VER	RO	0x3	**Power Management Capability Version.** Complies with version the PCI Bus Power Management Interface Specification, Revision 1.2.
19	PMECLK	RO	0x0	**PME Clock.** Does not apply to PCI Express.

Notes

Bit Field	Field Name	Type	Default Value	Description
20	Reserved	RO	0x0	Reserved field.
21	DEVSP	RWL	0x0 SWSticky	**Device Specific Initialization.** The value of zero indicates that no device specific initialization is required.
24:22	AUXI	RO	0x0	**AUX Current.** The switch does not use auxiliary current.
25	D1	RO	0x0	**D1 Support.** This field indicates that this function does not support D1.
26	D2	RO	0x0	**D2 Support.** This field indicates that this function does not support D2.
31:27	PME	RO	0x0	**PME Support.** This function does not generate PME in any power state.

PMCSR - PCI Power Management Control and Status (0x0C4)

Bit Field	Field Name	Type	Default Value	Description
1:0	PSTATE	RW	0x0	**Power State.** This field is used to determine the current power state of the function and to set a new power state. 0x0 - (d0) D0 state 0x1 - (d1) D1 state (not supported by the switch and reserved) 0x2 - (d2) D2 state (not supported by the switch and reserved) 0x3 - (d3) $D3_{hot}$ state
2	Reserved	RO	0x0	Reserved field.
3	NOSOFTRST	RWL	0x1 SWSticky	**No Soft Reset.** This bit indicates if the configuration context is preserved by the function when the device transitions from a D3hot to D0 power management state. 0x0 - (reset) State reset 0x1 - (preserved) State preserved
7:4	Reserved	RO	0x0	Reserved field.
8	PMEE	RO	0x0	**PME Enable.** Not applicable since this function does not support generation of PME events.
12:9	DSEL	RO	0x0	**Data Select.** The optional data register is not implemented.
14:13	DSCALE	RO	0x0	**Data Scale.** The optional data register is not implemented.
15	PMES	RW1C	0x0 Sticky	**PME Status.** Since this function never generates a PME, this bit will never be set.

Notes

Bit Field	Field Name	Type	Default Value	Description
21:16	Reserved	RO	0x0	Reserved field.
22	B2B3	RO	0x0	**B2/B3 Support.** Does not apply to PCI Express.
23	BPCCE	RO	0x0	**Bus Power/Clock Control Enable.** Does not apply to PCI Express.
31:24	DATA	RO	0x0	**Data.** This optional field is not implemented.

Message Signaled Interrupt Capability Structure

MSICAP - Message Signaled Interrupt Capability and Control (0x0D0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x5	**Capability ID.** The value of 0x5 identifies this capability as a MSI capability structure.
15:8	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section DMA Function Registers on page 19-23 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.
16	EN	RW	0x0	**Enable.** This bit enables MSI. 0x0 -(disable) disabled 0x1 -(enable) enabled
19:17	MMC	RO	0x0	**Multiple Message Capable.** This field contains the number of requested messages.
22:20	MME	RW	0x0	**Multiple Message Enable.** Hardwired to one message.
23	A64	RO	0x1	**64-bit Address Capable.** The function is capable of generating messages using a 64-bit address.
31:24	Reserved	RO	0x0	Reserved field.

Notes

MSIADDR - Message Signaled Interrupt Address (0x0D4)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	ADDR	RW	0x0	**Message Address.** This field specifies the lower portion of the DWORD address of the MSI memory write transaction. Refer to section Interrupts on page 15-24 for restrictions on the programming of this field.

MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8)

Bit Field	Field Name	Type	Default Value	Description
31:0	UADDR	RW	0x0	**Upper Message Address.** This field specifies the upper portion of the DWORD address of the MSI memory write transaction. If the contents of this field are non-zero, then 64-bit address is used in the MSI memory write transaction. If the contents of this field are zero, then the 32-bit address specified in the MSIADDR register is used. Refer to section Interrupts on page 15-24 for restrictions on the programming of this field.

MSIMDATA - Message Signaled Interrupt Message Data (0x0DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	MDATA	RW	0x0	**Message Data.** This field contains the lower 16-bits of data that are written when a MSI is signaled.
31:16	Reserved	RO	0x0	Reserved field.

Extended Configuration Space Access Registers

ECFGADDR - Extended Configuration Space Access Address (0x0F8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
7:2	REG	RW	0x0	**Register Number.** This field selects the configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification Rev. 2.1. The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). Violation of this rule produces undefined results. Also, the value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). Violation of this rule produces undefined results.
11:8	EREG	RW	0x0	**Extended Register Number.** This field selects the extended configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification Rev. 2.1. The value of this register must not be programmed to point to the address offset of this register (i.e., 0xF8) or the ECFGDATA register (i.e., 0xFC). Violation of this rule produces undefined results. Also, the value of this register must not be programmed to point to the global address space access registers (GASAADDR and GASADATA). Violation of this rule produces undefined results.
31:12	Reserved	RO	0x0	Reserved field.

ECFGDATA - Extended Configuration Space Access Data (0x0FC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Configuration Data.** A read from this field will return the configuration space register value pointed to by the ECFGADDR register. A write to this field will update the contents of the configuration space register pointed to by the ECFGADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

Notes

Advanced Error Reporting (AER) Extended Capability

AERCAP - AER Capabilities (0x100)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x1	**Capability ID.** The value of 0x1 indicates an Advanced Error Reporting capability structure.
19:16	CAPVER	RO	0x2	**Capability Version.** The value of 0x2 indicates compatibility with the PCI Express Base 2.1 Specification.
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section DMA Function Registers on page 19-23 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

AERUES - AER Uncorrectable Error Status (0x104)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW1C	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW1C	0x0 Sticky	**Data Link Protocol Error Status.** This bit is set when a data link layer protocol error is detected.
5	SDOENERR	RO	0x0	**Surprise Down Error Status.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW1C	0x0 Sticky	**Poisoned TLP Status.** This bit is set when a poisoned TLP is detected.
13	FCPERR	RO	0x0	**Flow Control Protocol Error Status.** Not applicable (the switch does not support flow control protocol error checking).
14	COMPTO	RW1C	0x0 Sticky	**Completion Timeout Status.** This bit is set when a completion timeout error is detected.
15	CABORT	RO	0x0	**Completer Abort Status.** Not applicable (this bit is never set as this function never responds to a non-posted request with a completer abort).
16	UECOMP	RW1C	0x0 Sticky	**Unexpected Completion Status.** This bit is set when an unexpected completion is detected.

Notes

Bit Field	Field Name	Type	Default Value	Description
17	RCVOVR	RW1C	0x0 Sticky	**Receiver Overflow Status.** This bit is set when a receiver overflow is detected.
18	MALFORMED	RW1C	0x0 Sticky	**Malformed TLP Status.** This bit is set when a malformed TLP is detected.
19	ECRC	RW1C	0x0 Sticky	**ECRC Status.** This bit is set when an ECRC error is detected.
20	UR	RW1C	0x0 Sticky	**UR Status.** This bit is set when an unsupported request is detected.
21	ACSV	RW1C	0x0 Sticky	**ACS Violation Status.** This bit is set when an ACS violation is detected by this function.
22	UIE	RW1C	0x0 Sticky	**Uncorrectable Internal Error Status.** This bit is set when an uncorrectable internal error associated with the this function is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RO	0x0	**MC Blocked TLP Status.** Not applicable (the DMA function does not have a multicast capability structure).
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Status.** Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Status.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERUEM - AER Uncorrectable Error Mask (0x108)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	DLPERR	RW	0x0 Sticky	**Data Link Protocol Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
5	SDOENERR	RO	0x0	**Surprise Down Error Mask.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
13	FCPERR	RO	0x0	**Flow Control Protocol Error Mask.** Not applicable.
14	COMPTO	RW	0x0	**Completion Timeout Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
15	CABORT	RO	0x0	**Completer Abort Mask.** Not applicable.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
17	RCVOVR	RW	0x0 Sticky	**Receiver Overflow Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
18	MALFORMED	RW	0x0 Sticky	**Malformed TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
19	ECRC	RW	0x0 Sticky	**ECRC Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
20	UR	RW	0x0 Sticky	**UR Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.
21	ACSV	RW	0x0 Sticky	**ACS Violation Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the AER Header Log registers, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
22	UIE	RW	0x0 Sticky	**Uncorrectable Internal Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure, the First Error Pointer field (FEPTR) in the AERCTL register is not updated, and an error is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERUES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RO	0x0	**MC Blocked TLP Mask.** Not applicable.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Status.** Not applicable.
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Status.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERUESV - AER Uncorrectable Error Severity (0x10C)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x1 Sticky	**Data Link Protocol Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
5	SDOENERR	RO	0x1	**Surprise Down Error Severity.** Not applicable.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	**Poisoned TLP Status Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
13	FCPERR	RO	0x1	**Flow Control Protocol Error Severity.** Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	COMPTO	RW	0x0 Sticky	**Completion Timeout Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
15	CABORT	RO	0x0	**Completer Abort Severity.** Not applicable.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
17	RCVOVR	RW	0x1 Sticky	**Receiver Overflow Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
18	MALFORMED	RW	0x1 Sticky	**Malformed TLP Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
19	ECRC	RW	0x0 Sticky	**ECRC Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
20	UR	RW	0x0 Sticky	**UR Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
21	ACSV	RW	0x0 Sticky	**ACS Violation Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error.
22	UIE	RW	0x0 Sticky	**Uncorrectable Internal Error Severity.** This bit controls the severity of the reported error. If this bit is set, the event is reported as a fatal error. When this bit is cleared, the event is reported as an uncorrectable error. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of one. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
23	MCBLKTLP	RO	0x0	**MC Blocked TLP Severity.** Not applicable.
24	ATOPEB	RO	0x0	**AtomicOp Egress Blocked Status.** Not applicable.

Notes

Bit Field	Field Name	Type	Default Value	Description
25	TLPPBE	RO	0x0	**TLP Prefix Blocked Error Status.** Not applicable.
31:26	Reserved	RO	0x0	Reserved field.

AERCES - AER Correctable Error Status (0x110)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW1C	0x0 Sticky	**Receiver Error Status.** This bit is set when the Physical Layer detects a receiver error.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW1C	0x0 Sticky	**Bad TLP Status.** This bit is set when a bad TLP is detected.
7	BADDLLP	RW1C	0x0 Sticky	**Bad DLLP Status.** This bit is set when a bad DLLP is detected.
8	RPLYROVR	RW1C	0x0 Sticky	**Replay Number Rollover Status.** This bit is set when a replay number rollover has occurred indicating that the data link layer has abandoned replays and has requested that the link be retrained.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW1C	0x0 Sticky	**Replay Timer timeout Status.** This bit is set when the replay timer in the data link layer times out.
13	ADVISORYNF	RW1C	0x0 Sticky	**Advisory Non-Fatal Error Status.** This bit is set when an advisory non-fatal error is detected as described in Section 6.2.3.2.4 of the PCI Express Base Specification Rev 2.1.
14	CIE	RW1C	0x0 Sticky	**Correctable Internal Error Status.** This bit is set whenever an correctable internal error associated with the port is detected. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW1C	0x0 Sticky	**Header Log Overflow Status.** This bit is set when an error that requires packet-header logging occurs but the packet header cannot be logged by the function's AER Header Log registers (AERHL[1:4]DW). A packet's header cannot be logged in the AER Header Log registers when an error occurs while the First Error Pointer (FEPTR field in the AERCTL register) is valid. The First Error Pointer is valid when it points to a set bit in the AERUES register (i.e., indicating the occurrence of a prior uncorrectable error which has not been cleared by soft-ware). When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IER-RORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Regis-ters in the PCI-to-PCI Bridge Function on page 19-11 for details.
31:16	Reserved	RO	0x0	Reserved field.

AERCEM - AER Correctable Error Mask (0x114)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 Sticky	**Receiver Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 Sticky	**Bad TLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
7	BADDLLP	RW	0x0 Sticky	**Bad DLLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	RPLYROVR	RW	0x0 Sticky	**Replay Number Rollover Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 Sticky	**Replay Timer timeout Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
13	ADVISORYNF	RW	0x1 Sticky	**Advisory Non-Fatal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register.
14	CIE	RW	0x0	**Correctable Internal Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	HLO	RW	0x0 Sticky	**Header Log Overflow Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex. This bit does not affect the state of the corresponding bit in the AERCES register. When the Internal Error Reporting Enable (IERROREN) bit is cleared in the Internal Error Reporting Control (IERRORCTL) register, this field becomes read-only with a value of zero. The IERRORCTL register is a proprietary register located in the configuration space of the port's PCI-to-PCI bridge function. Refer to section Proprietary Port-Specific Registers in the PCI-to-PCI Bridge Function on page 19-11 for details.
31:16	Reserved	RO	0x0	Reserved field.

AERCTL - AER Control (0x118)

Bit Field	Field Name	Type	Default Value	Description
4:0	FEPTR	RO	0x0 Sticky	**First Error Pointer.** This field contains a pointer to the bit in the AERUES register that resulted in the first reported error. This field is valid only when it points to a set bit in the AERUES register.
5	ECRCGC	RWL	0x1 SWSticky	**ECRC Generation Capable.** This bit indicates if the function is capable of generating ECRC.
6	ECRCGE	RW	0x0 Sticky	**ECRC Generation Enable.** When this bit is set, ECRC generation is enabled for the function.
7	ECRCCC	RWL	0x1 SWSticky	**ECRC Check Capable.** This bit indicates if the function is capable of checking ECRC.
8	ECRCCE	RW	0x0 Sticky	**ECRC Check Enable.** When this bit is set, ECRC checking is enabled for the function.
9	MHRC	RO	0x0	**Multiple Header Recording Capable.** Switch ports do not support recording of multiple packet headers.
10	MHRE	RO	0x0	**Multiple Header Recording Enable.** Switch ports do not support recording of multiple packet headers. As a result, this bit is hardwired to 0x0.
31:11	Reserved	RO	0x0	Reserved field.

Notes

AERHL1DW - AER Header Log 1st Doubleword (0x11C)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 1st doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL2DW - AER Header Log 2nd Doubleword (0x120)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 2nd doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL3DW - AER Header Log 3rd Doubleword (0x124)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 3rd doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL4DW - AER Header Log 4th Doubleword (0x128)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RWL	0x0 Sticky	**Header Log.** This field contains the 4th doubleword of the TLP header that resulted in the first reported uncorrectable error.

ACS Extended Capability

ACSECAPH - ACS Extended Capability Header (0x320)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0xD	**Capability ID.** The value of 0xD indicates an ACS extended capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with the PCI Express Base Specification.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:20	NXTPTR	RWL	HWINIT (See description) MSWSticky	**Next Pointer.** This field contains a pointer to the next capability structure. The default value of this register depends on the port's operating mode. See section DMA Function Registers on page 19-23 for details. Note that this field is MSWSticky. Therefore, if this field is modified by software, its value will be preserved regardless of any port operating mode change.

ACSCAP - ACS Capability (0x324)

Bit Field	Field Name	Type	Default Value	Description
0	V	RO	0x0	**ACS Source Validation.** Not applicable to multi-function upstream ports.
1	B	RO	0x0	**ACS Translation Blocking.** Not applicable to multi-function upstream ports.
2	R	RWL	0x1 SWSticky	**ACS P2P Request Redirect.** If set, indicates the port implements ACS Peer-to-Peer Request Redirect among its functions. Refer to section Access Control Services (ACS) Support on page 15-25 for details.
3	C	RWL	0x1 SWSticky	**ACS P2P Completion Redirect.** If set, indicates the port implements ACS Peer-to-Peer Completion Redirect among its functions. Refer to section Access Control Services (ACS) Support on page 15-25 for details.
4	U	RO	0x0	**ACS Upstream Forwarding.** Not applicable to multi-function upstream ports.
5	E	RO	0x0	**ACS P2P Egress Control.** The device does not support ACS P2P Egress Control among functions in a multi-function upstream port.
6	T	RO	0x0	**ACS Direct Translated P2P.** If set, indicates this function implements ACS Direct Translated Peer-to-Peer. The DMA function does not support ACS Direct Translated P2P.
15:7	Reserved	RO	0x0	Reserved field.

ACSCTL - ACS Control (0x326)

Bit Field	Field Name	Type	Default Value	Description
0	V	RO	0x0	**ACS Source Validation Enable.** Not applicable to multi-function upstream ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
1	B	RO	0x0	**ACS Translation Blocking Enable.** Not applicable to multi-function upstream ports.
2	R	RW	0x0	**ACS P2P Request Redirect Enable.** When set, this function performs ACS Peer-to-Peer Request Redirect for function-to-function transfers. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
3	C	RW	0x0	**ACS P2P Completion Redirect Enable.** When set, this function performs ACS Peer-to-Peer Completion Redirect for function-to-function transfers. Note: This field becomes read-only-zero when the corresponding bit in the ACSCAP register is cleared.
4	U	RO	0x0	**ACS Upstream Forwarding Enable.** Not applicable to multi-function upstream ports.
5	E	RO	0x0	**ACS P2P Egress Control Enable.** The device does not support ACS P2P Egress Control among functions in a multi-function upstream port.
6	T	RO	0x0	**ACS Direct Translated P2P Enable.** Not supported by the DMA function.
15:7	Reserved	RO	0x0	Reserved field.

DMA Registers

BAR Configuration

BARSETUP0 - BAR 0 Setup (0x400)

Bit Field	Field Name	Type	Default Value	Description
0	MEMSI	RW	0x0 SWSticky	**MEMSI Select.** This field determines the MEMSI type returned in the MEMSI field of the corresponding BAR. 0x0 - (memory) memory space 0x1 - reserved
2:1	TYPE	RW	0x0 SWSticky	**Address Select.** This field determines the value reported in the TYPE field of the corresponding BAR and selects the address space decoding used when memory space is selected in the MEMSI field in this register. 0x0 - (addr32) 32-bit addressing. Located in lower 4 GB address space. 0x1 - (reserved) reserved. 0x2 - (addr64) 64-bit addressing. 0x3 - (reserved) reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	PREF	RW	0x0 SWSticky	**Prefetchable Select.** This field determines the value reported in the PREF field of the corresponding BAR. 0x0 - (nonprefetch) non-prefetchable. 0x1 - (prefetch) prefetchable.
30:4	Reserved	RO	0x0	Reserved field.
31	EN	RW	0x0 SWSticky	**BAR Enable.** When cleared, the corresponding BAR is disabled and returns a zero when read (i.e., configuration values in this register are ignored and all fields of the BAR take on a value of zero). 0x0 - (disabled) disabled. 0x1 - (enabled) enabled.

DMA AER Error Emulation

DMAUEEM - DMA Uncorrectable Error Emulation (0x408)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x0 SWSticky	**Data Link Protocol Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
11:5	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 SWSticky	**Poisoned TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
15:13	Reserved	RO	0x0	Reserved field.
16	UECOMP	RW	0x0 SWSticky	**Unexpected Completion Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
17	RCVOVR	RW	0x0 SWSticky	**Receiver Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
18	MALFORMED	RW	0x0 SWSticky	**Malformed TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
19	ECRC	RW	0x0 SWSticky	**ECRC Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
20	UR	RW	0x0 SWSticky	**UR Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
21	Reserved	RO	0x0	Reserved field.
22	UIE	RW	0x0 SWSticky	**Uncorrectable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERUES register. This bit always returns 0x0 when read.
30:23	Reserved	RO	0x0	Reserved field.
31	ADVISORYNF	RW	0x0 SWSticky	**Advisory Non-Fatal Error Trigger.** If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible (refer to the PCI Express Base Specification), an advisory non-fatal error is logged an reported in the PCI-to-PCI bridge function's AER capability structure, provided the error severity for the selected uncorrectable error is configured such that the error will be of type non-fatal. If this bit is set together with another error bit in this register for which an advisory non-fatal error is not possible, the operation is undefined. If this bit is set together with another error bit in this register for which an advisory non-fatal error is possible, but the severity of the selected uncorrectable error is fatal, then this bit is ignored and the selected error is logged and reported as a fatal error.

DMACEEM - DMA Correctable Error Emulation (0x40C)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 SWSticky	**Receiver Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 SWSticky	**Bad TLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
7	BADDLLP	RW	0x0 SWSticky	**Bad DLLP Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	RPLYROVR	RW	0x0 SWSticky	**Replay Number Rollover Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 SWSticky	**Replay Timer Timeout Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
13	Reserved	RO	0x0	Reserved field.
14	CIE	RW	0x0 SWSticky	**Correctable Internal Error Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
15	HLO	RW	0x0 SWSticky	**Header Log Overflow Trigger.** Writing a one to this bit causes the corresponding error bit to get set in the PCI-to-PCI Bridge function's AERCES register. This bit always returns 0x0 when read.
31:16	Reserved	RO	0x0	Reserved field.

Internal Error Reporting Masks

DMAIERRORMSK0 - Internal Error Reporting Mask 0 (0x410)

Bit Field	Field Name	Type	Default Value	Description
0	IFBPTLPTO	RW	0x0 SWSticky	**IFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	IFBNPTLPTO	RW	0x0 SWSticky	**IFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
2	IFBCPTLPTO	RW	0x0 SWSticky	**IFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	EFBPTLPTO	RW	0x0 SWSticky	**EFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	EFBNPTLPTO	RW	0x0 SWSticky	**EFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
6	EFBCPTLPTO	RW	0x0 SWSticky	**EFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
7	IFBDATSBE	RW	0x0 SWSticky	**IFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
8	IFBDATDBE	RW	0x0 SWSticky	**IFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
9	IFBCTLSBE	RW	0x0 SWSticky	**IFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
10	IFBCTLDBE	RW	0x0 SWSticky	**IFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11	EFBDATSBE	RW	0x0 SWSticky	**EFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
12	EFBDATDBE	RW	0x0 SWSticky	**EFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
13	EFBCTLSBE	RW	0x0 SWSticky	**EFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
14	EFBCTLDBE	RW	0x0 SWSticky	**EFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
15	E2EPE	RW	0x0 SWSticky	**End-to-End Data Path Parity Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
16	ULD	RW	0x0 SWSticky	**Unreliable Link Detected.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
17	RBCTLSBE	RW	0x0 SWSticky	**Replay Buffer Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
18	RBCTLDBE	RW	0x0 SWSticky	**Replay Buffer Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19	DIFBPTLPTO	RW	0x0 SWSticky	**DMA IFB Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
20	DIFBNPTLPTO	RW	0x0 SWSticky	**DMA IFB Non-Posted TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
21	DIFBCPTLPTO	RW	0x0 SWSticky	**DMA IFB Completion TLP Time-Out.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
22	Reserved	RO	0x0	Reserved field.
23	DIFBDATSBE	RW	0x0 SWSticky	**DMA IFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
24	DIFBDATDBE	RW	0x0 SWSticky	**DMA IFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
25	DIFBCTLSBE	RW	0x0 SWSticky	**DMA IFB Control Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
26	DIFBCTLDBE	RW	0x0 SWSticky	**DMA IFB Control Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
27	DEFBDATSBE	RW	0x0 SWSticky	**DMA EFB Data Single Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
28	DEFBDATDBE	RW	0x0 SWSticky	**DMA EFB Data Double Bit Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

Notes

Bit Field	Field Name	Type	Default Value	Description
30:29	Reserved	RW	0x0 SWSticky	This field is reserved but remains read-write in the hardware. Modifying this field has no effect other than changing the value of the field.
31	DE2EPE	RW	0x0 SWSticky	**DMA End-to-End Data Path Parity Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.

DMAIERRORMSK1 - Internal Error Reporting Mask 1 (0x414)

Bit Field	Field Name	Type	Default Value	Description
0	P0AER	RW	0x1 SWSticky	**Port 0 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
1	Reserved	RO	0x0	Reserved field.
2	P2AER	RW	0x1 SWSticky	**Port 2 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
3	Reserved	RO	0x0	Reserved field.
4	P4AER	RW	0x1 SWSticky	**Port 4 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
5	Reserved	RO	0x0	Reserved field.
6	P6AER	RW	0x1 SWSticky	**Port 6 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
7	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	P8AER	RW	0x1 SWSticky	**Port 8 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
11:9	Reserved	RO	0x0	Reserved field.
12	P12AER	RW	0x1 SWSticky	**Port 12 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
15:13	Reserved	RO	0x0	Reserved field.
16	P16AER	RW	0x1 SWSticky	**Port 16 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
19:17	Reserved	RO	0x0	Reserved field.
20	P20AER	RW	0x1 SWSticky	**Port 20 AER Error.** When this bit is set, the corresponding error bit in the IERRORSTS0/1 register is masked from reporting an internal error to the AER Capability Structure of the DMA function. This bit does not affect the state of the corresponding bit in the IERRORSTS0/1 register.
31:21	Reserved	RO	0x0	Reserved field.

Notes

DMA Multicast Control

MCRCVINT - Multicast Receive Interpretation (0x4FC)

Bit Field	Field Name	Type	Default Value	Description
0	MCRCVINT	RW	0x0 SWSticky	**Multicast Receive Interpretation.** This bit controls whether multicast TLPs emitted by the DMA (i.e., posted TLPs whose address falls within a multicast BAR aperture in the upstream port's PCI-to-PCI bridge or NT functions) are transmitted on the upstream port's link. When this bit is zero, a multicast TLP emitted by the DMA is always transmitted on the upstream port's link. In addition, if the upstream port has a PCI-to-PCI bridge function and the TLP falls within this function's multicast BAR aperture, the TLP is subject to transparent multicast handling (see section Transparent Multicast Operation on page 17-1). Also, if the upstream port has an NT function and the TLP falls within this function's multicast BAR aperture, the TLP is subject to NT Multicast handling (see section Non-Transparent Multicast Operation on page 17-6). When this bit is one, a multicast TLP emitted by the DMA is transmitted on the upstream port's link only when the multicast receive vector bit corresponding to the TLP's multicast group is set in the upstream port's PCI-to-PCI bridge function. Refer to section Multicast TLP Routing on page 17-5 for details on the multicast receive vector. If the upstream port is configured without a PCI-to-PCI bridge function (i.e., NT function with DMA mode) and this bit is set, the user must configure the multicast receive vector in the PCI-to-PCI bridge function by accessing the Multicast Receive Low (MCRCVL) and Multicast Receive High (MCRCVH) registers via the switch's global address space. Note that this bit has no effect on non-multicast TLPs. Such TLPs are logically routed to the upstream port's function that claims the TLP (i.e., PCI-to-PCI bridge or NT). If no upstream port function claims the TLP, such TLPs are routed to the upstream port's link. Refer to section DMA Multicast on page 15-23 for further information on DMA multicast support.
31:1	Reserved	RO	0x0	Reserved field

Notes

DMA Channel Registers

DMAC[1:0]CTL - DMA Channel Control (0x500/600)

Bit Field	Field Name	Type	Default Value	Description
0	RUN	RW	0x0	**Run.** Writing a one into this bit position initiates DMA descriptor processing if the DMA channel is idle and the E bit in the DMACxSTS register is cleared. In addition, under certain circumstances this bit is automatically set to a one when a DMA operation is initiated as a side effect of other events (e.g., when a value is written to the DMAxDPTRL register). Writing a one into this bit position while the DMA channel is suspended, resumes DMA channel operation if the E bit in the DMACxSTS register is cleared. Writing a one into this bit position while the DMA channel is running (i.e., the bit is already a one) can be used to perform dynamic appending of descriptor lists (refer to section Dynamic Appending of Descriptor Lists on page 15-19). Writing a zero to this bit position has no effect on the operation of the DMA channel. This bit is automatically cleared when DMA descriptor processing halts or is aborted. In the case that software sets this bit in the same clock cycle that it is cleared by hardware, the bit is set (i.e., software is given priority).
1	ABORT	RW	0x0	**Abort.** Writing a one into this bit position causes the DMA controller to abort the DMA operation currently in progress. The abortion of a DMA operation is acknowledged when the Abort (A) bit is set in the DMACxSTS register. Writing a zero to this bit position has no effect on the operation of the DMA channel.
2	SUSPEND	RW	0x0	**Suspend.** Writing a one to this bit position suspends DMA descriptor processing.
31:3	Reserved	RO	0x0	Reserved field.

DMAC[1:0]CFG - DMA Channel Configuration (0x504/604)

Bit Field	Field Name	Type	Default Value	Description
0	DISDPTRL	RW	0x1	**Disable DMACxDPTRL Descriptor Processing Initiation.** When this bit is set, initiation of DMA descriptor processing as a side-effect writing to the DMCAxDPTRL register is disabled.

Notes

Bit Field	Field Name	Type	Default Value	Description
1	DISDPTRH	RW	0x1	**Disable DMACxDPTRH Descriptor Processing Initiation.** When this bit is set, initiation of DMA descriptor processing as a side-effect writing to the DMACxDPTRH register is disabled.
2	DISNDPTRL	RW	0x1	**Disable DMACxNDPTRL Descriptor Processing Initiation.** When this bit is set, initiation of DMA descriptor processing as a side-effect writing to the DMACxNDPTRL register is disabled.
3	DISNDPTRH	RW	0x1	**Disable DMACxNDPTRH Descriptor Processing Initiation.** When this bit is set, initiation of DMA descriptor processing as a side-effect writing to the DMACxNDPTRH register is disabled.
5:4	DSCP	RW	0x0	**Descriptor Status Check Processing.** This field indicates the action taken by the DMA channel when a fetched descriptor to be processed has a descriptor status (i.e., DSTS field) other than “unprocessed descriptor”. 0x0 - Abort processing 0x1 - Process descriptor 0x2 - Process next descriptor 0x3 - reserved Refer to section DMA Descriptor Processing on page 15-15 for details.
6	ODRC	RW	0x1	**Outstanding Data Request Control.** This field controls the number of outstanding requests issued by the DMA channel when processing a descriptor. 0x0- One outstanding request. 0x1 - Two outstanding requests. Refer to section DMA Outstanding Requests on page 15-21 for details.
7	PCRC	RW	0x0	**Poisoned Completion Reception Control.** This field controls the behavior of the DMA channel upon receiving a poisoned completion TLP associated with an outstanding data read request during descriptor processing. 0x0 - Discard the poisoned completion TLP and abort operation. 0x1 - Process the completion TLP normally and continue operation (i.e., do not abort). Regardless of the setting of this field, the reception of the poisoned completion TLP is handled as indicated in Table 15.12 (e.g., the poisoned error is logged in the DMA function’s PCI Status (PCISTS) and PCI Express AER registers appropriately).
8	DRRO	RW	0x0	**Descriptor Read Relaxed Ordering.** This field specifies the state of the relaxed ordering attribute in descriptor read operations.

Notes

Bit Field	Field Name	Type	Default Value	Description
9	DRNS	RW	0x0	**Descriptor Read No Snoop.** This field specifies the state of the no snoop attribute in descriptor read operations.
10	DWRO	RW	0x0	**Descriptor Write Relaxed Ordering.** This field specifies the state of the relaxed ordering attribute in descriptor write operations.
11	DWNS	RW	0x0	**Descriptor Write No Snoop.** This field specifies the state of the no snoop attribute in descriptor write operations.
14:12	DTC	RW	0x0	**Descriptor Traffic Class.** This field specifies the traffic class used by descriptor read and write operations.
15	Reserved	RO	0x0	Reserved field.
17:16	DPREFETCH	RW	0x0	**DMA Descriptor Prefetch Level.** This field specifies the number of DMA descriptors that DMA channel will attempt to prefetch. 0x0 - (disable) Disable DMA descriptor prefetching 0x1 -(one) Prefetch one DMA descriptor 0x2 - Reserved 0x3 - Reserved
31:18	Reserved	RO	0x0	Reserved field.

DMAC[1:0]STS - DMA Channel Status (0x508/608)

Bit Field	Field Name	Type	Default Value	Description
0	F	RW1C	0x0	**Finished.** This bit is set when descriptor processing of a descriptor with the IOF bit set completes normally. Once set, this bit is never cleared by hardware.
1	A	RW1C	0x0	**Abort.** This bit is set when descriptor processing is aborted. Once set, this bit is never cleared by hardware.
2	E	RW1C	0x0	**Error.** This bit is set when an unmasked channel error is detected (i.e., an unmasked error bit in the DMACxERRSTS register is set). Initiation and resumption of DMA channel descriptor processing is inhibited while this bit is set. See RUN bit definition in the DMACxCTL register. Once set, this bit is never cleared by hardware.
3	C	RW1C	0x0	**Chain.** This bit is set when a descriptor chaining operation takes place. Once set, this bit is never cleared by hardware.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	H	RW1C	0x0	**Halt.** This bit is set when the DMA channel halts descriptor processing. Once set, this bit is never cleared by hardware.
5	S	RW1C	0x0	**Suspend.** This bit is set when the DMA channel suspends descriptor processing. Once set, this bit is never cleared by hardware.
31:6	Reserved	RO	0x0	Reserved field.

DMAC[1:0]MSK - DMA Channel Status Mask (0x50C/60C)

Bit Field	Field Name	Type	Default Value	Description
0	F	RW	0x1	**Finished.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
1	A	RW	0x1	**Abort.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
2	E	RW	0x1	**Error.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
3	C	RW	0x1	**Chain.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
4	H	RW	0x1	**Halt.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
5	S	RW	0x1	**Suspend.** When this bit is set, the corresponding bit in the DMACxSTS register is masked from generating an interrupt.
31:6	Reserved	RO	0x0	Reserved field.

Notes

DMAC[1:0]ERRSTS - DMA Channel Error Status (0x510/610)

Bit Field	Field Name	Type	Default Value	Description
0	DSCA	RW1C	0x0	**Descriptor Alignment Error.** De-featured.
1	DSCP	RW1C	0x0	**Descriptor Poisoned Error.** This bit is set when a poisoned completion is received in response to a descriptor read request. Refer to section Poisoned TLP Reception on page 15-32 for details.
2	Reserved	RO	0x0	Reserved field.
3	DSCUR	RW1C	0x0	**Descriptor Unsupported Request Error.** This bit is set when an unsupported request completion is received in response to a descriptor read. Refer to section Completion with UR Status Received on page 15-33 for details.
4	DSCCA	RW1C	0x0	**Descriptor Completer Abort Error.** This bit is set when a completer abort completion is received in response to a descriptor read. Refer to section Completion with CA Status Received on page 15-34 for details.
5	DSCCT	RW1C	0x0	**Descriptor Completion Time-Out Error.** This bit is set when a completion time-out is detected during a descriptor read. Refer to section Completion Timeout on page 15-33 for details.
6	DSCFMT	RW1C	0x0	**Descriptor Format Error.** De-featured.
7	DSCF	RW1C	0x0	**Descriptor Status Check Failed.** De-featured.
8	SAE	RW1C	0x0	**Source Address Expired.** De-featured.
9	DAE	RW1C	0x0	**Destination Address Expired.** De-featured.
15:10	Reserved	RO	0x0	Reserved field.
16	DATP	RW1C	0x0	**Data Poisoned Error.** This bit is set when a poisoned completion is received in response to a data read request. Refer to section Poisoned TLP Reception on page 15-32 for details.
17	Reserved	RO	0x0	Reserved field.
18	DATUR	RW1C	0x0	**Data Unsupported Request Error.** This bit is set when a completion with status UR is received in response to a data read request. Refer to section Completion with UR Status Received on page 15-33 for details.

Notes

Bit Field	Field Name	Type	Default Value	Description
19	DATCA	RW1C	0x0	**Data Completer Abort Error.** This bit is set when a completion with status CA is received in response to a data read request. Refer to section Completion with CA Status Received on page 15-34 for details.
20	Reserved	RW1C	0x0	Reserved field.
30:21	Reserved	RO	0x0	Reserved field.
31	ECRCE	RW1C	0x0	**ECRC Error.** This bit is set when the DMA function receives a TLP with ECRC error. Refer to section ECRC Errors on page 15-32 for details.

DMAC[1:0]ERRMSK - DMA Channel Error Mask (0x514/614)

Bit Field	Field Name	Type	Default Value	Description
0	DSCA	RW	0x1	**Descriptor Alignment Error.** De-featured.
1	DSCP	RW	0x1	**Descriptor Poisoned Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
2	DSCECRC	RW	0x1	**De-featured.** This bit has no effect as the corresponding status bit in the DMAC[1:0]ERRSTS register is invalid.
3	DSCUR	RW	0x1	**Descriptor Unsupported Request Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
4	DSCCA	RW	0x1	**Descriptor Completer Abort Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
5	DSCCT	RW	0x1	**Descriptor Completion Time-Out Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
6	DSCFMT	RW	0x1	**Descriptor Format Error.** De-featured.
7	DSCF	RW	0x1	**Descriptor Status Check Failed.** De-featured.
8	SAE	RW	0x1	**Source Address Expired.** De-featured.
9	DAE	RW	0x1	**Destination Address Expired.** De-featured.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:10	Reserved	RO	0x0	Reserved field.
16	DATP	RW	0x1	**Data Poisoned Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
17	DATECRC	RW	0x1	**De-featured.** This bit has no effect as the corresponding status bit in the DMAC[1:0]ERRSTS register is invalid.
18	DATUR	RW	0x1	**Data Unsupported Request Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
19	DATCA	RW	0x1	**Data Completer Abort Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.
20	Reserved	RW	0x1	Reserved field.
30:21	Reserved	RO	0x0	Reserved field.
31	ECRCE	RW	0x0	**ECRC Error.** When this bit is set, the corresponding bit in the DMACx-ERRSTS register is masked from setting the Error (E) bit in the DMACxSTS register.

DMAC[1:0]SSIZE - DMA Channel Stride Size (0x518/618)

Bit Field	Field Name	Type	Default Value	Description
11:0	SSSIZE	RO	0x0	**Source Stride Size.** This field specifies the DMA channel source addressing stride size. A value of zero indicates an infinite stride size. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.
15:12	Reserved	RO	0x0	Reserved field.
27:16	DSSIZE	RO	0x0	**Destination Stride Size.** This field specifies the DMA channel destination addressing stride size. A value of zero indicates an infinite stride size. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.
31:28	Reserved	RO	0x0	Reserved field.

Notes

DMAC[1:0]SSCTL - DMA Channel Source Stride Control (0x51C/61C)

Bit Field	Field Name	Type	Default Value	Description
15:0	SDIST	RO	0x0	**Stride Distance.** This field specifies the DMA channel stride distance in bytes. This value in this field is a signed number in two's complement notation. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.
31:16	SCOUNT	RO	0x1	**Stride Count.** This field specifies the DMA channel stride count. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.

DMAC[1:0]DSCTL - DMA Channel Destination Stride Control (0x520/620)

Bit Field	Field Name	Type	Default Value	Description
15:0	SDIST	RO	0x0	**Stride Distance.** This field specifies the DMA channel stride distance in bytes. This value in this field is a signed number in two's complement notation. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.
31:16	SCOUNT	RO	0x1	**Stride Count.** This field specifies the DMA channel stride count. The value in this field is modified by a stride control DMA descriptor. The value in this field is not modified as a result of a data transfer operation.

DMAC[1:0]RRCTL - DMA Channel Request Rate Control (0x524/624)

Bit Field	Field Name	Type	Default Value	Description
15:0	RR	RW	0x0	**Request Rate.** This field specifies the minimum time between DMA data transfer read requests issued by the DMA channel in units of 4 nanoseconds per requested DWord. A value of zero indicates that there is no minimum gap and that DMA data read requests should be issued as fast as possible. Note that the value in this register may be updated by the RR field in a Stride Control DMA descriptor, when the RRU bit is set in the descriptor (refer to Table 15.4). To avoid race conditions, software must avoid writing to this field when the RRU bit is set in a Stride Control descriptor being processed by the DMA.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:16	Reserved	RO	0x0	Reserved field.

DMAC[1:0]DPTRL - DMA Channel Descriptor Pointer Low (0x528/628)

Bit Field	Field Name	Type	Default Value	Description
31:0	DPTRL	RW	0x0	**Descriptor Pointer Low.** This field is initialized with the lower 32-bits of the 64-bit address of the first DMA descriptor in a descriptor list. Writing a value to this register automatically starts DMA descriptor processing and causes the RUN bit in the DMAxC register to be set if the E bit in the DMACxSTS register is cleared. Writing to this field while a DMA is in progress produces undefined results. The DISADPTRL bit in the DMACxCFG register may be used to disabled the initiation of DMA descriptor processing as a side-effect of writing to this register. Writing a zero to this field modifies the contents of the register but does not automatically start DMA descriptor processing. The value returned by this field when read represents the address of the currently active DMA descriptor. The DMACxDPTRL and DMACxDPTRH together form a 64-bit address.

DMAC[1:0]DPTRH - DMA Channel Descriptor Pointer High (0x52C/62C)

Bit Field	Field Name	Type	Default Value	Description
31:0	DPTRH	RW	0x0	**Descriptor Pointer High.** This field is initialized with the upper 32-bits of the 64-bit address of the first DMA descriptor in a descriptor list. Writing a value to this register automatically starts DMA descriptor processing and causes the RUN bit in the DMAxC register to be set if the E bit in the DMACxSTS register is cleared. Writing to this field while a DMA is in progress produces undefined results. The DISADPTRH bit in the DMACxCFG register may be used to disabled the initiation of DMA descriptor processing as a side-effect of writing to this register. Writing a zero to this field modifies the contents of the register but does not automatically start DMA descriptor processing. The value returned by this field when read represents the address of the currently active DMA descriptor if the DMA is running. The DMACxDPTRL and DMACxDPTRH together form a 64-bit address.

Notes

DMAC[1:0]NDPTRL - DMA Channel Next Descriptor Pointer Low (0x530/630)

Bit Field	Field Name	Type	Default Value	Description
31:0	NDPTRL	RW	0x0	**Next Descriptor Pointer Low.** This field is initialized with the lower 32-bits of the 64-bit address of the first DMA descriptor in the chaining descriptor list. If the DMA is not running, writing a value to this register automatically starts DMA descriptor processing and causes the RUN bit in the DMAxC register to be set if the E bit in the DMACxSTS register is cleared. The DISANDPTRL bit in the DMACxCFG register may be used to disabled the initiation of DMA descriptor processing as a side-effect of writing to this register. Writing a zero to this field modifies the contents of the register but does not automatically start DMA descriptor processing. The DMACxNDPTRL and DMACxNDPTRH together form a 64-bit address.

DMAC[1:0]NDPTRH - DMA Channel Next Descriptor Pointer High (0x534/634)

Bit Field	Field Name	Type	Default Value	Description
31:0	NDPTRH	RW	0x0	**Next Descriptor Pointer High.** This field is initialized with the upper 32-bits of the 64-bit address of the first DMA descriptor in the chaining descriptor list. If the DMA is not running, writing a value to this register automatically starts DMA descriptor processing and causes the RUN bit in the DMAxC register to be set if the E bit in the DMACxSTS register is cleared. The DISANDPTRH bit in the DMACxCFG register may be used to disabled the initiation of DMA descriptor processing as a side-effect of writing to this register. Writing a zero to this field modifies the contents of the register but does not automatically start DMA descriptor processing. The DMACxNDPTRL and DMACxNDPTRH together form a 64-bit address.

Global Address Space Access Registers

GASAADDR - Global Address Space Access Address (0xFF8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
18:2	GADDR	RW	0x0	**Global Address.** This field selects the system address of the register to be accessed via the GASADATA register. The following restrictions apply regarding the programming of this register: 1) The value of this register must not be programmed to point to the address of the GASAADDR or GASADATA register in this or any other function. 2) The value of this register must not be programmed to point to the address of the Extended Configuration Address and Data registers (ECFGADDR and ECFGDATA) in this or any other function. 3) The value of this register must not be programmed to point to the NT Mapping Table Address and Data (NTMTBLADDR and NTMTBLDATA) registers in any NT function. Violations of these rules produce undefined results.
31:19	Reserved	RO	0x0	Reserved field.

GASADATA - Global Address Space Access Data (0xFFC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Data.** A read from this field will return the global space register value pointed to by the GASAADDR register. A write to this field will update the contents of the global space register pointed to by the GASAADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. SMBus reads of this field return a value of zero and SMBus writes have no effect.

Switch Configuration and Status Registers

Notes

Switch Control and Status Registers

SWCTL - Switch Control (0x0000)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
2	RSTHALT	RW	HWINIT SWSticky	**Reset Halt.** When this bit is set, all of the switch logic except the SMBus interface remains in a quasi-reset state. In this state, registers in the device may be initialized by the slave SMBus interface. When this bit is cleared, normal operation ensues. The initial value of this bit is that of the RSTHALT signal in the boot configuration vector. Software/Firmware is only allowed to write a value of zero to this bit. Writing a value of one to this bit produces undefined results
3	REGUNLOCK	RW	0x0 SWSticky	**Register Unlock.** When this bit is set, the contents of registers and fields of type Read and Write when Unlocked (RWL) are modified when written. When this bit is cleared, all registers and fields denoted as become read-only. While the initial value of this field is cleared, it is set during a switch fundamental reset sequence to allow the serial EEPROM to modify the contents of fields.
18:4	Reserved	RW	0x1000	Reserved field.
19	BDISCARD	RW	0x0 SWSticky	**Discard Vendor Defined Broadcast Messages.** When this bit is set, vendor defined Type 1 broadcast messages received on the upstream port are silently discarded and not forwarded downstream. Silently discarding a TLP means that flow control credits are returned, TLP contents are discarded, and no error bits are set.
31:20	Reserved	RW	0x1	Reserved field.

Notes

BCVSTS - Boot Configuration Vector Status (0x0004)

Bit Field	Field Name	Type	Default Value	Description
3:0	SWMODE	RO	HWINIT	**Switch Mode.** Boot configuration vector value sampled during a switch fundamental reset.
4	Reserved	RO	0x0	Reserved field.
5	GCLKFSEL	RO	HWINIT	**Global Clock Frequency Select.** Boot configuration vector value sampled during a switch fundamental reset.
6	Reserved	RO	0x0	Reserved field.
8:7	SSMBADDR	RO	HWINIT	**Slave SMBus Address.** Boot configuration vector value sampled during a switch fundamental reset.
9	RSTHALT	RO	HWINIT	**Reset Halt.** Boot configuration vector value sampled during a switch fundamental reset.
13:10	Reserved	RO	0x0	Reserved field.
15:14	CLKMODE	RO	HWINIT	**Clock Mode.** Boot configuration vector value sampled during a switch fundamental reset.
17:16	STK0CFG	RO	HWINIT	**Stack 0 Initial Configuration.** Boot configuration vector value sampled during a switch fundamental reset.
19:18	STK1CFG	RO	HWINIT	**Stack 1 Initial Configuration.** Boot configuration vector value sampled during a switch fundamental reset.
24:20	STK2CFG	RO	HWINIT	**Stack 2 Initial Configuration.** Boot configuration vector value sampled during a switch fundamental reset.
29:25	STK3CFG	RO	HWINIT	**Stack 3 Initial Configuration.** Boot configuration vector value sampled during a switch fundamental reset.
31:30	Reserved	RO	0x0	

Notes

PCLKMODE - Port Clocking Mode (0x0008)

Bit Field	Field Name	Type	Default Value	Description
1:0	P0CLKMODE	RW	0x0 SWSticky	**Ports 0 and 1 Clocking Mode.** This field selects the port clocking mode used by the corresponding switch port(s). The port clocking mode may be modified at any time. See section Port Clocking Modes on page 2-3 for details. 0x0 - (global) Port is configured to operate in global clocked mode. 0x1 - (local) Port is configured to operate in local port clocked mode. 0x2 - Reserved 0x3 - Reserved
3:2	P2CLKMODE	RW	0x0 SWSticky	**Ports 2 and 3 Clocking Mode.** See P0CLKMODE description.
5:4	P4CLKMODE	RW	0x0 SWSticky	**Ports 4 and 5 Clocking Mode.** See P0CLKMODE description.
7:6	P6CLKMODE	RW	0x0 SWSticky	**Ports 6 and 7 Clocking Mode.** See P0CLKMODE description.
9:8	P8CLKMODE	RW	0x0 SWSticky	**Ports 8 to 11 Clocking Mode.** See P0CLKMODE description.
11:10	P12CLKMODE	RW	0x0 SWSticky	**Port 12 to 15 Clocking Mode.** See P0CLKMODE description.
13:12	P16CLKMODE	RW	0x0 SWSticky	**Port 16 to 19 Clocking Mode.** See P0CLKMODE description.
15:14	P20CLKMODE	RW	0x0 SWSticky	**Port 20 to 23 Clocking Mode.** See P0CLKMODE description.
31:16	Reserved	RO	0x0	

STK0CFG - Stack Configuration (0x0010)

Bit Field	Field Name	Type	Default Value	Description
4:0	STKCFG	RW	HWINIT SWSticky	**Stack Configuration.** This field selects the configuration of the stack. The initial value of bits[4:2] in this field is 0x0. The initial value of bits [1:0] in this field depends on the setting of the STK0CFG[1:0] pins, as described in section Stack Configuration on page 3-5. Refer to this section for further details.
31:5	Reserved	RO	0x0	Reserved field.

Notes

STK1CFG - Stack Configuration (0x0014)

Bit Field	Field Name	Type	Default Value	Description
4:0	STKCFG	RW	HWINIT SWSticky	**Stack Configuration.** This field selects the configuration of the stack. The initial value of bits[4:2] in this field is 0x0. The initial value of bits [1:0] in this field depends on the setting of the STK1CFG[1:0] pins, as described in section Stack Configuration on page 3-5. Refer to this section for further details.
31:5	Reserved	RO	0x0	Reserved field.

STK2CFG - Stack Configuration (0x0018)

Bit Field	Field Name	Type	Default Value	Description
4:0	STKCFG	RW	HWINIT SWSticky	**Stack Configuration.** This field selects the configuration of the stack. The initial value of this field depends on the setting of the STK2CFG pins, as described in section Stack Configuration on page 3-5. Refer to this section for further details.
31:5	Reserved	RO	0x0	Reserved field.

STK3CFG - Stack Configuration (0x001C)

Bit Field	Field Name	Type	Default Value	Description
4:0	STKCFG	RW	HWINIT SWSticky	**Stack Configuration.** This field selects the configuration of the stack. The initial value of this field depends on the setting of the STK3CFG pins, as described in section Stack Configuration on page 3-5. Refer to this section for further details.
31:5	Reserved	RO	0x0	Reserved field.

Notes

Internal Switch Timers

RDRAINDELAY - Reset Drain Delay (0x0080)

Bit Field	Field Name	Type	Default Value	Description
15:0	DRAINDELAY	RW	0x0FA SWSticky	**Reset Drain Delay** This field specifies the delay in microseconds for TLPs queued in the switch (i.e., IFB or EFB) to drain during a port reset, partition upstream secondary bus reset, partition downstream secondary bus reset, partition hot reset, or partition fundamental reset. The default value corresponds to 250 microseconds. Given that reset events may occur at the time that TLP traffic is flowing through the switch, decreasing this value is discouraged except for the cases explicitly stated in this specification.
31:16	Reserved	RO	0x0	Reserved field.

POMCDELAY - Port Operating Mode Change Drain Delay (0x0084)

Bit Field	Field Name	Type	Default Value	Description
15:0	POMCDELAY	RW	0x03E8 SWSticky	**Port Operating Mode Change Drain Delay** This field specifies the delay in microseconds for TLPs queued in the switch (i.e., IFB or EFB) to drain during a port operating mode change. This corresponds to the minimum time between the assertion of the Operating Mode Change Initiated (OMCI) bit in the Switch Port Status (SWPORTxSTS) register and the Operating Mode Change Completed (OMCC) bit SWPORTxSTS register. The default value corresponds to 1 millisecond. Reducing this delay is not recommended unless the platform can guarantee that traffic will be quiesced in a port before the port's operating mode is changed. If this guarantee is met, this delay may be reduced down to 0.
31:16	Reserved	RO	0x0	Reserved field.

Notes

SEDELAY - Side Effect Delay (0x0088)

Bit Field	Field Name	Type	Default Value	Description
15:0	SEDELAY	RW	0x03E8 SWSticky	**Side Effect Delay** This field specifies the delay in microseconds from the generation of a completion for a configuration request with an associated side-effect to the side effect action taking place. The intent of this delay is provide sufficient time for a completion to be returned to the link partner prior to the side effect. Refer to section Configuration Register Side-Effects on page 19-3 for details. The default value corresponds to 1 millisecond. Decreasing this delay is discouraged, except for the cases explicitly listed in this specification.
31:16	Reserved	RO	0x0	Reserved field.

USSBRDELAY - Upstream Secondary Bus Reset Delay (0x008C)

Bit Field	Field Name	Type	Default Value	Description
15:0	USSBR	RW	0x0 SWSticky	**Upstream Secondary Bus Reset Delay** This field specifies the delay in microseconds from when a configuration request that initiates a secondary bus reset is processed to the start of the secondary bus reset action. The default value corresponds to no delay. Decreasing this delay is discouraged, except for the cases explicitly listed in this specification.
31:16	Reserved	RO	0x0	Reserved field.

Notes

Switch Partition and Port Registers

SWPART[7:0]CTL - Switch Partition x Control

Bit Field	Field Name	Type	Default Value	Description
1:0	STATE	RW	HWINIT SWSticky	**Switch Partition State.** This field controls the state of the switch partition. 0x0 - (disable) Disabled 0x1 - (active) Active 0x2 - reserved 0x3 - (reset) Reset The initial value of this field depends on the Switch Mode selected via the boot configuration vector. Refer to section Partition Resets on page 3-10 for details.
2	DLDHRST	RW	0x0 SWSticky	**Disable Link Down Hot Reset.** When this bit is set, hot resets due to the data link layer of the upstream port reporting a DL_Down condition are disabled. As a result, the DL_Down condition on the upstream port does not cause a hot reset on the upstream port (i.e., the port's configuring space is not affected) and the hot reset is not propagated to the downstream switch ports in the partition. Note that the upstream port's data-link and physical layers are not affected by this bit.
18:3	Reserved	RO	0x0	Reserved field.
19	FEN	RW	0x0 SWSticky	**Failover Enable**. When set, this bit enables initiation of the failover reconfiguration specified by the SWPARTxFCTL register when a failover is triggered by the failover capability selected by the Failover Capability Select (FCAPSEL) field in this register.
21:20	FCAPSEL	RW	0x0 SWSticky	**Failover Capability Select.** This field selects the failover capability associated with the partition. 0x0 - Failover Capability 0 0x1 - Failover Capability 1 0x2 - Failover Capability 2 0x3 - Failover Capability 3
31:22	Reserved	RO	0x0	Reserved field.

Notes

SWPART[7:0]STS - Switch Partition x Status

Bit Field	Field Name	Type	Default Value	Description
0	SCI	RW1C	0x0 SWSticky	**Switch Partition State Change Initiated.** This bit is set when a switch partition state change is initiated.
1	SCC	RW1C	0x0 SWSticky	**Switch Partition State Change Completed.** This bit is set when a switch partition state change has completed.
2	PFAILOVER	RW1C	0x0 SWSticky	**Primary Failover Status.** This bit is set when a switch partition state is associated with a primary failover operation.
3	SFAILOVER	RW1C	0x0 SWSticky	**Secondary Failover Status.** This bit is set when a switch partition state is associated with a secondary failover operation.
4	Reserved	RO	0x0	Reserved field.
6:5	STATE	RO	HWINIT	**Switch Partition State.** This field contains the current state of the switch partition. Due to the time it takes for a partition state change to complete, this value may be different than that in the STATE field in the SWPARTxCTL register. 0x0 - (disable) Disabled 0x1 - (active) Active 0x2 - reserved 0x3 - (reset) Reset
7	Reserved	RO	0x0	Reserved field.
8	US	RO	HWINIT	**Upstream Port.** This bit is set when there is an upstream port associated with the switch partition. Refer to section Overview on page 5-1 for a list of port operating modes that are considered upstream port modes.
13:9	USID	RO	HWINIT	**Upstream Port ID.** When the Upstream Port (US) bit is set in this register, this field specifies the switch port ID of the port that will act as the upstream port of the switch partition.
14	NT	RO	HWINIT	**NT Function in Upstream Port.** This bit is set when there is an NT function in the upstream port associated with the switch partition.
15	DMA	RO	HWINIT	**DMA Function in Upstream Port.** This bit is set when there is a DMA function in the upstream port associated with the switch partition.
31:16	Reserved	RO	0x0	Reserved field.

Notes

SWPART[7:0]FCTL - Switch Partition x Failover Control

Bit Field	Field Name	Type	Default Value	Description
1:0	PFSTATE	RW	0x0 SWSticky	**Primary Failover Switch Partition State** This field specifies the primary failover state of the partition. 0x0 - (disable) Disabled 0x1 - (active) Active 0x2 - reserved 0x3 - (reset) Reset
9:2	Reserved	RO	0x0	Reserved field.
11:10	SFSTATE	RW	0x0 SWSticky	**Secondary Failover Switch Partition State** This field specifies the secondary failover state of the partition. 0x0 - (disable) Disabled 0x1 - (active) Active 0x2 - reserved 0x3 - (reset) Reset
31:12	Reserved	RO	0x0	Reserved field.

SWPORTxCTL - Switch Port x Control

Bit Field	Field Name	Type	Default Value	Description
3:0	MODE	RW	HWINIT SWSticky	**Port Mode.** This field controls the operating mode of the switch port. 0x0 - (disable) Disabled 0x1 -(downstream) Downstream switch port 0x2 - (upstream) Upstream switch port 0x3 - (ntb) NT function 0x4 - (upstream_ntb) Upstream switch port with NT function 0x5 - (unattached) Unattached 0x6 - (upstream_dma) Upstream switch port with DMA function 0x7 - (upstream_ntb_dma) Upstream switch port with NT and DMA functions 0x8 - (ntb_dma) NT with DMA function Others - reserved The initial value of this field depends on the Switch Mode selected via the boot configuration vector. Refer to section Partition Resets on page 3-10 for details.
6:4	SWPART	RW	0x0 SWSticky	**Switch Partition.** For operational port modes (i.e., all modes except disabled and unattached), this field specifies the switch partition to which the port is attached. In non operational modes (i.e., disabled or unattached), the value of this field has no effect. Valid settings for this field are zero to seven.
9:7	Reserved	RO	0x0	Reserved field.
14:10	DEVNUM	RW	HWINIT SWSticky	**Device Number.** When the port is configured as a downstream switch port, this field specifies the device number associated with the port. In all other port modes, this field has no effect. The initial value of this field corresponds to the port number.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	Reserved	RO	0x0	Reserved field.
17:16	OMA	RW	0x1 SWSticky	**Operating Mode Change Action.** This field specifies the action taken when a modification is made to the operating mode of a port. 0x0 - (noaction) No action - preserve state 0x1 - (reset) Port reset - behavior associated with fundamental reset others - reserved
18	Reserved	RO	0x0	Reserved field.
19	FEN	RW	0x0 SWSticky	**Failover Enable.** When set, this bit enables initiation of the failover reconfiguration specified by the SWPORTxFCTL register when a failover is triggered by the failover capability selected by the Failover Capability Select (FCAPSEL) field in this register.
21:20	FCAPSEL	RW	0x0 SWSticky	**Failover Capability Select.** This field selects the failover capability associated with the port. 0x0 - Failover Capability 0 0x1 - Failover Capability 1 0x2 - Failover Capability 2 0x3 - Failover Capability 3
31:22	Reserved	RO	0x0	Reserved field.

SWPORTxSTS - Switch Port x Status

Bit Field	Field Name	Type	Default Value	Description
0	OMCI	RW1C	0x0 SWSticky	**Operating Mode Change Initiated.** This bit is set when a port operating mode change is initiated.
1	OMCC	RW1C	0x0 SWSticky	**Operating Mode Change Completed.** This bit is set when a port operating mode change has completed.
2	PFAILOVER	RW1C	0x0 SWSticky	**Primary Failover Status.** This bit is set when a port change is associated with a primary failover operation.
3	SFAILOVER	RW1C	0x0 SWSticky	**Secondary Failover Status.** This bit is set when a port change is associated with a secondary failover operation.
4	LINKUP	RO	0x0	**Link Up.** This bit is set when the PCI Express data link layer associated with the port is 'DL_Up'. This bit is cleared when the data link layer is 'DL_Down'.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	LINKMODE	RO	HWINIT	**Link Mode.** This field indicates the operating mode of the lanes for the link associated with the port when the link is up. The value of this field is undefined when the link is down. Note that PHY link mode is not equivalent to port mode (i.e., a port in 'Unattached' mode which initially has an upstream PHY link mode may automatically change to a downstream PHY link mode if it link-trained with an endpoint device using crosslink). 0x0 - (upstream) Lane(s) associated with port behave as upstream lanes 0x1 - (downstream) Lane(s) associated with port behave as downstream lanes
9:6	MODE	RO	HWINIT	**Port Mode.** This field contains the current operating mode of the switch port. Due to the time it takes for a partition state change to complete, this value may be different than that of the MODE field in the SWPORTxCTL register. If a port is disabled due to the associated partition being disabled, then this field indicates that the port is disabled (i.e., the field does not simply echo back the MODE field in the SWPORTxCTL register). 0x0 - (disable) Disabled 0x1 -(downstream) Downstream switch port 0x2 - (upstream) Upstream switch port 0x3 - (ntb) NT function 0x4 - (upstream_ntb) Upstream switch port with NT function 0x5 - (unattached) Unattached 0x6 - upstream_dma) Upstream switch port with DMA function 0x7 - (upstream_ntb_dma) Upstream switch port with NT and DMA functions 0x8 - (ntb_dma) NT with DMA function Others - reserved
12:10	SWPART	RO	HWINIT	**Switch Partition.** For operational port modes (e.g., downstream switch port, upstream switch port, NT endpoint, upstream switch port with NT endpoint), this field contains the current switch partition to which the port is attached. In non operational modes (i.e., disabled or unattached), the value of this field is undefined. Due to the time it takes for a partition state change to complete, this value may be different than that in the SWPART field in the SWPORTxCTL register.
15:13	Reserved	RO	0x0	Reserved field.
20:16	DEVNUM	RO	HWINIT	**Device Number.** When the port is configured as a downstream switch port, this field contains the device number associated with the port. In all other port modes, this value of this field is undefined. Due to the time it takes for a partition state change to complete, this value may be different than that in the DEVNUM field in the SWPORTxCTL register.
31:21	Reserved	RO	0x0	Reserved field.

Notes

SWPORTxFCTL - Switch Port x Failover Control

Bit Field	Field Name	Type	Default Value	Description
3:0	PFMODE	RW	0x0 SWSticky	**Primary Failover Port Mode.** This field specifies the primary failover port mode. On a primary failover event, the value of this field is transferred to the MODE field of the SWPORTxCTL register. 0x0 - (disable) Disabled 0x1 - (downstream) Downstream switch port 0x2 - (upstream) Upstream switch port 0x3 - (ntb) NT function 0x4 - (upstream_ntb) Upstream switch port with NT function 0x5 - (unattached) Unattached 0x6 - (upstream_dma) Upstream switch port with DMA function 0x7 - (upstream_ntb_dma) Upstream switch port with NT and DMA functions 0x8 - (ntb_dma) NT with DMA function Others - reserved
6:4	PFSWPART	RW	0x0 SWSticky	**Primary Failover Switch Partition.** This field specifies the primary failover switch partition. On a primary failover event, the value of this field is transferred to the SWPART field of the SWPORTxCTL register.
9:7	Reserved	RO	0x0	Reserved field.
14:10	PFDEVNUM	RW	0x0 SWSticky	**Primary Failover Device Number.** This field specifies the primary failover device number. On a primary failover event, the value of this field is transferred to the DEVNUM field of the SWPORTxCTL register.
15	Reserved	RO	0x0	Reserved field.
19:16	SFMODE	RW	0x0 SWSticky	**Secondary Failover Port Mode.** This field specifies the secondary failover port mode. On a secondary failover event, the value of this field is transferred to the MODE field of the SWPORTxCTL register. 0x0 - (disable) Disabled 0x1 - (downstream) Downstream switch port 0x2 - (upstream) Upstream switch port 0x3 - (ntb) NT function 0x4 - (upstream_ntb) Upstream switch port with NT function 0x5 - (unattached) Unattached 0x6 - (upstream_dma) Upstream switch port with DMA function 0x7 - (upstream_ntb_dma) Upstream switch port with NT and DMA functions 0x8 - (ntb_dma) NT with DMA function Others - reserved
22:20	SFSWPART	RW	0x0 SWSticky	**Secondary Failover Switch Partition.** This field specifies the secondary failover switch partition. On a secondary failover event, the value of this field is transferred to the SWPART field of the SWPORTxCTL register.

Notes

Bit Field	Field Name	Type	Default Value	Description
25:23	Reserved	RO	0x0	Reserved field.
30:26	SFDEVNUM	RW	0x0 SWSticky	**Secondary Failover Device Number.** This field specifies the secondary failover device number. On a secondary failover event, the value of this field is transferred to the DEVNUM field of the SWPORTxCTL register.
31	Reserved	RO	0x0	Reserved field.

Failover Capability Registers

FCAP[3:0]CTL - Failover Capability x Control

Bit Field	Field Name	Type	Default Value	Description
0	FSWTRIG	RW	0x0 SWSticky	**Failover Software Trigger.** Writing a one to this bit triggers a failover. If the current failover mode reported in the Failover Mode (FMODE) field in the FCAPxSTS register is primary, then a secondary failover is triggered (i.e., transition from primary to secondary). If the current failover mode is secondary, then a primary failover is triggered (i.e., transition from secondary to primary). This field always returns a value of zero when read.
1	FSIGEN	RW	0x0 SWSticky	**Failover Signal Trigger Enable.** When this bit is set, a failover is initiated when the state of the corresponding Failover Trigger (FAILOVERx) input signal changes state.
2	FSIGPOL	RW	0x0 SWSticky	**Failover Signal Polarity.** This field controls the polarity of the Failover Trigger (FAILOVERx) input signal. 0x0 - (activehigh) When enabled, a secondary failover is triggered when the FAILOVERx signal transitions from low to high. When enabled a primary failover is triggered when the FAILOVERx signal transitions from high to low. 0x1 - (activelow) When enabled, a secondary failover is triggered when the FAILOVERx signal transitions from high to low. When enabled a primary failover is triggered when the FAILOVERx signal transitions from low to high.
3	FTIMEN	RW	0x0 SWSticky	**Failover Timer Trigger Enable.** When this bit is set, a failover is triggered whenever the Count (COUNT) field in the FCAPxTIMER register reaches zero (i.e., the failover watchdog timer expires).
31:4	Reserved	RO	0x0	Reserved field.

Notes

FCAP[3:0]STS - Failover Capability x Status

Bit Field	Field Name	Type	Default Value	Description
0	FMODE	RO	0x0 SWSticky	**Failover Mode**. This field indicates the current failover mode. When a failover mode change is in progress, this field returns the new mode. 0x0 -(primary) Primary mode. 0x1 -(secondary) Secondary mode.
1	FMCI	RW1C	0x0 SWSticky	**Failover Mode Change Initiated.** This bit is set whenever a failover mode change is initiated (i.e., when a failover trigger associated with the failover capability is actuated).
2	FMCC	RW1C	0x0 SWSticky	**Failover Mode Change Completed.** This bit is set whenever a failover mode change and all associated side-effects have completed. Note that in the case where no ports or partitions are sensitive to the failover capability, the failover mode change completes immediately after the failover is triggered (i.e., this bit is set immediately after the FMCI bit is set).
31:3	Reserved	RO	0x0	Reserved field.

FCAP[3:0]TIMER - Failover Capability x Watchdog Timer

Bit Field	Field Name	Type	Default Value	Description
31:0	COUNT	RW	0x0 SWSticky	**Watchdog Timer Count.** This field contains the current failover watchdog timer count value. The value in this field is decremented by one every microsecond (i.e., 1 µS). When the field reaches zero, it stops decrementing and is said to have expired. When the value of this field transitions from one to zero and the Failover Timer Trigger Enable (FTIMEN) bit in the corresponding FCAPxCTL register is set, then a failover is triggered. If the current failover mode reported in the Failover Mode (FMODE) field in the FCAPxSTS register is primary, then a secondary failover is triggered (i.e., transition from primary to secondary). If the current failover mode is secondary, then a primary failover is triggered (i.e., transition from secondary to primary).

Notes

Protection

GASAPROT - Global Address Space Access Protection (0x0700)

Bit Field	Field Name	Type	Default Value	Description
23:0	PORT	RW	0x0 SWSticky	**Port.** Each bit in this field corresponds to a switch port. When a bit in this field is set, access to global address space using a GASADDR and GASADATA register pair from the port is disabled and fields in both registers become read-only with a value of zero. Note that in multi-function upstream port, a bit in this field affects global address space accesses from all functions of the port.
31:24	Reserved	RO	0x0	Reserved field.

NTMTBLPROT[7:0] - Partition x NT Mapping Table Protection

Bit Field	Field Name	Type	Default Value	Description
5:0	TBLBASE	RW	0x0 SWSticky	**NT Mapping Table Base.** This field specifies the NT Mapping table base value for the corresponding partition.
7:6	Reserved	RO	0x0	Reserved field.
13:8	TBLLIMIT	RW	0x3F SWSticky	**NT Mapping Table Limit.** This field specifies the NT Mapping table limit value for the corresponding partition.
15:14	Reserved	RO	0x0	Reserved field.
23:16	PARTBLOCK	RW	0x0 SWSticky	**Partition Blocking Vector.** This field specifies values that may be written to the Partition (PART) field in the NT Mapping table by the corresponding partition. Each bit in this field represents a partition number. Updates to the NT Mapping table are blocked if the PART field value of the update corresponds to a bit that is set in this field.
31:24	Reserved	RO	0x0	Reserved field.

Notes

Switch Event Registers

Refer to section Switch Events on page 16-1 for details on the operation of these registers.

SESTS - Switch Event Status (0x0C00)

Bit Field	Field Name	Type	Default Value	Description
0	LINKUP	RO	0x0	**Link Up Status.** This bit is set if an unmasked bit is set in the Switch Event Link Up Status (SELINKUPSTS) register.
1	LINKDN	RO	0x0	**Link Down Status.** This bit is set if an unmasked bit is set in the Switch Event Link Down Status (SELINKDNSTS) register.
2	FRST	RO	0x0	**Fundamental Reset Status.** This bit is set if an unmasked bit is set in the Switch Event Fundamental Reset Status (SEFRSTSTS) register.
3	HRST	RO	0x0	**Hot Reset Status.** This bit is set if an unmasked bit is set in the Switch Event Hot Reset Status (SEHRSTSTS) register.
4	FOVER	RO	0x0	**Failover Status.** This bit is set if an unmasked failover mode change bit is set.
5	GSIGNAL	RO	0x0	**Global Signal Status.** This bit is set if an unmasked bit is set in the Switch Event Global Signal Status (SEGSIGSTS) register.
7:6	Reserved	RO	0x0	Reserved field.
8	P0AER	RW1C	0x0 SWSticky	**Port 0 AER Error** This bit is set at the time that port 0 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
9	Reserved	RO	0x0	Reserved field.
10	P2AER	RW1C	0x0 SWSticky	**Port 2 AER Error** This bit is set at the time that port 2 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
11	Reserved	RO	0x0	Reserved field.
12	P4AER	RW1C	0x0 SWSticky	**Port 4 AER Error** This bit is set at the time that port 4 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
13	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	P6AER	RW1C	0x0 SWSticky	**Port 6 AER Error** This bit is set at the time that port 6 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
15	Reserved	RO	0x0	Reserved field.
16	P8AER	RW1C	0x0 SWSticky	**Port 8 AER Error** This bit is set at the time that port 8 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
19:17	Reserved	RO	0x0	Reserved field.
20	P12AER	RW1C	0x0 SWSticky	**Port 12 AER Error** This bit is set at the time that port 12 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
23:21	Reserved	RO	0x0	Reserved field.
24	P16AER	RW1C	0x0 SWSticky	**Port 16 AER Error** This bit is set at the time that port 16 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
27:25	Reserved	RO	0x0	Reserved field.
28	P20AER	RW1C	0x0 SWSticky	**Port 20 AER Error** This bit is set at the time that port 20 detects an AER error in one of its functions (i.e., any bit is set in the corresponding internal, non-software visible PAERSTS register) and the error is not masked by the corresponding Port AER Mask (PAERMSK) register.
31:29	Reserved	RO	0x0	Reserved field.

SEMSK - Switch Event Mask (0x0C04)

Bit Field	Field Name	Type	Default Value	Description
0	LINKUP	RW	0x1 SWSticky	**Link Up.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
1	LINKDN	RW	0x1 SWSticky	**Link Down.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
2	FRST	RW	0x1 SWSticky	**Fundamental Reset.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.

Notes

Bit Field	Field Name	Type	Default Value	Description
3	HRST	RW	0x1 SWSticky	**Hot Reset.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
4	FOVER	RW	0x1 SWSticky	**Failover.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
5	GSIGNAL	RW	0x1 SWSticky	**Global Signal.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
7:6	Reserved	RO	0x0	Reserved field.
8	P0AER	RW	0x1 SWSticky	**Port 0 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
9	Reserved	RO	0x0	Reserved field.
10	P2AER	RW	0x1 SWSticky	**Port 2 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
11	Reserved	RO	0x0	Reserved field.
12	P4AER	RW	0x1 SWSticky	**Port 4 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
13	Reserved	RO	0x0	Reserved field.
14	P6AER	RW	0x1 SWSticky	**Port 6 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
15	Reserved	RO	0x0	Reserved field.
16	P8AER	RW	0x1 SWSticky	**Port 8 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
17	Reserved	RO	0x0	Reserved field.
18	P10AER	RW	0x1 SWSticky	**Port 10 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
19	Reserved	RO	0x0	Reserved field.
20	P12AER	RW	0x1 SWSticky	**Port 12 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
23:21	Reserved	RO	0x0	Reserved field.
24	P16AER	RW	0x1 SWSticky	**Port 16 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.

Notes

Bit Field	Field Name	Type	Default Value	Description
27:25	Reserved	RO	0x0	Reserved field.
28	P20AER	RW	0x1 SWSticky	**Port 20 AER Error.** When this bit is set, the corresponding bit in the SESTS register is masked from generating a switch event.
31:29	Reserved	RO	0x0	Reserved field.

SEPMSK - Switch Event Partition Mask (0x0C08)

Bit Field	Field Name	Type	Default Value	Description
7:0	PMSK	RW	0xFF SWSticky	**Partition Mask.** Each bit in this field corresponds to a switch partition. When a bit in this field is set, switch events are masked to the corresponding partition. Bits associated with a partition not present in a product option have no effect on the operation of the device (i.e., they are a don't care).
31:8	Reserved	RO	0x0	Reserved field.

SELINKUPSTS - Switch Event Link Up Status (0x0C0C)

Bit Field	Field Name	Type	Default Value	Description
23:0	LINKUP	RW1C	0x0 SWSticky	**Link Up.** Each bit in this field corresponds to a switch port. When a link associated with a switch port transitions from link down to link up (i.e., the data link layer status transitions from DL_Down to DL_Up), then the corresponding bit in this register is set. Bits associated with ports that are not present due to the stack's configuration are never set (i.e., remain cleared).
31:24	Reserved	RO	0x0	Reserved field.

SELINKUPMSK - Switch Event Link Up Mask (0x0C10)

Bit Field	Field Name	Type	Default Value	Description
23:0	LINKUP	RW	0xFF_FFFF SWSticky	**Link Up.** When a bit in this field is set, the corresponding bit in the SELINKUPSTS register is masked from generating a switch event.
31:24	Reserved	RO	0x0	Reserved field.

Notes

SELINKDNSTS - Switch Event Link Down Status (0x0C14)

Bit Field	Field Name	Type	Default Value	Description
23:0	LINKDN	RW1C	0x0 SWSticky	**Link Down.** Each bit in this field corresponds to a switch port. When a link associated with a switch port transitions from link up to link down (i.e., the data link layer status transitions from DL_Up to DL_Down), then the corresponding bit in this register is set. Bits associated with ports that are not present due to the stack's configuration are never set (i.e., remain cleared).
31:24	Reserved	RO	0x0	Reserved field.

SELINKDNMSK - Switch Event Link Down Mask (0x0C18)

Bit Field	Field Name	Type	Default Value	Description
23:0	LINKDN	RW	0xFF_FFFF SWSticky	**Link Down.** When a bit in this field is set, the corresponding bit in the SELINKDNSTS register is masked from generating a switch event.
31:24	Reserved	RO	0x0	Reserved field.

SEFRSTSTS - Switch Event Fundamental Reset Status (0x0C1C)

Bit Field	Field Name	Type	Default Value	Description
7:0	FRST	RW1C	0x0 SWSticky	**Partition Fundamental Reset.** Each bit in this field corresponds to a switch partition. A bit in this field is set when a fundamental reset is detected in the corresponding switch partition. Bits associated with a disabled partition are never set (i.e., remain cleared)
31:8	Reserved	RO	0x0	Reserved field.

SEFRSTMSK - Switch Event Fundamental Reset Mask (0x0C20)

Bit Field	Field Name	Type	Default Value	Description
7:0	FRST	RW	0xFF SWSticky	**Partition Fundamental Reset.** When a bit in this field is set, the corresponding bit in the SEFRSTSTS register is masked from generating a switch event.
31:8	Reserved	RO	0x0	Reserved field.

Notes

SEHRSTSTS - Switch Event Hot Reset Status (0x0C24)

Bit Field	Field Name	Type	Default Value	Description
7:0	HRST	RW1C	0x0 SWSticky	**Partition Hot Reset.** Each bit in this field corresponds to a switch partition. A bit in this field is set when a hot reset is detected in the corresponding switch partition. Bits associated with a disabled partition are never set (i.e., remain cleared)
31:8	Reserved	RO	0x0	Reserved field.

SEHRSTMSK - Switch Event Hot Reset Mask (0x0C28)

Bit Field	Field Name	Type	Default Value	Description
7:0	HRST	RW	0xFF SWSticky	**Partition Hot Reset.** When a bit in this field is set, the corresponding bit in the SEHRSTSTS register is masked from generating a switch event.
31:8	Reserved	RO	0x0	Reserved field.

SEFOVRMSK - Switch Event Failover Mask (0x0C2C)

Bit Field	Field Name	Type	Default Value	Description
0	FCAP0FNCI	RW	0x1 SWSticky	**Failover Capability 0 Failover Mode Change Initiated Mask.** When this bit is set, the Failover Mode Change Initiated (FMCI) bit in the Failover Capability 0 Status (FCAP0STS) register is masked from generating a switch event.
1	FCAP1FNCI	RW	0x1 SWSticky	**Failover Capability 1 Failover Mode Change Initiated Mask.** When this bit is set, the Failover Mode Change Initiated (FMCI) bit in the Failover Capability 1 Status (FCAP1STS) register is masked from generating a switch event.
2	FCAP2FNCI	RW	0x1 SWSticky	**Failover Capability 2 Failover Mode Change Initiated Mask.** When this bit is set, the Failover Mode Change Initiated (FMCI) bit in the Failover Capability 2 Status (FCAP2STS) register is masked from generating a switch event.
3	FCAP3FNCI	RW	0x1 SWSticky	**Failover Capability 3 Failover Mode Change Initiated Mask** When this bit is set, the Failover Mode Change Initiated (FMCI) bit in the Failover Capability 3 Status (FCAP3STS) register is masked from generating a switch event.
15:4	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
16	FCAP0FNCC	RW	0x1 SWSticky	**Failover Capability 0 Failover Mode Change Completed Mask.** When this bit is set, the Failover Mode Change Completed (FMCC) bit in the Failover Capability 0 Status (FCAP0STS) register is masked from generating a switch event.
17	FCAP1FNCC	RW	0x1 SWSticky	**Failover Capability 1 Failover Mode Change Completed Mask.** When this bit is set, the Failover Mode Change Initiated (FMCC) bit in the Failover Capability 1 Status (FCAP1STS) register is masked from generating a switch event.
18	FCAP2FNCC	RW	0x1 SWSticky	**Failover Capability 2 Failover Mode Change Completed Mask.** When this bit is set, the Failover Mode Change Initiated (FMCC) bit in the Failover Capability 2 Status (FCAP2STS) register is masked from generating a switch event.
19	FCAP3FNCC	RW	0x1 SWSticky	**Failover Capability 3 Failover Mode Change Completed Mask.** When this bit is set, the Failover Mode Change Initiated (FMCC) bit in the Failover Capability 3 Status (FCAP3STS) register is masked from generating a switch event.
31:20	Reserved	RO	0x0	Reserved field.

SEGSIGSTS - Switch Event Global Signal Status (0x0C30)

Bit Field	Field Name	Type	Default Value	Description
7:0	GSIGNAL	RW1C	0x0 SWSticky	**Global Signal.** Each bit in this field corresponds to a switch partition. A bit in this field is set when a global signal is generated from the corresponding partition. Bits associated with a disabled partition are never set (i.e., remain cleared)
31:8	Reserved	RO	0x0	Reserved field.

SEGSIGMSK - Switch Event Global Signal Mask (0x0C34)

Bit Field	Field Name	Type	Default Value	Description
7:0	GSIGNAL	RW	0xFF SWSticky	**Global Signal.** When a bit in this field is set, the corresponding bit in the SEGSIGSTS register is masked from generating a switch event.
31:8	Reserved	RO	0x0	Reserved field.

Notes

Global Doorbells and Message Registers

GDBELLSTS - NT Global Doorbell Status (0x0C3C)

Bit Field	Field Name	Type	Default Value	Description
31:0	GDBELLSTS	RO	0x0	**Global Doorbell Status.** Each bit in this field corresponds to one of the 32 global doorbells. A bit in this field is set when there exists a corresponding unmasked pending inbound doorbell request.

GODBELLMSK[31:0] - NT Global Outbound Doorbell Mask [31:0]

Bit Field	Field Name	Type	Default Value	Description
7:0	GODBELLMSK	RW	0x0 SWSticky	**Global Outbound Doorbell Mask.** Each bit in this field corresponds to a partition. When a bit in this field is set, the outbound doorbell corresponding to this register from the partition associated with the bit is masked from affecting the state of the global doorbell status bit.
31:8	Reserved	RO	0x0	Reserved field.

GIDBELLMSK[31:0] - NT Global Inbound Doorbell Mask [31:0]

Bit Field	Field Name	Type	Default Value	Description
7:0	GIDBELLMSK	RW	0x0 SWSticky	**Global Inbound Doorbell Mask.** Each bit in this field corresponds to a partition. When a bit in this field is set, the global doorbell corresponding to this register is masked from affecting the state of the inbound doorbell in the partition associated with this bit.
31:8	Reserved	RO	0x0	Reserved field.

Notes

SWP[7:0]MSGCTL[3:0] - Switch Partition x Message Control [3:0]

Bit Field	Field Name	Type	Default Value	Description
1:0	REG	RW	0x0 SWSticky	**Register Select.** This field selects the Inbound Message (INMSG) register number to which the Outbound Message (OUTMSG) register associated with this register maps. SWPxMSGCTLy corresponds to Outbound Message (OUTMSGy) register y in partition x.
3:2	Reserved	RO	0x0	Reserved field.
6:4	PART	RW	0x0 SWSticky	**Partition Select.** This field selects the partition to which the Outbound Message (OUTMSG) register associated with this register maps. SWPxMSGCTLy corresponds to Outbound Message (OUTMSGy) register y in partition x.
31:7	Reserved	RO	0x0	Reserved field.

SerDes Control and Status Registers

Please refer to Chapter 8 for a details on programming SerDes controls. Note that in order to program the SerDes controls for a given port, it is necessary to identify which SerDes block is associated with the port. Refer to section SerDes Numbering and Port Association on page 8-1 for details.

Notes

S[7:0]CTL- SerDes x Control

Bit Field	Field Name	Type	Default Value	Description
4:0	LANESEL	RW	0x10 SWSticky	**Lane Select.** This field selects the lane on which the SerDes lane control registers (S[x]TXLCTL0, S[x]TXLCTL1, S[x]RXLCTL, and S[x]RXEQLCTL) operate when written. 0x0 - Operate on lane 0 only 0x1 - Operate on lane 1 only 0x2 - Operate on lane 2 only 0x3 - Operate on lane 3 only 0x10 - Operate on all lanes simultaneously Others - Reserved For example, when LANESEL=0x0, configuration writes to the above listed registers affect lane 0 of the SerDes only. When LANESEL=0x10, the settings in the SerDes lane control registers are applied to all lanes simultaneously. Read operations are not affected by this field (i.e., reading from a SerDes lane control register returns the last value written to that register, regardless of the setting of this field). Operating on a reserved lane results in undefined consequences.
5	POWERDN	RW	0x0 SWSticky	**SerDes Power-Down.** When this bit is set, the SerDes is placed in a deep low-power state (i.e., the SerDes lanes are placed in P2 and the CMU is powered-down). In addition, the PHY LTSSM in the corresponding port(s) is immediately transitioned to the Detect state. When this bit is cleared, the SerDes is powered-on, initialized, and the PHY LTSSM initiates link training. This bit has no effect when the SerDes is already powered-down (e.g., the SerDes quad associated with a disabled port). When a SerDes is powered-down, the serial Tx/Rx pins and reference resistor pins may be left unconnected. Refer to section SerDes Power Management on page 8-14 for further details on SerDes power management.
31:6	Reserved	RO	0x0	Reserved field.

Notes

S[7:0]TXLCTL0 - SerDes x Transmitter Lane Control 0

Bit Field	Field Name	Type	Default Value	Description
1:0	FDC_FS3DBG1	RW	0x2 SWSticky	**Transmit Driver Fine De-emphasis Control for Full Swing Mode with -3.5dB in Gen 1.** This field provides fine level control of the transmit driver de-emphasis level in full-swing mode and Gen 1 data rate (i.e., 2.5 GT/s). Note that when operating in Gen 1 data rate, the de-emphasis should nominally be set to -3.5dB of the transmit driver voltage level. This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
3:2	FDC_FS3DBG2	RW	0x3 SWSticky	**Transmit Driver Fine De-emphasis Control for Full Swing Mode with -3.5dB in Gen 2.** This field provides fine level control of the transmit driver de-emphasis level in Gen 2 mode, when the SDE field in the associated port's PCIELCTL2 register is set to -3.5dB de-emphasis. This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
5:4	FDC_FS6DBG2	RW	0x2 SWSticky	**Transmit Driver Fine De-emphasis Control for Full Swing Mode with -6.0dB in Gen 2.** This field provides fine level control of the transmit driver de-emphasis level in Gen 2 mode, when the SDE field in the associated port's PCIELCTL2 register is set to -6.0dB de-emphasis. This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.

Notes

Bit Field	Field Name	Type	Default Value	Description
17:6	Reserved	RO	0x0	Reserved field.
20:18	TX_SLEW_G1	RW	0x1 SWSticky	**Transmit Driver Slew Adjustment in Gen 1.** This field controls the output driver's slew rate at Gen 1 data-rate, for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register. This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). 0x0 - 58 ps 0x1 - 98 ps Others - Reserved
22:21	Reserved	RO	0x0	Reserved field.
25:23	TX_SLEW_G2	RW	0x0 SWSticky	**Transmit Driver Slew Adjustment in Gen 2.** This field controls the output driver's slew rate at Gen 2 data-rate, for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register. This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). 0x0 - 58 ps 0x1 - 98 ps Others - Reserved
26	Reserved	RO	0x0	Reserved field.
28:27	TX_AMPBOOST	RW	0x0 SWSticky	**Transmit Driver Amplitude Boost.** This field increases the transmitter driver's differential swing for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register. 0x0 - No amplitude boost. 0x1 - Boost amplitude by ~5%. Other - Reserved Note that setting amplitude boost to 0x1 will also increase the power consumed by the affected lanes by ~2% over the previous setting. This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register).
31:29	Reserved	RO	0x0	Reserved field.

Notes

S[7:0]TXLCTL1 - SerDes x Transmitter Lane Control 1

Bit Field	Field Name	Type	Default Value	Description
4:0	TDVL_FS3DBG1	RW	0x12 SWSticky	**Transmit Driver Voltage Level for Full-Swing Mode with -3.5dB De-emphasis in Gen 1.** This field controls the SerDes transmit driver voltage level in full-swing mode and Gen 1 data rate (i.e., 2.5 GT/s). The value of this field corresponds to the peak-to-peak differential voltage at the transmitter pins, prior to de-emphasis being applied. This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
7:5	CDC_FS3DBG1	RW	0x3 SWSticky	**Transmit Driver Coarse De-Emphasis Control for Full Swing mode in Gen 1.** This field provides coarse level control of the transmit driver de-emphasis level in full-swing mode and Gen 1 data rate (i.e., 2.5 GT/s). This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Note that when operating in Gen 1 data rate, the de-emphasis should nominally be set to -3.5dB of the transmit driver voltage level. Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
12:8	TDVL_FS3DBG2	RW	0x18 SWSticky	**Transmit Driver Voltage Level for Full-Swing Mode with -3.5dB De-emphasis in Gen 2.** This field controls the SerDes transmit driver voltage level in full-swing mode and Gen 2 data rate, when the SDE field in the associated port's PCIELCTL2 register is set to -3.5dB de-emphasis. The value of this field corresponds to the peak-to-peak differential voltage at the transmitter pins, prior to de-emphasis being applied. This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:13	CDC_FS3DBG2	RW	0x3 SWSticky	**Transmit Driver Coarse De-Emphasis Control for Full Swing mode with -3.5dB in Gen 2.** This field provides coarse level control of the transmit driver de-emphasis level in Gen 2 mode, when the SDE field in the associated port's PCIELCTL2 register is set to -3.5dB de-emphasis. This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
20:16	TDVL_FS6DBG2	RW	0x15 SWSticky	**Transmit Driver Voltage Level for Full-Swing Mode with -6.0dB De-emphasis in Gen 2.** This field controls the SerDes transmit driver voltage level in full-swing mode and Gen 2 data rate, when the SDE field in the associated port's PCIELCTL2 register is set to -6.0dB de-emphasis. The value of this field corresponds to the peak-to-peak differential voltage at the transmitter pins, prior to de-emphasis being applied. This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.
23:21	CDC_FS6DBG2	RW	0x6 SWSticky	**Transmit Driver Coarse De-Emphasis Control for Full Swing Mode with -6.0dB in Gen 2.** This field provides coarse level control of the transmit driver de-emphasis level in Gen 2 mode, when the SDE field in the associated port's PCIELCTL2 register is set to -6.0dB de-emphasis. This field has no effect when the port operates in low-swing mode (i.e., de-emphasis is turned-off in this mode). This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Programmable Voltage Margining and De-Emphasis on page 8-4 for further details on programming this field.

Notes

Bit Field	Field Name	Type	Default Value	Description
27:24	TDVL_LSG1	RW	0xA SWSticky	**Transmit Driver Voltage Level for Low-Swing Mode in Gen 1.** This field controls the SerDes transmit driver voltage level when the associated port operates in low-swing mode (i.e., the LSE bit in the port's SERDESCFG register is set to 0x1) and Gen 1 data rate. The value of this field corresponds to the peak-to-peak differential voltage at the transmitter pins. This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Low-Swing Transmitter Voltage Mode on page 8-12 for further details on programming this field.
31:28	TDVL_LSG2	RW	0xC SWSticky	**Transmit Driver Voltage Level for Low-Swing Mode in Gen 2.** This field controls the SerDes transmit driver voltage level when the associated port operates in low-swing mode (i.e., the LSE bit in the port's SERDESCFG register is set to 0x1) and Gen 2 data rate. The value of this field corresponds to the peak-to-peak differential voltage at the transmitter pins. This field controls the voltage level for the lane(s) selected by the Lane Select (LANESEL[3:0]) field in the SerDes Control (S[x]CTL) register.This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register). Refer to section Low-Swing Transmitter Voltage Mode on page 8-12 for further details on programming this field.

Notes

S[7:0]RXEQLCTL - SerDes x Receiver Equalization Lane Control

Bit Field	Field Name	Type	Default Value	Description
2:0	RXEQZ	RW	0x1 SWSticky	**Receiver Equalization Zero.** Amplifies the high-frequency gain of the equalizer. Setting both RXEQZ and RXEQB to zero results in turning off the receiver equalization completely. An RXEQZ value of 0x7 results in the highest amount of high frequency gain. Together with the RXEQB default value, the RXEQZ default value corresponds to a long, lossy channel. Refer to section Receiver Equalization Controls on page 8-14 for further information of Receiver Equalization. This field controls the receiver equalization for the lane(s) selected by the Lane Select (LANESEL) field in the SerDes Control (S[x]CTL) register. This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register).
5:3	RXEQB	RW	0x7 SWSticky	**Receive Equalization Boost.** Reduces the low-frequency gain of the equalizer. Setting both RXEQZ and RXEQB to zero results in turning off the receiver equalization completely. An RXEQB value of 0x7 results in the smallest low frequency gain and largest amount of boost. Together with the RXEQZ default value, the RXEQB default value corresponds to a long, lossy channel. Refer to section Receiver Equalization Controls on page 8-14 for further information of Receiver Equalization. This field controls the receiver equalization for the lane(s) selected by the Lane Select (LANESEL) field in the SerDes Control (S[x]CTL) register. This value is SWSticky for all lanes (i.e., even those not selected by the LANESEL field in the S[x]CTL register).
31:6	Reserved	RO	0x0	Reserved field.

General Purpose I/O Registers

GPIOFUNC - General Purpose I/O Function (0x116C)

Bit Field	Field Name	Type	Default Value	Description
8:0	GPIOFUNC	RW	0x0 SWSticky	**GPIO Function.** Each bit in this field controls the corresponding GPIO pin. When set, the corresponding GPIO pin operates as the selected alternate function. When a bit is cleared, the corresponding GPIO pin operates as a general purpose I/O pin. Bit x in this field corresponds to GPIO pin x.
31:9	Reserved	RO	0x0	Reserved field.

Notes

GPIOAFSEL - General Purpose I/O Alternate Function Select (0x1170)

Bit Field	Field Name	Type	Default Value	Description
1:0	AFSEL0	RW	0x0 SWSticky	**GPIO Pin 0 Alternate Function Select.** This field selects the alternate function associated with the corresponding GPIO pin when the GPIO pin is configured to operate as an alternate function. See Chapter 13 and Table 13.2 for details. 0x0 - Alternate function 0 0x1 - Alternate function 1 Field AFSELx corresponds to GPIO pin x.
3:2	AFSEL1	RW	0x0 SWSticky	**GPIO Pin 1 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
5:4	AFSEL2	RW	0x0 SWSticky	**GPIO Pin 2 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
7:6	AFSEL3	RW	0x0 SWSticky	**GPIO Pin 3 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
9:8	AFSEL4	RW	0x0 SWSticky	**GPIO Pin 4 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
11:10	AFSEL5	RW	0x0 SWSticky	**GPIO Pin 5 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
13:12	AFSEL6	RW	0x0 SWSticky	**GPIO Pin 6 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
15:14	AFSEL7	RW	0x0 SWSticky	**GPIO Pin 7 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
17:16	AFSEL8	RW	0x0 SWSticky	**GPIO Pin 8 Alternate Function Select.** See AFSEL0 field description in the GPIOAFSEL0 register.
31:18	Reserved	RO	0x0	Reserved field.

GPIOCFG - General Purpose I/O Configuration (0x1174)

Bit Field	Field Name	Type	Default Value	Description
8:0	GPIOCFG	RW	0x0 SWSticky	**GPIO Configuration.** Each bit in this field controls the corresponding GPIO pin. When a bit is configured as a general purpose I/O pin and the corresponding bit in this field is set, then the pin is configured as a GPIO output. When a bit is configured as a general purpose I/O pin and the corresponding bit in this field is cleared, then the pin is configured as an input. When the pin is configured as an alternate function, the behavior of the pin is defined by the alternate function. Bit x in this field corresponds to GPIO pin x.
31:9	Reserved	RO	0x0	Reserved field.

Notes

GPIOD - General Purpose I/O Data (0x1178)

Bit Field	Field Name	Type	Default Value	Description
8:0	GPIOD	RW	HWINIT SWSticky	**GPIO Data.** Each bit in this field controls the corresponding GPIO pin. Reading this field returns the current value of each GPIO pin regardless of GPIO pin mode (i.e., alternate function or GPIO pin). Writing a value to this field causes the corresponding pins which are configured as GPIO outputs to change state to the value written. Bit x in this field corresponds to GPIO pin x.
31:9	Reserved	RO	0x0	Reserved field.

Hot-Plug and SMBus Interface Registers

HPCFGCTL - Hot-Plug Configuration Control (0x117C)

Bit Field	Field Name	Type	Default Value	Description
0	IPXAPN	RW	0x0 SWSticky	**Invert Polarity of PxAPN.** When this bit is set, the polarity of the PxAPN input is inverted in all ports.
1	IPXPDN	RW	0x0 SWSticky	**Invert Polarity of PxPDN.** When this bit is set, the polarity of the PxPDN input is inverted in all ports.
2	IPXPFN	RW	0x0 SWSticky	**Invert Polarity of PxPFN.** When this bit is set, the polarity of the PxPFN input is inverted in all ports.
3	IPXMRLN	RW	0x0 SWSticky	**Invert Polarity of PxMRLN.** When this bit is set, the polarity of the PxMRLN input is inverted in all ports.
4	IPXAIN	RW	0x0 SWSticky	**Invert Polarity of PxAIN.** When this bit is set, the polarity of the PxAIN output is inverted in all ports.
5	IPXPIN	RW	0x0 SWSticky	**Invert Polarity of PxPIN.** When this bit is set, the polarity of the PxPIN output is inverted in all ports.
6	IPXPEP	RW	0x0 SWSticky	**Invert Polarity of PxPEP.** When this bit is set, the polarity of the PxPEP output is inverted in all ports.
7	IPXILOCKP	RW	0x0 SWSticky	**Invert Polarity of PxILOCKP.** When this bit is set, the polarity of the PxILOCKP output is inverted in all ports.
8	IPXPWRGDN	RW	0x0 SWSticky	**Invert Polarity of PxPWRGDN.** When this bit is set, the polarity of the PxPWRGDN input is inverted in all ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
10:9	PDETECT	RW	0x0 SWSticky	**Presence Detect Control.** This field controls the manner in which presence of an adapter in a slot is reported to the hot-plug controller associated with a downstream switch port. 0x0 - (both) Presence of an adapter in the slot is reported as the logical "OR" of the receiver detect mechanism selected by the RDETECT field in the PHYLCFG0 register and the hot-plug presence detect input (PxPDN). 0x1 - (signal) Presence of an adapter in the slot is reported as the state of the hot-plug presence detect input (PxPDN). 0x2 - (always) When selected this mode always informs the hot-plug controller that an adapter is present. 0x3 - (never) When selected this mode always informs the hot-plug controller that an adapter is not present.
11	MRLPWROFF	RW	0x1 SWSticky	**MRL Automatic Power Off.** When this bit is set and the Manual Retention Latch Present (MRLP) bit is set in the PCI Express Slot Capability (PCIESCAP) register, then power to the slot is automatically turned off when the MRL sensor indicates that the MRL is open. This occurs regardless of the state of the Power Controller Control (PCC) bit in the PCI Express Slot Control (PCIESCTL) register.
12	IPXILOCKST	RW	0x0 SWSticky	**Invert Polarity of PxILOCKST.** When this bit is set, the polarity of the PxILOCKST input is inverted in all ports.
13	TEMICTL	RW	0x0 SWSticky	**Toggle Electromechanical Interlock Control.** When this bit is cleared, the Electromechanical Interlock (PxILOCKP) output is pulsed for at least 100 ms and at most 150 ms when a one is written to the EIC bit in the PCIESCTL register. When this bit is set, writing a one to the EIC register inverts the state of the PxILOCKP output (i.e., the state of the PxILOCKP signal is imply inverted and not pulsed).

Notes

Bit Field	Field Name	Type	Default Value	Description
15:14	RSTMODE	RW	0x0 SWSticky	**Reset Mode.** This field controls the manner in which port reset outputs are generated. 0x0 - (pec) Power enable controlled reset output 0x1 - (pgc) Power good controlled reset output 0x2 - Reserved 0x3 - Reserved
23:16	PWR2RST	RW	0x14 SWSticky	**Slot Power to Reset Negation.** This field contains the delay from stable downstream switch port power to negation of the downstream switch port reset in units of 10 mS. A value of zero corresponds to no delay. This field may be used to meet the T_{PCPERL} specification. The default value corresponds to 200 mS.
31:24	RST2PWR	RW	0x14 SWSticky	**Reset Negation.** This field contains the delay from negation of a downstream switch port's reset to disabling of a downstream switch port's power in units of 10 mS. A value of zero corresponds to no delay. The default value corresponds to 200 mS.

SMBUSSTS - SMBus Status (0x1188)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	SSMBADDR	RO	HWINIT	**Slave SMBus Address.** This field contains the SMBus address assigned to the slave SMBus interface. Refer to section Initialization on page 12-19.
8	Reserved	RO	0x0	Reserved field.
15:9	MSMBADDR	RO	HWINIT	**Master SMBus Address.** This field contains the SMBus address issued by the master SMBus interface. Refer to section Serial EEPROM on page 12-2.
19:16	Reserved	RO	0x0	Reserved field.
20	EED	RW1C	0x0 SWSticky	**EEPROM Error Detected.** This bit is set when an error is detected by the master SMBus interface during serial EEPROM initialization. The occurrence of such an error causes the EEPROM loading to be aborted and the RSTHALT bit to be set in the SWCTL register. When a known error is detected, this bit is set in conjunction with another bit in this register that indicates the type of error (e.g., Blank Serial EEPROM, Initialization Checksum Error, etc.) When an unknown error is detected, only this bit is set.

Notes

Bit Field	Field Name	Type	Default Value	Description
21	ICB	RW1C	0x0 SWSticky	**Invalid Configuration Block.** This bit is set when the master SMBus interface detects an invalid configuration block during serial EEPROM initialization. The valid configuration blocks are: - Single double-word initialization sequence - Sequential double-word initialization sequence - Jump block - Wait block - Configuration done sequence
22	BLANK	RW1C	0x0 SWSticky	**Blank Serial EEPROM.** When the switch is configured to operate in a mode in which serial EEPROM initialization occurs during a Switch Fundamental Reset, this bit is set when a blank serial EEPROM is detected. This is not considered an error.
23	ROLLOVER	RW1C	0x0 SWSticky	**Serial EEPROM Rollover.** When the switch is configured to operate in a mode in which serial EEPROM initialization occurs during a Switch Fundamental Reset, this bit is set when a Serial EEPROM address rollover error is detected.
24	EEPROM-DONE	RW1C	0x0 SWSticky	**Serial EEPROM Initialization Done.** When the switch is configured to operate in a mode in which serial EEPROM initialization occurs during a Switch Fundamental Reset, this bit is set when serial EEPROM initialization completes or is aborted.
25	NAERR	RW1C	0x0 SWSticky	**No Acknowledge Error.** This bit is set if an unexpected NACK is observed during a master SMBus transaction. The setting of this bit may indicate the following: that the addressed device does not exist on the SMBus (i.e., addressing error); data is unavailable or the device is busy; an invalid command was detected by the slave; or invalid data was detected by the slave.
26	LAERR	RW1C	0x0	**Lost Arbitration Error.** When the master SMBus interface loses arbitration for the SMBus, it automatically re-arbitrates for the SMBus. If the master SMBus interface loses 16 consecutive arbitration attempts, then the transaction is aborted and this bit is set.
27	OTHERERR	RW1C	0x0	**Other Error.** This bit is set if a misplaced START or STOP condition is detected by the master SMBus interface, or when some other unforeseen error condition is detected.
28	ICSERR	RW1C	0x0	**Initialization Checksum Error.** This bit is set if an invalid checksum is computed during Serial EEPROM initialization or when a configuration done command is not found in the serial EEPROM.

Notes

Bit Field	Field Name	Type	Default Value	Description
29	URA	RW1C	0x0 SWSticky	**Unmapped Register Error.** This bit is set if an attempt is made to access via serial EEPROM a register that is not defined in the global address space. This is not considered an error.
30	Reserved	RO	0x0	Reserved field.
31	WCBTO	RW1C	0x0 SWSticky	**Wait Configuration Block Timeout.** This bit is set when a timeout is detected while executing a Wait configuration block during EEPROM loading.

SMBUSCTL - SMBus Control (0x118C)

Bit Field	Field Name	Type	Default Value	Description
15:0	MSMBCP	RW	HWINIT SWSticky	**Master SMBus Clock Prescalar.** This field contains a clock prescalar value used during master SMBus transactions. The prescalar clock period is equal to 32 ns multiplied by the value in this field. When the field is cleared to zero or one, the clock is stopped. The initial value of this field is 0x0053[1], indicating that the master SMBus clock prescalar is configured to operate in fast mode (i.e., 400 KHz).
16	MSMBIOM	RW	0x0 SWSticky	**De-featured.** This field is not applicable to this device. Setting this field results in an undefined operation.
17	ICHECKSUM	RW	0x0 SWSticky	**Ignore Checksum Errors.** When this bit is set, serial EEPROM initialization checksum errors are ignored (i.e., the checksum always passes).
19:18	SSMBMODE	RW	0x0 SWSticky	**Slave SMBus Mode.** The slave SMBus contains internal glitch counters on the SSMBCLK and SSMBDAT signals that wait approximately 1uS before sampling or driving these signals. This field allows the glitch counter time to be reduced or entirely removed. In some systems, this may permit high speed slave SMBus operation. 0x0 - (normal) Slave SMBus normal mode. Glitch counters operate with 1uS delay. 0x1 - (fast) Slave SMBus interface fast mode. Glitch counters operate with 100nS delay. 0x2 - (disabled) Slave SMBus interface with glitch counters disabled. Glitch counters operate with zero delay which effectively removes them. 0x3 - reserved.

Notes

Bit Field	Field Name	Type	Default Value	Description
21:20	MSMBMODE	RW	0x0 SWSticky	**Master SMBus Mode.** The master SMBus contains internal glitch counters on the MSMBCLK and MSMBDAT signals that wait approximately 1uS before sampling or driving these signals. This field allows the glitch counter time to be reduced or entirely removed. In some systems, this may permit high speed master SMBus operation. 0x0 - (normal) Master SMBus normal mode. Glitch counters operate with 1uS delay. 0x1 - (fast) Master SMBus interface fast mode. Glitch counters operate with 100nS delay. 0x2 - (disabled) Master SMBus interface with glitch counters disabled. Glitch counters operate with zero delay which effectively removes them. 0x3 - reserved.
22	SMBDTO	RW	0x0 SWSticky	**SMBus Disable Time-out.** When this bit is set, SMBus timeouts are disabled on the master and slave SMBuses.
25:23	WCBT	RW	0x0 SWSticky	**Wait Configuration Block Timeout.** This field controls the timeout value for the SMBus Master interface when executing a Wait Configuration block during EEPROM loading. Refer to section Initialization from Serial EEPROM on page 12-3 for details. A value of zero indicates an infinite wait time. 0x0 - Infinite wait time 0x1 - 1 us 0x2 - 5 us 0x3 - 10 ms 0x4 - 100 ms Others - Reserved
31:26	Reserved	RO	0x0	Reserved field.

1. The MSMBCLK low minimum pulse width is equal to half the period programmed in this field. The value of 0x53, which corresponds to ~373 KHz, allows the min low pulse width to be satisfied. In systems where this timing parameter is not critical, the operating frequency may be increased.

Notes

SMBUSCBHL - SMBus Configuration Block Header Log (0x11E8)

Bit Field	Field Name	Type	Default Value	Description
7:0	BYTE0	RO	0x0	**Configuration Block Byte 0.** This field contains byte 0 of the last serial EEPROM configuration block processed normally by the SMBus master interface. Refer to section Initialization from Serial EEPROM on page 12-3 for details on serial EEPROM configuration blocks. This register is meant for debugging purposes.
15:8	BYTE1	RO	0x0	**Configuration Block Byte 1.** This field contains byte 1 of the last serial EEPROM configuration block processed normally by the SMBus master interface. If the configuration block has less than 2 bytes, this field is undefined. Refer to section Initialization from Serial EEPROM on page 12-3 for details on serial EEPROM configuration blocks. This register is meant for debugging purposes.
23:16	BYTE2	RO	0x0	**Configuration Block Byte 2.** This field contains byte 2 of the last serial EEPROM configuration block processed normally by the SMBus master interface. If the configuration block has less than 3 bytes, this field is undefined. Refer to section Initialization from Serial EEPROM on page 12-3 for details on serial EEPROM configuration blocks. This register is meant for debugging purposes.
31:24	Reserved	RO	0x0	Reserved field.

Notes

EEPROMINTF - Serial EEPROM Interface (0x1190)

Bit Field	Field Name	Type	Default Value	Description
15:0	ADDR	RW	0x0 SWSticky	**EEPROM Address.** This field contains the byte address in the Serial EEPROM to be read or written.
23:16	DATA	RW	0x0 SWSticky	**EEPROM Data.** A write to this field will initiates a serial EEPROM read or write operation, as selected by the OP field, to the address specified in the ADDR field. When a write operation is selected, the value written to this field is the value written to the serial EEPROM. When a read operation is selected, the value written to this field is ignored and the value read from the serial EEPROM may be read from this field when the DONE bit is set.
24	BUSY	RO	0x0	**EEPROM Busy.** This bit is set when a serial EEPROM read or write operation is in progress. 0x0 - (idle) serial EEPROM interface idle 0x1 - (busy) serial EEPROM interface operation in progress
25	DONE	RW1C	0x0 SWSticky	**EEPROM Operation Completed.** This bit is set when a serial EEPROM operation has completed. 0x0 - (notdone) interface is idle or operation in progress 0x1 - (done) operation completed
26	OP	RW	0x0 SWSticky	**EEPROM Operation Select.** This field selects the type of EEPROM operation to be performed when the DATA field is written 0x0 - (write) serial EEPROM write 0x1 - (read) serial EEPROM read
31:27	Reserved	RO	0x0	Reserved field.

IOEXPADDR0 - SMBus I/O Expander Address 0 (0x1198)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE0ADDR	RWL	0x0 SWSticky	**I/O Expander 0 Address.** This field contains the SMBus address assigned to I/O expander 0 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE1ADDR	RWL	0x0 SWSticky	**I/O Expander 1 Address.** This field contains the SMBus address assigned to I/O expander 1 on the master SMBus interface.
16	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
23:17	IOE2ADDR	RWL	0x0 SWSticky	**I/O Expander 2 Address.** This field contains the SMBus address assigned to I/O expander 2 on the master SMBus interface.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE3ADDR	RWL	0x0 SWSticky	**I/O Expander 3 Address.** This field contains the SMBus address assigned to I/O expander 3 on the master SMBus interface.

IOEXPADDR1 - SMBus I/O Expander Address 1 (0x119C)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE4ADDR	RWL	0x0 SWSticky	**I/O Expander 4 Address.** This field is unused.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE5ADDR	RWL	0x0 SWSticky	**I/O Expander 5 Address.** This field is unused.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE6ADDR	RWL	0x0 SWSticky	**I/O Expander 6 Address.** This field is unused.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE7ADDR	RWL	0x0 SWSticky	**I/O Expander 7 Address.** This field is unused.

IOEXPADDR2 - SMBus I/O Expander Address 2 (0x11A0)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE8ADDR	RWL	0x0 SWSticky	**I/O Expander 8 Address.** This field is unused.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE9ADDR	RWL	0x0 SWSticky	**I/O Expander 9 Address.** This field is unused.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE10ADDR	RWL	0x0 SWSticky	**I/O Expander 10 Address.** This field is unused.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE11ADDR	RWL	0x0 SWSticky	**I/O Expander 11 Address.** This field is unused.

Notes

IOEXPADDR3 - SMBus I/O Expander Address 3 (0x11A4)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE12ADDR	RWL	0x0 SWSticky	**I/O Expander 12 Address.** This field contains the SMBus address assigned to I/O expander 12 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE13ADDR	RWL	0x0 SWSticky	**I/O Expander 13 Address.** This field contains the SMBus address assigned to I/O expander 13 on the master SMBus interface.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE14ADDR	RWL	0x0 SWSticky	**I/O Expander 14 Address.** This field contains the SMBus address assigned to I/O expander 14 on the master SMBus interface.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE15ADDR	RWL	0x0 SWSticky	**I/O Expander 15 Address.** This field is unused.

IOEXPADDR4 - SMBus I/O Expander Address 4 (0x11A8)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE16ADDR	RWL	0x0 SWSticky	**I/O Expander 16 Address.** This field is unused.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE17ADDR	RWL	0x0 SWSticky	**I/O Expander 17 Address.** This field contains the SMBus address assigned to I/O expander 17 on the master SMBus interface.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE18ADDR	RWL	0x0 SWSticky	**I/O Expander 18 Address.** This field contains the SMBus address assigned to I/O expander 18 on the master SMBus interface.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE19ADDR	RWL	0x0 SWSticky	**I/O Expander 19 Address.** This field contains the SMBus address assigned to I/O expander 19 on the master SMBus interface.

Notes

IOEXPADDR5 - SMBus I/O Expander Address 5 (0x11AC)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE20ADDR	RWL	0x0 SWSticky	**I/O Expander 20 Address.** This field contains the SMBus address assigned to I/O expander 20 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE21ADDR	RWL	0x0 SWSticky	**I/O Expander 21 Address.** This field is unused.
31:16	Reserved	RO	0x0	Reserved field.

GPECTL - General Purpose Event Control (0x11B0)

Bit Field	Field Name	Type	Default Value	Description
23:0	GPEE	RW	0x0 SWSticky	**General Purpose Event Enable.** Each bit in this field corresponds to a port. When a bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCI Express Base Specification Rev 2.1 are disabled for that port and are instead signaled through General Purpose Event (GPEN) signal assertions. GPEN is a GPIO alternate function.
30:24	Reserved	RO	0x0	Reserved field.
31	IGPE	RW	0x0 SWSticky	**Invert General Purpose Event Enable Signal Polarity.** When this bit is set, the polarity of all General Purpose Event (GPEN) signals is inverted. 0x0 -(normal) GPEN signals are active low 0x1 -(invert) GPEN signals are active high

GPESTS - General Purpose Event Status (0x11B4)

Bit Field	Field Name	Type	Default Value	Description
23:0	GPES	RO	0x0	**General Purpose Event Status.** Each bit in this field corresponds to a port. When a bit is set, the corresponding port is signaling a general purpose event by asserting the GPEN signal. This bit is never set if the corresponding general purpose event is not enabled in the port's GPECTL register.
31:24	Reserved	RO	0x0	Reserved field.

Notes

Temperature Sensor Registers

TMPCTL - Temperature Sensor Control (0x11D4)

Bit Field	Field Name	Type	Default Value	Description
7:0	LTH	RW	0x0 SWSticky	**Low Temperature Threshold.** This field contains the low temperature threshold. The value in this field represents a fixed-point 0:7:1 temperature in degrees Celsius (i.e., an unsigned number with 7 integer bits and 1 fractional bit).
15:8	MTH	RW	0x0 SWSticky	**Middle Temperature Threshold.** This field contains the middle temperature threshold. The value in this field represents a fixed-point 0:7:1 temperature in degrees Celsius (i.e., an unsigned number with 7 integer bits and 1 fractional bit).
23:16	HTH	RW	0x0 SWSticky	**High Temperature Threshold.** This field contains the high temperature threshold. The value in this field represents a fixed-point 0:7:1 temperature in degrees Celsius (i.e., an unsigned number with 7 integer bits and 1 fractional bit).
24	ALTH	RW	0x0 SWSticky	**Above Low Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively.
25	BHTH	RW	0x0 SWSticky	**Below High Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively.
26	BLTH	RW	0x0 SWSticky	**Below Low Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively
27	AMTH	RW	0x0 SWSticky	**Above Middle Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively.
28	BMTH	RW	0x0 SWSticky	**Below Middle Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively.

Notes

Bit Field	Field Name	Type	Default Value	Description
29	AHTH	RW	0x0 SWSticky	**Above High Temperature Threshold Interrupt Enable.** When this bit is set and the corresponding bit in the Temperature Sensor Alarm (TMPALARM) register is set, the TMPSENSOR bit is set in the P2PINTSTS and NTINTSTS registers of all upstream port PCI-to-PCI bridge and NT functions respectively.
30	Reserved	RO	0x0	Reserved field.
31	PDOWN	RW	0x1 SWSticky	**Power Down.** When this bit is set, the temperature sensor is powered down and the value reported by the temperature sensor is 0 degrees Celsius.

TMPSTS - Temperature Sensor Status (0x11D8)

Bit Field	Field Name	Type	Default Value	Description
7:0	TEMP	RO	—	**Current Temperature.** This field contains the current temperature. The value in the field represents a fixed-point 0:7:1 temperature in degrees C (i.e., an unsigned number with 7 integer bits and 1 fractional bit).
15:8	LTEMP	RO	0xFF	**Low Temperature.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
23:16	HTEMP	RO	0x0	**High Temperature.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
24	BLTH	RO	0x0	**Below Low Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
25	ALTH	RO	0x0	**Above Low Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
26	BMTH	RO	0x0	**Below Middle Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
27	AMTH	RO	0x0	**Above Middle Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
28	BHTH	RO	0x0	**Below High Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.

Notes

Bit Field	Field Name	Type	Default Value	Description
29	AHTH	RO	0x0	**Above High Temperature Threshold.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.
30	Reserved	RO	0x0	Reserved field.
31	UPDATED	RO	0x0	**Temperature Updated.** This field contains the current value of the corresponding field in the Temperature Sensor Alarm (TMPALARM) register.

TMPALARM - Temperature Sensor Alarm (0x11DC)

Bit Field	Field Name	Type	Default Value	Description
7:0	Reserved	RO	0x0	Reserved field.
15:8	LTEMP	RW	0xFF SWSticky	**Low Temperature.** This field contains the lowest temperature recorded since the field was reset. The value in the field represents a fixed-point 0:7:1 temperature in degrees C (i.e., an unsigned number with 7 integer bits and 1 fractional bit). The field may be set to any value by software by writing to this register (i.e., as a way to reset the value in between readings).
23:16	HTEMP	RW	0x0 SWSticky	**High Temperature.** This field contains the highest temperature recorded since the field was reset. The value in the field represents a fixed-point 0:7:1 temperature in degrees C (i.e., an unsigned number with 7 integer bits and 1 fractional bit). The field may be set to any value by software by writing to this register (i.e., as a way to reset the value in between readings).
24	BLTH	RW1C	0x0 SWSticky	**Below Low Temperature Threshold.** This bit is set when the current temperature is below the threshold set in the Low Temperature Threshold (LTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.
25	ALTH	RW1C	0x0 SWSticky	**Above Low Temperature Threshold.** This bit is set when the current temperature is above the threshold set in the Low Temperature Threshold (LTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.

Notes

Bit Field	Field Name	Type	Default Value	Description
26	BMTH	RW1C	0x0 SWSticky	**Below Middle Temperature Threshold.** This bit is set when the current temperature is below the threshold set in the Middle Temperature Threshold (MTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.
27	AMTH	RW1C	0x0 SWSticky	**Above Middle Temperature Threshold.** This bit is set when the current temperature is above the threshold set in the Middle Temperature Threshold (MTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.
28	BHTH	RW1C	0x0 SWSticky	**Below High Temperature Threshold.** This bit is set when the current temperature is below the threshold set in the High Temperature Threshold (HTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.
29	AHTH	RW1C	0x0 SWSticky	**Above High Temperature Threshold.** This bit is set when the current temperature is above the threshold set in the High Temperature Threshold (HTH) field in the Temperature Sensor Control (TMPCTL) register. This field is automatically cleared as a side effect of register being read.The field may be set to any value to facilitate software testing by writing to this register.
30	Reserved	RO	0x0	Reserved field.
31	UPDATED	RW1C	0x0 SWSticky	**Temperature Updated.** This bit is set when the temperature sensor has produced an updated temperature value. This value is used to update fields in this register as well as the Temperature (TEMP) field in the Temperature Sensor Status (TMPSTS) register.

Notes

TMPADJ - Temperature Sensor Adjustment (0x11E0)

Bit Field	Field Name	Type	Default Value	Description
7:0	OFFSET	RW	0xC4 SWSticky	**Offset.** Absolute temperature offset in degrees C. This two's complement value is added to the temperature value returned by the temperature sensor to produce the reported temperature. Each incremental setting in this field corresponds to an offset of 0.5 degrees C. The default value of this field corresponds to an offset of -24 degrees C.
9:8	DACSETTLE	RW	0x2 SWSticky	**D to A Converter Settling Time.** Time for each successive approximation to analog voltage. 0 - (512) 512 clocks 1 - (1024) 1024 clocks 2 - (2048) 2048 clocks 3 - (4096) 4096 clocks
11:10	Reserved	RO	0x0	Reserved field.
14:12	DACOFFSET	RW	0x0 SWSticky	**D to A Converter Offset.** Current source offset for D to A Converter (DAC).
15	Reserved	RO	0x0	Reserved field.
19:16	DACGAIN	RW	0x0 SWSticky	**D to A Converter Gain.** Current gain for PTAT measurement current.
28:20	SABV	RW	0x0 SWSticky	**Successive Approximation Bypass Value.**
29	SABE	RW	0x0 SWSticky	**Successive Approximation Bypass Enable.** When this bit is set, the D to A Converter's successive approximation algorithm is bypassed, and the value in the SABV field is used.
31:30	Reserved	RO	0x0	Reserved field.

TSSLOPE - Temperature Sensor Slope (0x11E4)

Bit Field	Field Name	Type	Default Value	Description
3:0	ADJ0	RW	0x8 SWSticky	**Slope Adjustment 0.** -40 degree adjustment.
7:4	ADJ1	RW	0x6 SWSticky	**Slope Adjustment 1.** 40 to 60 degree adjustment.
11:8	ADJ2	RW	0x4 SWSticky	**Slope Adjustment 2.** 60 to 75 degree adjustment.
15:12	ADJ3	RW	0x3 SWSticky	**Slope Adjustment 3.** 75 to 90 degree adjustment.
19:16	ADJ4	RW	0x1 SWSticky	**Slope Adjustment 4.** 90 to 105 degree adjustment.
23:20	ADJ5	RW	0x0 SWSticky	**Slope Adjustment 5.** 105 to 120 degree adjustment.

Notes

Bit Field	Field Name	Type	Default Value	Description
27:24	ADJ6	RW	0x0 SWSticky	**Slope Adjustment 6.** 120+ degree adjustment.
30:28	Reserved	RO	0x0	Reserved field.
31	ADJDOWN	RW	0x1 SWSticky	**Slope Adjustment Down.** If cleared, slope adjustment values in these register represent positive adjustments. If set, slope adjustment values in this register represent negative adjustments.

Notes

JTAG Boundary Scan

Notes

Introduction

The JTAG Boundary Scan interface provides a way to test the interconnections between integrated circuit pins after they have been assembled onto a circuit board.

There are two pin types present in the switch: AC-coupled and DC-coupled (also called AC and DC pins). IEEE 1149.1 compliant boundary scan allows testing of the DC pins. The DC pins are the “normal” pins that do not require AC-coupling. The presence of AC-coupling capacitors on some of the device pins prevents DC values from being driven between a driver and receiver. AC Boundary Scan methodology described in IEEE 1149.6, is available to provide a time-varying signal to pass through the AC-coupling when in AC test mode. The IDT device supports both of these standards.

Test Access Point

The system logic utilizes a 16-state, TAP controller, a six-bit instruction register, and five dedicated pins to perform a variety of functions. The primary use of the JTAG TAP Controller state machine is to allow the five external JTAG control pins to control and access the switch's many external signal pins. The JTAG TAP Controller can also be used for identifying the device part number. The JTAG logic of the switch is depicted in Figure 25.1.

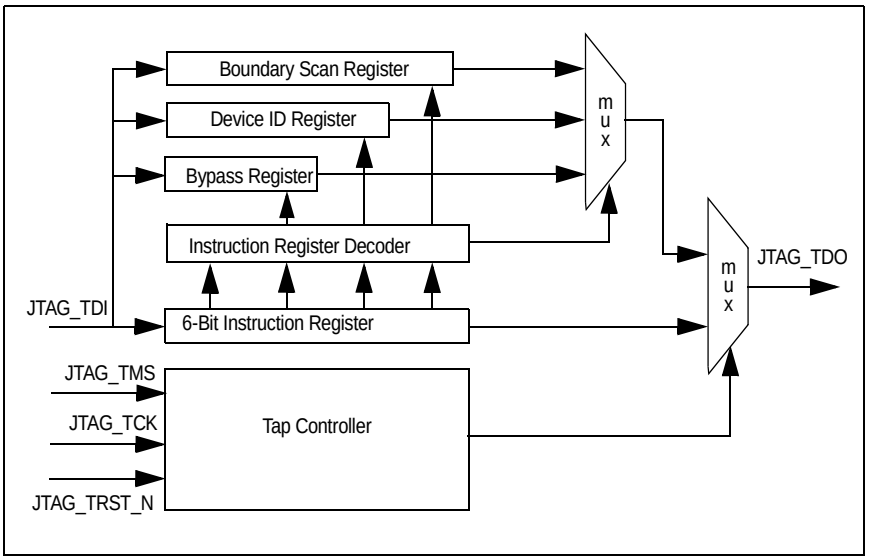

Figure 25.1 Diagram of the JTAG Logic

Refer to the IEEE 1149.1 document for an operational description of the Boundary Scan and TAP controller.

Signal Definitions

JTAG operations such as reset, state-transition control, and clock sampling are handled through the signals listed in Table 25.1. A functional overview of the TAP Controller and Boundary Scan registers is provided in the sections following the table.

Notes

Pin Name	Type	Description
JTAG_TRST_N	Input	JTAG RESET (active low) Asynchronous reset for JTAG TAP controller (internal pull-up)
JTAG_TCK	Input	JTAG Clock Test logic clock. JTAG_TMS and JTAG_TDI are sampled on the rising edge. JTAG_TDO is output on the falling edge.
JTAG_TMS	Input	JTAG Mode Select. Requires an external pull-up. Controls the state transitions for the TAP controller state machine (internal pull-up)
JTAG_TDI	Input	JTAG Input Serial data input for BSC chain, Instruction Register, IDCODE register, and BYPASS register (internal pull-up)
JTAG_TDO	Output	JTAG Output Serial data out. Tri-stated except when shifting while in Shift-DR and SHIFT-IR TAP controller states.

Table 25.1 JTAG Pin Descriptions

The TAP controller transitions from state to state, according to the value present on JTAG_TMS, as sampled on the rising edge of JTAG_TCK. The Test-Logic Reset state can be reached either by asserting JTAG_TRST_N or by applying a 1 to JTAG_TMS for five consecutive cycles of JTAG_TCK. A state diagram for the TAP controller appears in Figure 25.2. The value next to state represent the value that must be applied to JTAG_TMS on the next rising edge of JTAG_TCK, to transition in the direction of the associated arrow.

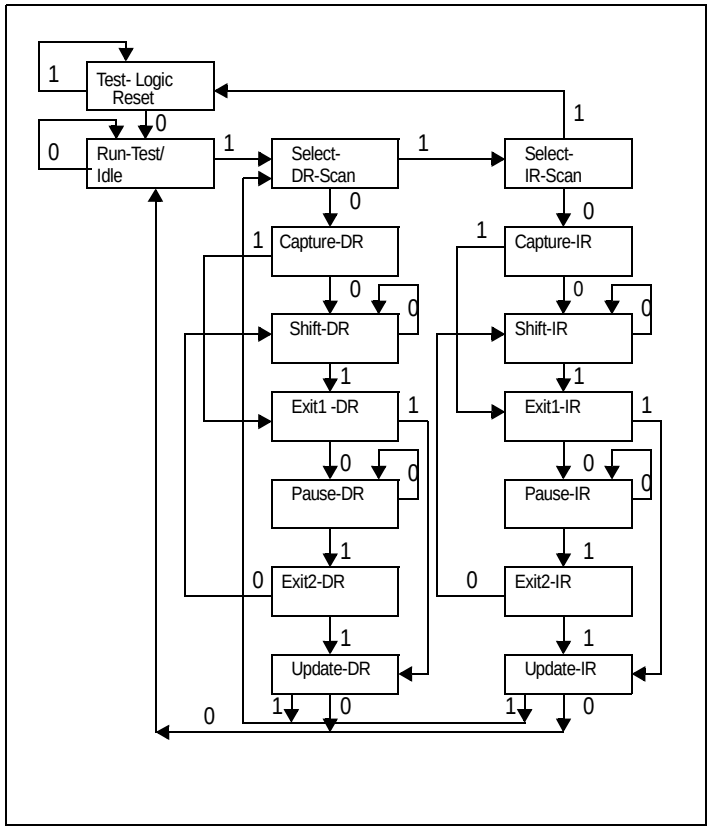

Figure 25.2 State Diagram of the TAP Controller

Notes

Boundary Scan Chain

Function	Pin Name	Type[1]	Boundary Cell[2]
PCI Express Interface	PE00RN[3:0]	I	O
	PE00RP[3:0]	I	
	PE00TN[3:0]	O	C
	PE00TP[3:0]	O	
	PE02RN[3:0]	I	O
	PE02RP[3:0]	I	
	PE02TN[3:0]	O	C
	PE02TP[3:0]	O	
	PE04RN[3:0]	I	O
	PE04RP[3:0]	I	
	PE04TN[3:0]	O	C
	PE04TP[3:0]	O	
	PE06RN[3:0]	I	O
	PE06RP[3:0]	I	
	PE06TN[3:0]	O	C
	PE06TP[3:0]	O	
PCI Express Interface (cont.)	PE08RN[3:0]	I	O
	PE08RP[3:0]	I	
	PE08TN[3:0]	O	C
	PE08TP[3:0]	O	C
	PE12RN[3:0]	I	O
	PE12RP[3:0]	I	
	PE12TN[3:0]	O	C
	PE12TP[3:0]	O	
	PE16RN[3:0]	I	O
	PE16RP[3:0]	I	
	PE162TN[3:0]	O	C
	PE16TP[3:0]	O	
	PE20RN[3:0]	I	O
	PE20RP[3:0]	I	
	PE20TN[3:0]	O	C
	PE20TP[3:0]	O	
	P[20,16,12,8,6,4,2,0] CLKN[3]	I	—
	P[20,16,12,8,6,4,2,0] CLKP[3]	I	
	GCLKN[1:0]	I	—
	GCLKP[1:0]	I	

Table 25.2 Boundary Scan Chain (Part 1 of 2)

Notes

Function	Pin Name	Type[1]	Boundary Cell[2]
SMBus	MSMBCLK	I/O	O/C
	MSMBDAT	I/O	O/C
	SSMBADDR[2:1][4]	I	O
	SSMBCLK	I/O	O/C
	SSMBDAT	I/O	O/C
General Purpose I/O	GPIO[8:0]	I/O	O/C
System Pins	CLKMODE[1:0]	I	O
	GCLKFSEL	I	O
	STK0CFG0	I	
	STK1CFG0	I	
	STK2CFG0		
	STK3CFG0		
	PERSTN	I	O
	RSTHALT	I	O
	SWMODE[3:0]	I	—
EJTAG / JTAG	JTAG_TCK	I	—
	JTAG_TDI	I	—
	JTAG_TDO	O	—
	JTAG_TMS	I	—
	JTAG_TRST_N	I	—
SerDes Reference Resistors	REFRES00	I/O	—
	REFRES01	I/O	—
	REFRES02	I/O	—
	REFRES03	I/O	—
	REFRES04	I/O	—
	REFRES05	I/O	—
	REFRES06	I/O	—
	REFRES07	I/O	—
	REFRESPLL	I/O	—

Table 25.2 Boundary Scan Chain (Part 2 of 2)

1. I = Input, O = Output
2. O = Observe, C = Control
3. Note that the PES32NT8BG2 device supports port clocking only for ports 0, 2, and 4.
4. The PES32NT8BG2 device supports only SSMBADDR2.

Test Data Register (DR)

The Test Data register contains the following:

- Bypass register
- Boundary Scan registers
- Device ID register

These registers are connected in parallel between a common serial input and a common serial data output and are described in the following sections. For more detailed descriptions, refer to IEEE Standard Test Access Port (IEEE Std. 1149.1).

Notes

Boundary Scan Registers

This boundary scan chain is connected between JTAG_TDI and JTAG_TDO when EXTEST or SAMPLE/PRELOAD instructions are selected. Once EXTEST is selected and the TAP controller passes through the UPDATE-IR state, whatever value that is currently held in the boundary scan register's output latches is immediately transferred to the corresponding outputs or output enables.

Therefore, the SAMPLE/PRELOAD instruction must first be used to load suitable values into the boundary scan cells, so that inappropriate values are not driven out onto the system pins. All of the boundary scan cells feature a negative edge latch, which guarantees that clock skew cannot cause incorrect data to be latched into a cell. The input cells are sample-only cells. The simplified logic configuration is shown in Figure 25.3.

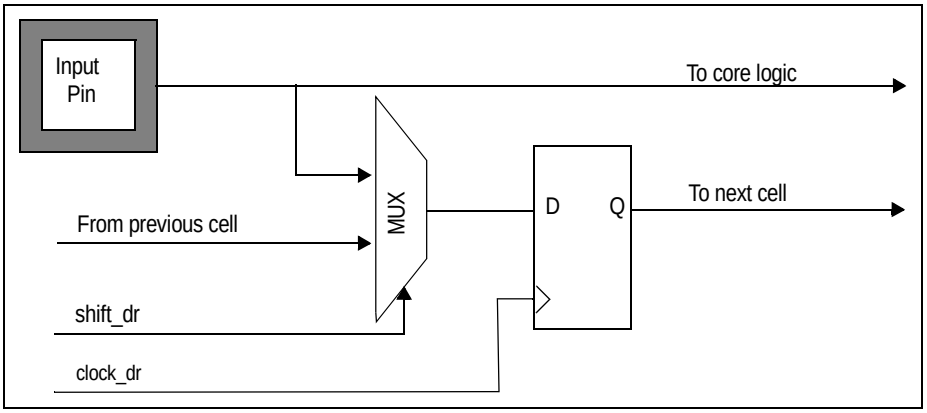

Figure 25.3 Diagram of Observe-only Input Cell

The simplified logic configuration of the output cells is shown in Figure 25.4.

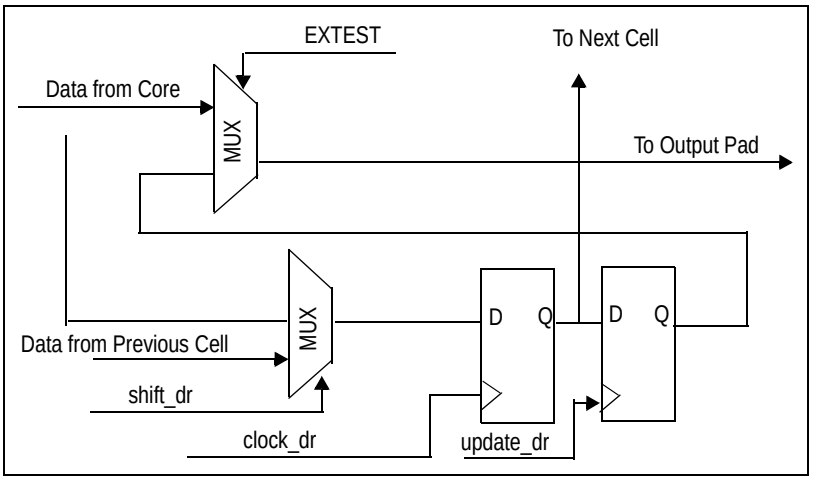

Figure 25.4 Diagram of Output Cell

The output enable cells are also output cells. The simplified logic is shown in Figure 25.5.

Notes

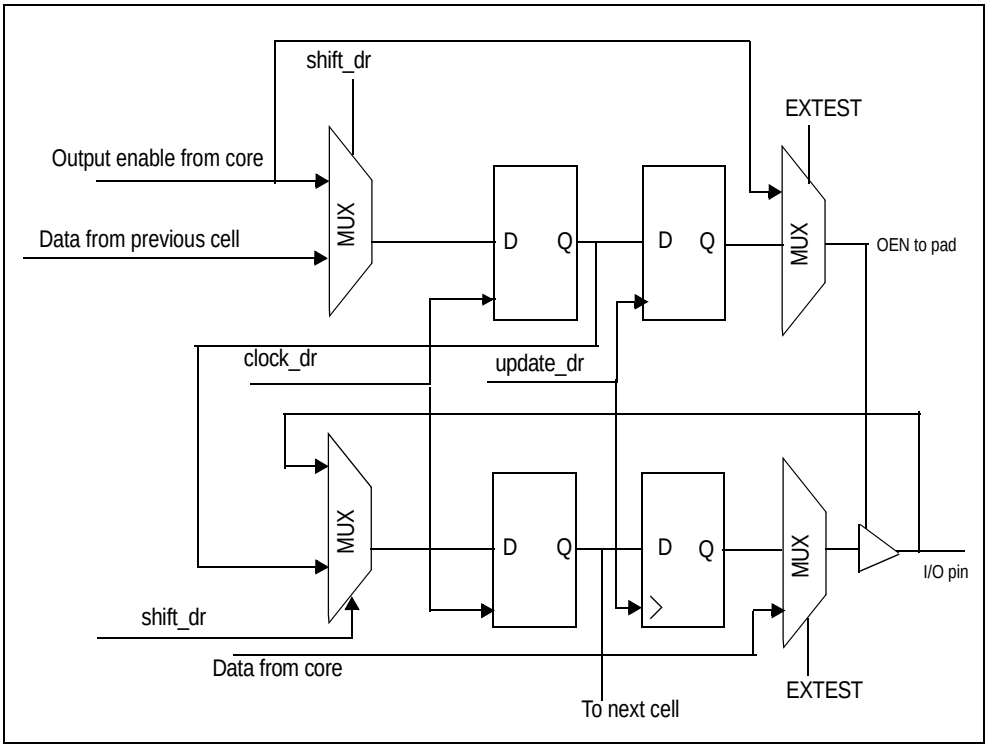

Figure 25.5 Diagram of Bidirectional Cell

The bidirectional cells are composed of only two boundary scan cells. They contain one output enable cell and one capture cell, which contains only one register. The input to this single register is selected via a mux that is selected by the output enable cell when EXTEST is disabled. When the Output Enable Cell is driving a high out to the pad (which enables the pad for output) and EXTEST is disabled, the Capture Cell will be configured to capture output data from the core to the pad.

However, in the case where the Output Enable Cell is low (signifying a tri-state condition at the pad) or EXTEST is enabled, the Capture Cell will capture input data from the pad to the core. The configuration is shown graphically in Figure 25.5.

Instruction Register (IR)

The Instruction register allows an instruction to be shifted serially into the device at the rising edge of JTAG_TCK. The instruction is then used to select the test to be performed or the test register to be accessed, or both. The instruction shifted into the register is latched at the completion of the shifting process, when the TAP controller is at the Update-IR state.

The Instruction register contains six shift-register-based cells that can hold instruction data. This register is decoded to perform the following functions:

- To select test data registers that may operate while the instruction is current. The other test data registers should not interfere with chip operation and selected data registers.
- To define the serial test data register path used to shift data between JTAG_TDI and JTAG_TDO during data register scanning.

The Instruction register is comprised of 6 bits to decode instructions, as shown in Table 25.3.

Notes

Instruction	Definition	Opcode
EXTEST	Mandatory instruction allowing the testing of board level interconnections. Data is typically loaded onto the latched parallel outputs of the boundary scan shift register using the SAMPLE/PRELOAD instruction prior to use of the EXTEST instruction. EXTEST will then hold these values on the outputs while being executed. Also see the CLAMP instruction for similar capability.	000000
SAMPLE/ PRELOAD	Mandatory instruction that allows data values to be loaded onto the latched parallel output of the boundary scan shift register prior to selection of the other boundary scan test instruction. The Sample instruction allows a snapshot of data flowing from the system pins to the on-chip logic or vice versa.	000001
IDCODE	Provided to select Device Identification to read out manufacturer's identity, part, and version number.	000010
HIGHZ	Tri-states all output and bidirectional boundary scan cells.	000011
VALIDATE	Automatically loaded into the instruction register whenever the TAP controller passes through the CAPTURE-IR state. The lower two bits '01' are mandated by the IEEE Std. 1149.1 specification.	101101
EXTEST_TRAIN	Used for AC pin test (IEEE 1149.6 specification)	111100
EXTEST_PULSE	Used for AC pin test (IEEE 1149.6 specification)	111101
CLAMP	Provides JTAG users with the option to bypass the part's JTAG controller while keeping the part outputs controlled similar to EXTEST.	111110
BYPASS	The BYPASS instruction is used to truncate the boundary scan register as a single bit in length.	111111
All other Opcodes are RESERVED		

Table 25.3 Instructions Supported by the JTAG Boundary Scan

EXTEST

The external test (EXTEST) instruction is used to control the boundary scan register, once it has been initialized using the SAMPLE/PRELOAD instruction. Using EXTEST, the user can then sample inputs from or load values onto the external pins of the device. Once this instruction is selected, the user then uses the SHIFT-DR TAP controller state to shift values into the boundary scan chain. When the TAP controller passes through the UPDATE-DR state, these values will be latched onto the output pins or into the output enables.

SAMPLE/PRELOAD

The sample/preload instruction has a dual use. The primary use of this instruction is for preloading the boundary scan register prior to enabling the EXTEST instruction. Failure to preload will result in unknown random data being driven onto the output pins when EXTEST is selected. The secondary function of SAMPLE/PRELOAD is for sampling the system state at a particular moment. Using the SAMPLE function, the user can halt the device at a certain state and shift out the status of all of the pins and output enables at that time.

BYPASS

The BYPASS instruction is used to truncate the boundary scan register to a single bit in length. During system level use of the JTAG, the boundary scan chains of all the devices on the board are connected in series. In order to facilitate rapid testing of a given device, all other devices are put into BYPASS mode.

Notes

Therefore, instead of having to shift many times to get a value through the device, the user only needs to shift one time to get the value from JTAG_TDI to JTAG_TDO. When the TAP controller passes through the CAPTURE-DR state, the value in the BYPASS register is updated to be 0.

CLAMP

This instruction, listed as optional in the IEEE 1149.1 JTAG Specifications, allows the boundary scan chain outputs to be clamped to fixed values. When the clamp instruction is issued, the bypass register is selected between TDI and TDO and the scan chain passes through this register to devices further downstream.

IDCODE

The IDCODE instruction is automatically loaded when the TAP controller state machine is reset either by the use of the JTAG_TRST_N signal or by the application of a '1' on JTAG_TMS for five or more cycles of JTAG_TCK as per the IEEE Std. 1149.1 specification. The least significant bit of this value must always be 1. Therefore, if a device has a Device ID register, it will shift out a 1 on the first shift if it is brought directly to the SHIFT-DR TAP controller state after the TAP controller is reset. The board- level tester can then examine this bit and determine if the device contains a Device ID register (the first bit is a 1), or if the device only contains a BYPASS register (the first bit is 0).

However, even if the device contains a Device ID register, it must also contain a BYPASS register. The only difference is that the BYPASS register will not be the default register selected during the TAP controller reset. When the IDCODE instruction is active and the TAP controller is in the Shift-DR state, the thirty-two bit value that will be shifted out of the Device ID register is shown in Figure 25.6.

Bit(s)	Mnemonic	Description	R/W	Reset
0	Reserved	Reserved	R	0x1
11:1	Manuf_ID	**Manufacturer Identity** (11 bits) This field identifies the manufacturer as IDT.	R	0x33
27:12	Part_number	**Part Number** (16 bits)		
		This field identifies the silicon as PES32NT8AG2.	R	0x808F
		This field identifies the silicon as PES32NT8BG2.	R	0x8088
31:28	Version	**Version** (4 bits) This field identifies the silicon revision of the PES32NT8xG2.	R	silicon-dependent

Table 25.4 System Controller Device Identification Register

Device	Version	Part Number	Mnfg. ID	LSB
PES32NT8AG2	xxxx	1000|0000|1000|1111	0000|0011|0011	1
PES32NT8BG2	xxxx	1000|0000|1000|1000	0000|0011|0011	1

Figure 25.6 Device ID Register Format

VALIDATE

The VALIDATE instruction is automatically loaded into the instruction register whenever the TAP controller passes through the CAPTURE-IR state. The lower two bits '01' are mandated by the IEEE Std. 1149.1 specification.

EXTEST_TRAIN

EXTEST_TRAIN instruction listed and explained in the IEEE 1149.6 JTAG specification. It is used to test AC pins during boundary scan by shifting data from TDI to TDO within the Shift-DR-TAP controller State. This instruction becomes effective on the falling edge of TCK in the Update-IR state.

Notes

After this instruction is asserted, the amount of time for which the pulses are generated is the amount of time for which the JTAG state machine is held in the Run-Test/Idle state.

If the Run-Test/Idle state is not entered, the output of the AC pins is not distinguishable from the output of the DC EXTEST instruction.

EXTEST_PULSE

EXTEST_PULSE is an instruction listed in IEEE 1149.6 JTAG specification and is used to test AC pins during boundary scan by shifting data from TDI to TDO within the Shift-DR-TAP controller State. This instruction becomes effective on the falling edge of TCK in the Update-IR state.

After this instruction is asserted, the width of the pulse is the amount of time for which the JTAG state machine is held in the Run-Test/Idle state.

If the Run-Test/Idle state is not entered, the output of the AC pins is not distinguishable from the output of the DC EXTEST instruction.

RESERVED

Reserved instructions implement various test modes used in the device manufacturing process. The user should not enable these instructions.

Usage Considerations

As previously stated, there are internal pull-ups on JTAG_TRST_N, JTAG_TMS, and JTAG_TDI. In order to guarantee that JTAG does not interfere with normal system operation, the TAP controller should be forced into the Test-Logic-Reset controller state by continuously holding JTAG_TRST_N low when the chip is in normal operation. If JTAG will not be used, apply an external pull-down resistor on JTAG_TRST_N to disable it.

Notes

Usage Models

Notes

Introduction

This chapter describes possible configurations of the PES32NT8xG2 switch and presents some important system usage models. The intent is to document important non-obvious device configuration procedures as well as usage models for the purpose of ensuring design correctness.

For each configuration, a series of steps to configure the device is outlined without necessarily delving into a detailed description of each step. Detailed descriptions on configuring the device are provided in other chapters of this document. Configurations other than those described in this chapter are possible.

The PES32NT8xG2 is one member of a family of PCI Express® devices from IDT. In this chapter, several related devices are used as examples; however, each example is also applicable to the PES32NT8xG2.

Boot-time Stack Reconfiguration

Goal

Reconfigure the stacks (at boot-time via serial EEPROM) to obtain the following configuration:

- One x8 port
- Sixteen x1 ports

Assumptions

- PES24NT24AG2 switch device.
- The switch boots in switch mode “Single partition with Serial EEPROM initialization”.
- Upstream port: Port 16 (x8)
- Downstream ports: Ports 0 to 15 (x1)

Figure 26.1 shows the configuration.

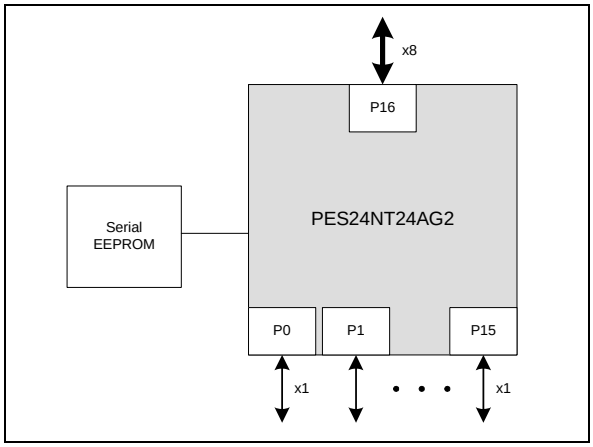

Figure 26.1 PES24NT24AG2 with One x8 port and Sixteen x1 Ports

Description

Stacks 0 and 1 will operate with four x1 ports each. In this product option, these stacks operate in this mode automatically and the stack configuration can’t be modified (i.e., the ports in the stacks are non-mergeable). Stack 2 will operate with eight x1 ports. To achieve this configuration, the STK2CFG[4:0] pins of the device can be tied to 0x1B on the system board.

Notes

Stack 3 will operate with one x8 port. To achieve this configuration, the STK3CFG[4:0] pins of the device can be tied to the appropriate value (refer to Table 3.7) on the system board. Alternatively, regardless of the value of the STK3CFG[4:0] pins, the stack may be dynamically reconfigured via serial EEPROM as follows:

1. For all ports associated with stack 3 (i.e., ports 16 to 23), the operating mode of each port is modified to 'Disabled' (i.e., via the SWPORTxCTL register).
   – Follow the guidelines in section Port Operating Mode Change via EEPROM on page 5-15.
   – The EEPROM Wait configuration block should be used to determine when the port operating mode change has completed, prior to executing the next step. Refer to section Initialization from Serial EEPROM on page 12-3.
2. Stack 3 is reconfigured to operate with one x8 port, by programming the STK3CFG register to the appropriate value (refer to Table 3.7). Note that only port 16 will be active in this configuration. Other ports in this stack (i.e., ports 17 to 23) are de-activated, regardless of the operating mode of those ports.
3. The operating mode of port 16 is modified to upstream switch port mode (i.e., via the SWPORT16CTL register).
4. The operating mode of ports 0 to 15 is modified to downstream switch port mode.
   – While the serial EEPROM loading executes, the switch is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2). To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset.
   – After EEPROM loading completes, the switch exits quasi-reset mode and the root complex can proceed to enumerate and configure the device. The root complex will find a switch with one x8 upstream port (port 16) and sixteen x1 downstream ports (ports 0 to 15). No other ports logically visible to enumeration software.

Port Clocking Configuration

Goal

Configure the switch (via serial EEPROM) such that the upstream port operates in local port clocked mode, and the downstream ports operate in global clocked mode.

Assumptions

- PES24NT6AG2 switch device.
- The switch boots in switch mode "Single partition with Serial EEPROM initialization".
- Upstream port: Port 0 (x8)
  - This port will operate in local port clocked mode, using the P0CLK clock input.
  - The upstream port is connected in a common clock configuration to its link partner (e.g., root port).
  - The P0CLK clock input is driven by a stable reference clock prior to the EEPROM loading.
- Downstream ports: Ports 4, 6, 8 and 12 (x4)
  - These ports will operate in global clocked mode, using the GCLK clock input.
  - Ports 4, 6, and 8 are connected in a common clock configuration to their respective link partners.
  - Port 12 is connected in a non-common configuration to its link partner.

Figure 26.2 shows the configuration.

Notes

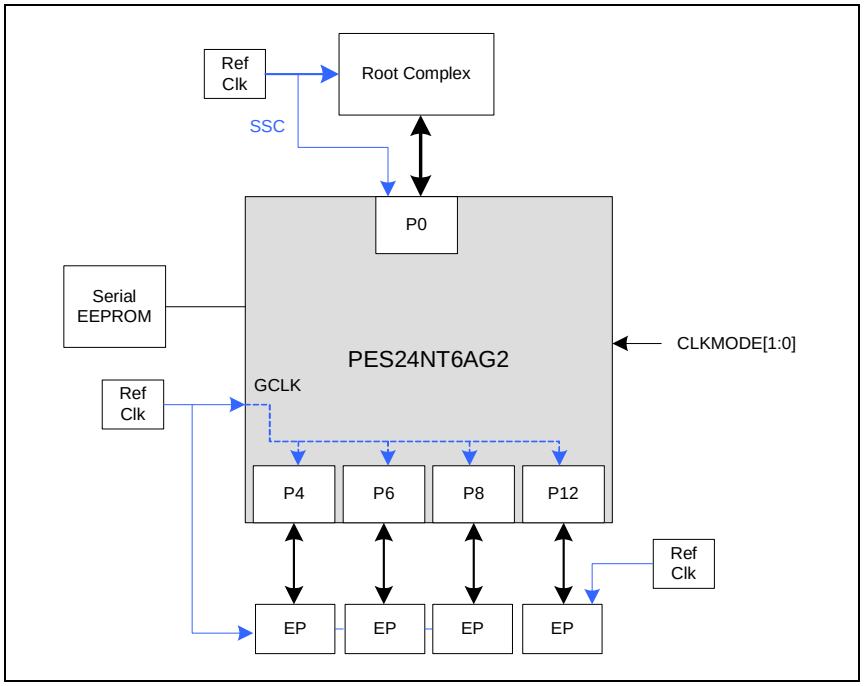

Figure 26.2 PES24NT6AG2 with Ports Operating in Different Clock Modes

Description

By default, all ports operate in global clocked mode. The CLKMODE[1:0] pins in the boot vector should be set to 0x1 to indicate that the upstream port of the switch operates in a common-clocked configuration with its link partner, while the downstream ports operate in a non-common clocked configuration with its link partner. Refer to Table 2.4.

Port 12 operates in a non-common clocked configuration with its link partner. Therefore, the serial EEPROM instruction sequence should clear the SCLK field in the PCI Express Link Status (PCIELSTS) of port 12. Follow the guidelines outlined in the implementation note entitled “Use of the Slot Clock Configuration and Common Clock Configuration Bits” in the PCI Express Base Specification 2.1.

To place the upstream port (port 0) in local port clocked mode, the serial EEPROM instruction sequence programs the P0CLKMODE field in the PCLKMODE register to select local port clocking mode.

- In response, the switch automatically changes the port’s clock mode using the sequence described in section Port Clocking Mode Selection on page 2-6.
- If port 0’s link was up at the time the EEPROM modifies the port’s clocking mode, the link transitions to the Detect state.
- After the hardware modifies the port’s clock mode, port 0’s link is automatically retrained.

With this configuration, port 0 and its link partner (e.g., a root-complex port) operate using the same reference clock, which is separate from the global clocked used by the switch downstream ports. Further, the reference clock used by the upstream port may have Spread Spectrum Clocking (SSC) as long as the switch’s global reference clock does not. Finally, port 12 operates in a non-common clocked configuration, while the other downstream ports operate in a common-clocked configuration.

Boot-time Switch Partitioning

This section describes examples of switch partitioning. The examples described here rely on the concepts and functionality described in Chapter 5, Switch Partition and Port Configuration.

Notes

Switch Partitioning via serial EEPROM

Goal

Configure switch partitions via the serial EEPROM (i.e., during switch fundamental reset).

Assumptions

- PES16NT8BG2 switch device.
- Two partitions will be created:
  - Partition 0 has ports 0, 8, and 10.
  - Partition 1 has ports 12, 16, and 18.
- Ports 0 and 12 are upstream, x4 each.
- Ports 8, 10, 16, and 18 are downstream, x2 each.
- The system board has a hardware fundamental reset signal connected to partitions 0 and 1, using switch's GPIO alternate function pins.
- The stacks are configured at boot-time using the STKxCFG pins to achieve the above link widths.
- The switch boots in switch mode "Multi-partition with Disabled ports and Serial EEPROM initialization".

Figure 26.3 shows the final configuration.

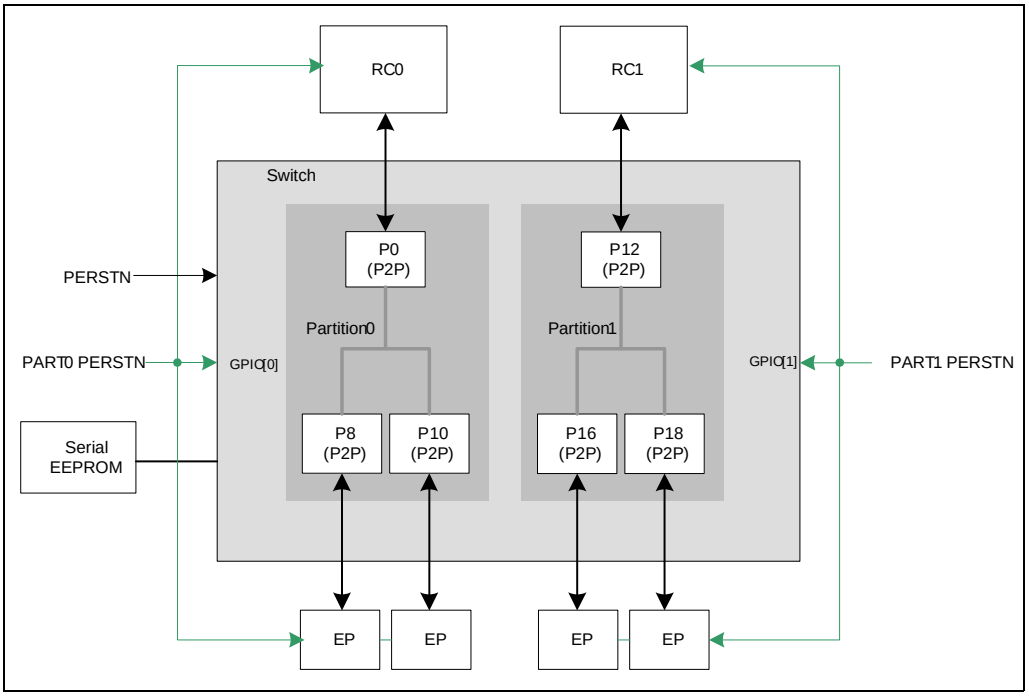

Figure 26.3 PES16NT8BG2 with Two Partitions Configured via Serial EEPROM

Description

As dictated by the switch mode, all ports and partitions are initially disabled. To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset. The guidelines regarding partition configuration via Serial EEPROM (section Port Operating Mode Change via EEPROM on page 5-15) should be followed. The serial EEPROM instruction sequence performs the steps listed below.

1. The GPIO pins 0 and 1 are configured for alternate function 0 operation (i.e., PARTxPERSTN reset input). These pins provide an external hardware control to reset the switch partitions.These pins may be asserted by the platform to do a fundamental reset on the corresponding switch partition and other devices in the same PCI Express hierarchy (e.g., root complex, endpoints).

   The PARTxPERSTN pins are free to be asserted and de-asserted by the platform while the follow-

Notes

ing steps take place.

2. Ports 0, 8, and 10 are migrated to partition 0.

   Port 0 is configured as an upstream switch port in partition 0 by programming fields in the SWPORT0CTL register appropriately. Ports 8 and 10 are configured as downstream switch ports in partition 0 by programming fields in the SWPORT8CTL and SWPORT10CTL registers appropriately.

   For this configuration, there is no need to use the EEPROM Wait configuration block, as each port is only configured once (i.e., the Wait configuration block is useful when applying multiple changes to a port's operating mode or to a partition's state).

3. Similarly, ports 12, 16, and 18 are migrated to partition 1.

4. The state of partitions 0 and 1 is modified from disabled to active by programming the SWPARTxCTL register appropriately.

   This causes all ports in the partitions to operate normally (i.e., the ports exit the disabled state and operate per the operating mode programmed in the SWPORTxCTL register).

   If the PARTxPERSTN reset signal is not asserted, all port links are trained. Otherwise, links remain down (i.e., in the Detect state) until the reset is de-asserted.

While the serial EEPROM loading executes, the switch is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2). After EEPROM loading completes, enumeration software associated with partition 0 will find a switch with one upstream port (port 0) and two downstream ports (ports 8 and 10). No other ports are logically visible in this partition.

Similarly, enumeration software associated with partition 1 will find a switch with one upstream port (port 12) and two downstream ports (ports 16 and 18). No other ports are logically visible in this partition.

Switch Partitioning via PCI Express Configuration Requests

Goal

Configure switch partitions via a port's PCI Express interface using configuration requests.

Assumptions

- PES16NT8BG2 switch device.
- A switch manager root complex configures the switch using a customized BIOS.
  - The switch manager is connected to port 0.
- Two partitions will be created:
  - Partition 0 has ports 0, 8, and 10.
  - Partition 1 has ports 12, 16, and 18.
- Ports 0 and 12 are upstream, x4 each.
- Ports 8, 10, 16, and 18 are downstream, x2 each.
- The stacks are configured at boot-time using the STKxCFG pins to achieve the above link widths.
- The switch boots in switch mode "Multi-partition with Unattached ports".
- Serial EEPROM is not used.

Figure 26.4 shows the final configuration.

Notes

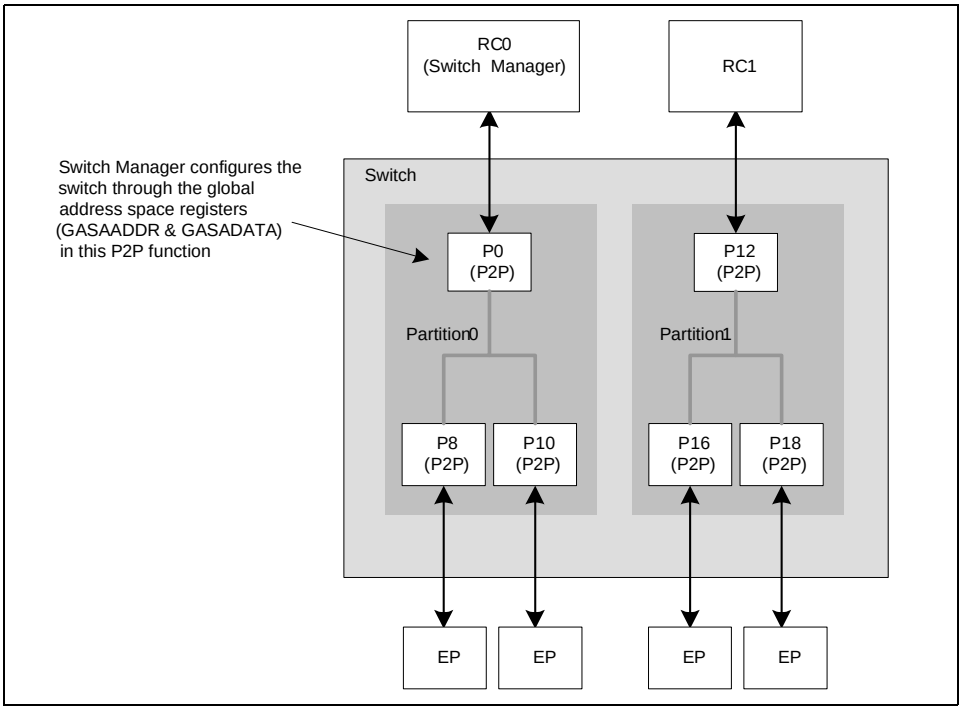

Figure 26.4 PES16NT8BG2 with Two Partitions Configured via a Switch Manager Root Complex

Description

As dictated by the switch mode, all ports and partitions are initially in unattached mode (see section Unattached on page 5-8).

- Ports connected to the root-complex or to another switch's downstream port on the system board link train normally.
- Ports connected to endpoint devices or to another switch's upstream port on the system board form a crosslink.[1]
- Ports respond to all received type 0 configuration request with configuration-request-retry-completion, except for accesses to the global space access registers in function 0 of the port (i.e., GASAADDR and GASADATA).
  - This in essence fences-off non switch manager roots from enumerating the switch until it is properly configured.
- The switch manager root (i.e., RC0, connected to port 0) has a customized BIOS to configure the switch. The other root complex (RC1 connected to port 12) is operating under a normal BIOS.
  - The customized BIOS starts by configuring switch partitions as described below, and then proceeds to enumeration.
- As indicated in section Switch Fundamental Reset on page 3-2, the switch manager has 1 second to configure the switch after the deassertion of the switch's fundamental reset signal (i.e., PERSTN) on the system board. This time limit represents the time during which RC1 won't give up on enumeration after receiving configuration-request-retry-completions.

Within this time, the switch manager performs the switch configuration as follows:

1. Waits for the data link of the root port to be DL_Up. Under normal circumstances this is completed within the first millisecond after de-assertion of PERSTN.
2. Issues type 0 configuration requests to the switch's port 0, targeting the GASAADDR and GASADATA registers.

   Through these registers, the switch manager is able to indirectly access the switch's global

[1] A crosslink is only formed if the link-partner supports crosslink. Otherwise, the crosslink formation fails until the unattached port's mode is modified to a downstream switch port mode.

Notes

address space, and therefore configure switch partitions by accessing the switch partition control (SWPARTxCTL) and switch port control (SWPORTxCTL) registers.

3. Disables access by other ports to the GASAADDR and GASADATA registers by programming the GASAPROT register.

   This action ensures that no other agent connected to the switch via PCI Express is allowed to configure the switch.

4. Modifies the state of partitions 0 and 1 from disabled to active by programming the SWPARTxCTL register appropriately.

   Since the switch manager is connected to port 0, and port 0 will be placed in partition 0, it is necessary that partition 0 be placed in the active state prior to migrating port 0 into this partition. Otherwise, if the partition was in disabled or reset state, port 0 would be automatically disabled/reset when migrated, thereby causing a link down and loosing connection to the switch manager.

5. The switch manager migrates ports 0, 8, and 10 to partition 0.

   Since partition 0 is active at the time the migration takes place, it is recommended that the downstream ports (ports 8 and 10) be migrated before the upstream port is migrated. This will ensure that by the time the port 0 is migrated and becomes an upstream switch port in partition 0, enumeration software in this partition will find a fully configured switch.

   Ports 8 and 10 are configured as downstream switch ports in partition 0 by programming fields in the SWPORT8CTL and SWPORT10CTL registers appropriately.

   Modifying the operating mode of these ports from unattached to downstream requires that the OMA field in the SWPORTxCTL register be set to 'Reset'. See section Port Operating Mode Change on page 5-13.

   Following the operating mode change, each port's link will train with its respective link partner.

   The switch manager should poll the SWPORT8STS and SWPORT10STS registers to check that the port operating mode change has completed before migrating port 0 to partition 0. This will ensure that the downstream ports are in the partition by the time the upstream port is migrated.

   Port 0 is configured as an upstream switch port in partition 0 by programming fields in the SWPORT0CTL register appropriately.

   The operating mode change action (OMA) field in the SWPORT0CTL register is set to 'No Action'. As a result, the operating mode change does not affect the port's link, and the connection between the switch manager and port 0 remains intact.

6. Following the same guidelines for partition 0, the switch manager migrates ports 12, 16, and 18 to partition 1.

   Since partition 1 is active at the time the migration takes place, it is recommended that the downstream ports (ports 16 and 18) be migrated before the upstream port is migrated. This will ensure that by the time the port 12 is migrated and becomes an upstream switch port in partition 1, enumeration software in this partition will find a fully configured switch.

After the switch manager completes configuration of the switch, enumeration software associated with partition 0 will find a switch with one upstream port (port 0) and two downstream ports (ports 8 and 10). No other ports are logically visible in this partition.

Similarly, enumeration software associated with partition 1 will find a switch with one upstream port (port 12) and two downstream ports (ports 16 and 18). No other ports are logically visible in this partition.

Notes

Dynamic Port and Partition Reconfiguration

I/O Load Balancing: Downstream Port Migration

Goal

The purpose of this section is to:

- Show the process of migrating a downstream port from one partition to another in an effort to perform I/O load balancing between two partitions.
- Use the global signals mechanism to coordinate the migration process.

Assumptions

The switch has been pre-configured by a switch manager device in the following configuration:

- Partition 0 has ports 0, 2, 3, and 4.
- Partition 1 has ports 6, 8, 10, and 12.
- Partition 2 has port 16.
- Ports 0 (x4) and 8 (x4) are the upstream ports in their respective partitions, each connected to a root-complex.
- Port 16 (x1) is configured in NT function mode, and is connected to a switch manager.
- The other ports (x2) are downstream ports in their respective partitions, and are connected to endpoint devices.
- Ports 2, 3, and 4 are assigned device numbers 0, 1, and 2 in partition 0.
- Ports 8, 10, and 12 are assigned device numbers 0, 1, and 2 in partition 1.

Figure 26.5 shows the initial configuration.

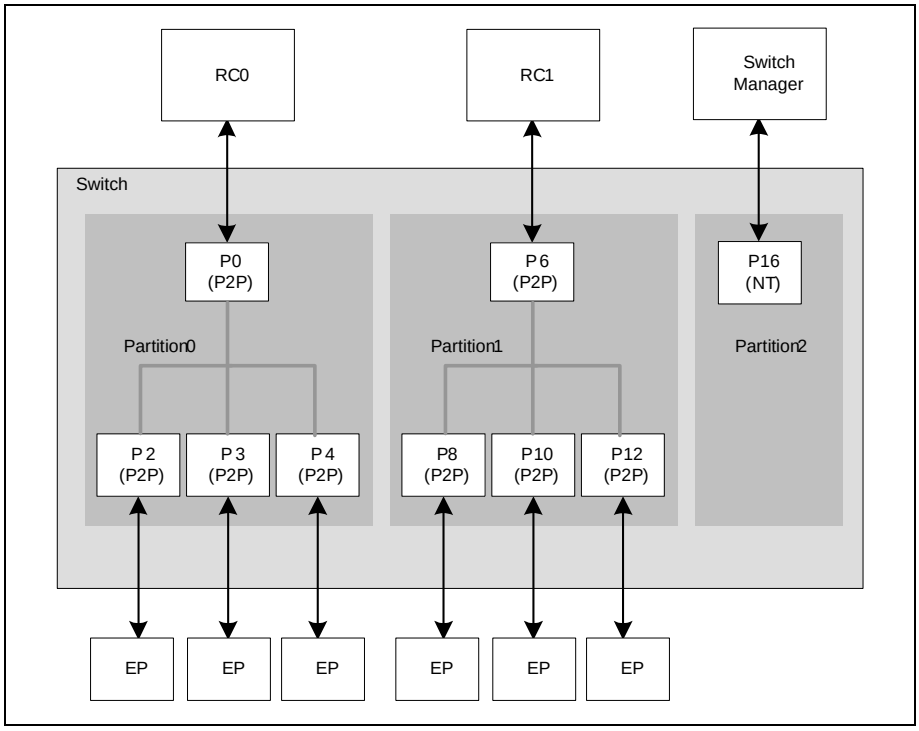

Figure 26.5 I/O Load Balancing Example: Initial Switch Configuration

The global signals mechanism is used for communication between the switch manager device and the root-complex in partitions 0 and 1. The mechanism is re-configured dynamically depending on the direction of the communication (i.e., from partitions 0 and 1 to the switch management agent or vice-versa).

Notes

When the switch manager device wishes to receive global signals from partitions 0 and 1, it configures the global signals mechanism as follows:

- The SEGSIGMSK register is configured to unmask global signals from partitions 0 and 1.
- The SEMSK register is configured to unmask global signals.
- The SEPMSK register is configured to unmask event signals to partition 2.

When the switch manager device wishes to send global signals to partition 0, it configures the global signals mechanism as follows:

- The SEGSIGMSK register is configured to unmask global signals from partition2.
- The SEMSK register is configured to unmask global signals.
- The SEPMSK register is configured to unmask event signals to partition 0.

When the switch manager device wishes to send global signals to partition 1, it configures the global signals mechanism as follows:

- The SEGSIGMSK register is configured to unmask global signals from partition2.
- The SEMSK register is configured to unmask global signals.
- The SEPMSK register is configured to unmask event signals to partition 1.
- The upstream ports in each partition are configured to unmask interrupts due to switch events.
- The NTINTMSK register in port 16 is configured to unmask interrupt generation due to switch events.
- The P2PINTMSK register in ports 0 and 1 is configured to unmask interrupt generation due to switch events.

Description

The switch mangement agent pre-configures the Global Signals mechanism to receive global signals from partitions 0 and 1 (as described above). The root-complex in partition 0 (RC0) requests I/O resources to the switch manager. It does so by using the global signaling mechanism as follows:

- RC0 writes a message to the P2PSDATA register in port 0. The message is a system-specific message requesting I/O resources. The encoding of such messages are outside the scope of this specification.
- RC0 sets the GSIGNAL field in the P2PGSIGNAL register in port 0. This triggers the global signal. This causes a switch event to be generated to partition 2. As a result, an interrupt is generated by the NT function in port 16 and sent to the switch manager device.

The switch manager device receives the interrupt and performs the following actions:

- Reads the NTINTSTS register in port 16 to determine the interrupt's cause.
- Upon noticing the interrupt was caused by a switch event, reads the SESTS register to determine the cause of the switch event.
- Upon noticing the switch event was caused by a global signal, reads the SEGSIGSTS register to determine the partition that issued the global signal.

The switch manager determines that it is partition 0 that issued the global signal, and reads the P2PSDATA register in port 0 to obtain the message. The message indicates that port 0 is requesting I/O resources. The message also indicates that port 0 has downstream port device numbers 0, 1, and 2 in its partition already. The port migration process should take this into account to prevent conflicting device numbers in the partition.

The switch manager re-arms the interrupt mechanism (by clearing the appropriate bits in the NTINTSTS and SEGSIGSTS registers).

The switch manager decides to honor port 0's request, and communicates with the root-complex in partition 1 (RC1) indicating that port 8 is about to be migrated out of partition 1. To allow this communication, the switch manager device dynamically re-configures the global signal mechanism such that it can send a global signal to partition 1 (as described above).

Notes

The communication between the switch manager device and RC1 is done using the global signals mechanism as follows:

- The switch manager writes to the NTSDATA register in port 16. The message is a system-specific message indicating that a root-complex should get ready to loose port 8 in the device. The encoding of such messages are outside the scope of this document.
- The switch manager sets the GSIGNAL field in the NTGSIGNAL register in port 16. This triggers the global signal which causes a switch event to be generated to partition 1. As a result, an interrupt is generated by the P2P function in port 6 and sent to RC1.
- RC1 receives the interrupt and performs the following actions:
  - Reads the P2PINTSTS register in port 16 to determine the interrupt's cause.
  - Upon noticing the interrupt was caused by a switch event, reads the SESTS register to determine the cause of the switch event.
  - Upon noticing the switch event was caused by a global signal, reads the SEGSIGSTS register to determine the partition that issued the global signal.
- RC1 determines that it is partition 2 that issued the global signal, and reads the NTSDATA register in port 16 to obtain the message. The message indicates that the switch manager device is requesting that port 8 is about to be migrated.
- RC1 re-arms the interrupt mechanism (by clearing the appropriate bits in the P2PINTSTS and SEGSIGSTS registers).

RC1 quiesces traffic on port 8, removes this port from its PCI Express hierarchy view, and issues a message to the switch manager by re-configuring the global signaling mechanism such that it can send global signals to the switch manager device (see description above).

- The quiescing of traffic, although recommended, is not required by the device prior to performing the next step.

Upon receiving the global signal from RC1, the switch manager device re-configures the global signal mechanism such that it can send global signals to RC0 (as described above). The switch manager device then issues a global signal and associated message to RC0 indicating that the I/O resource request has been granted. RC0 performs the necessary steps to prepare itself for the migration of port 8 into its partition and re-configures the global signal mechanism to communicate back to the switch manager device to indicate it is ready for the migration.

The switch manager performs the migration of port 8 by configuring the SWPORT8CTL register as follows:

- The OMA field is configured to 'Reset'.
- The SWPART field is configured to 0x0.
- The DEVNUM field is configured to 0x3. Device number 3 does not conflict with a downstream port device number in partition 0.

The switch manager polls the SWPORT8STS register to determine when the port operating mode change (e.g., port migration) is completed.

- Note that the migration of port 8 into partition 0 does not require that ongoing traffic in partition 0 be quiesced.

Once the port migration is completed, the switch manager re-configures the global signal mechanism and communicates with RC0 indicating that port 8 has been migrated to partition 0. RC0 proceeds to discover and configure the newly migrated port and its associated endpoint.

Figure 26.6 shows the partition configuration after the port migration has taken place.

Notes

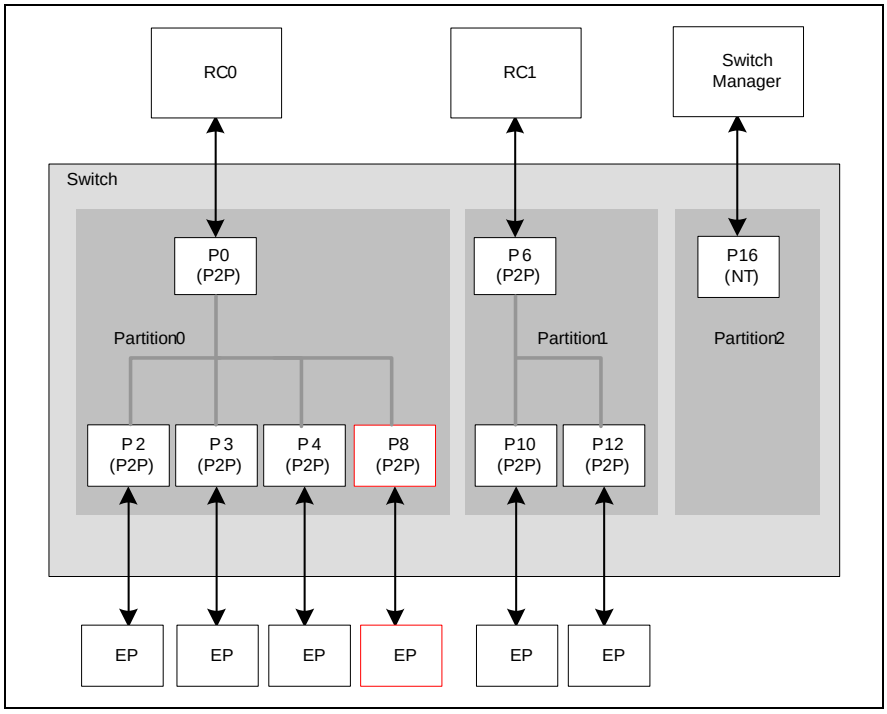

Figure 26.6 I/O Load Balancing Example: Switch Configuration after Port Migration

Non-Transparent Bridge (NTB) Usage Models

PES32NT8xG2 as a Multiprocessor System Interconnect

Goal

Describe the process of interconnecting loosely-coupled multiprocessors using the switch's non-transparent bridges.

Assumptions

- The switch is configured as shown in Figure 26.7.
- The configuration is achieved via firmware loaded from a serial EEPROM during a switch fundamental reset.

Notes

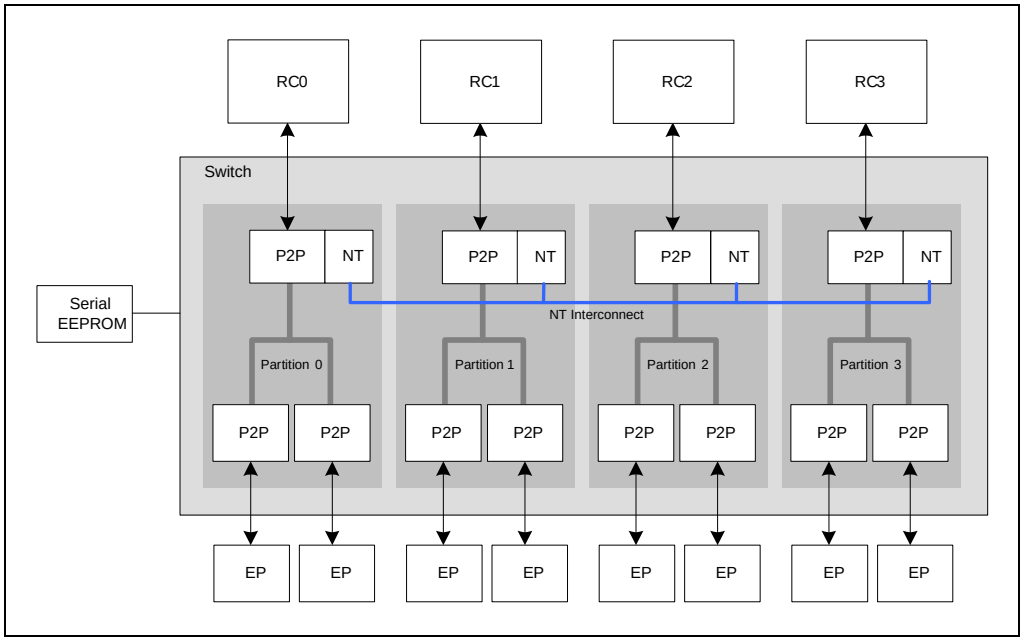

Figure 26.7 Multiprocessor System Interconnection Using the PES32NT8xG2

Description

The serial EEPROM preconfigures the switch partitions as shown in Figure 26.7. The serial EEPROM preconfigures the BARSETUP0 register in each NT function to map BARs 0/1 to the NT function's config space. The fields in the BARSETUP0 register are configured as follows:

- EN = 0x1 (i.e., enable the BAR)
- MODE = 0x1 (i.e., BAR is mapped to the NT function's configuration space)
- MEMSI = 0x0 (i.e., memory space BAR)
- TYPE = 0x2 (i.e., 64-bit addressing)
- PREF = 0x0 (i.e., non-prefetchable space)

Note: The value of other fields in this register is ignored by hardware once the MODE field is set to 0x1.

The fields in the BARSETUP1 register need not be configured since BAR 0 has been configured for 64-bit addressing in memory space. Later, when the root-complex in a partition enumerates the partition, it will allocate 4 KB of non-prefetchable memory space and assign it to BAR 0 in the NT function.

- The driver associated with the NT function accesses the function's configuration registers using memory read/writes to the memory space associated with NT BAR 0.

The serial EEPROM preconfigures BARSETUP2 register in each NT function to use NT lookup table translation. The fields in the BARSETUP2 register are configured as follows:

- EN = 0x1 (i.e., enable the BAR)
- ATRAN = 0x1 (i.e., 16-entry lookup table address translation)
- MODE = 0x0 (i.e., BAR is an address window)
- SIZE = 0x14 (i.e., BAR aperture size is 2^{20} = 1 MB)
- PREF = 0x1 (i.e., prefetchable space)
- TYPE = 0x2 (i.e., 64-bit addressing)
- MEMSI = 0x0 (i.e., memory space BAR)

Later, when the root-complex in a partition enumerates the partition, it will allocate 1 MB of prefetchable memory space and assign it to BAR 2 in the NT function.

Notes

The serial EEPROM preconfigures the NT messaging mechanism. In particular, it configures the manner in which messages issued by an NT function in a partition are transferred to NT functions in other partitions. During system operation, agents in each partition can communicate with each other using the switch's NT messaging mechanism. Such communication is useful when roots wish to exchange NT translation window information prior to configuring the NT lookup table.

Each NT function has 4 outbound message registers. Outbound message register X will be configured for messages destined to partition X. Each NT function will use a single inbound message register (INMSG0). Messages sent from NT functions in other partitions to this partition will be received in this register.

For example, the serial EEPROM configures the outbound messaging for partition 0 as follows:

- The PART field in the SWP0MSGCTL1 register is configured to 0x1.
- The PART field in the SWP0MSGCTL2 register is configured to 0x2.
- The PART field in the SWP0MSGCTL3 register is configured to 0x3.
- The REG field in the SWP0MSGCTL[3:1] registers is configured to 0x0 (i.e., all messages issued by partition 0 are mapped to the inbound message register 0 (INMSG0) in the partition targeted by the message).

The serial EEPROM preconfigures virtualization and protection in the NT Mapping table (i.e., the table that controls which devices in which partitions are allowed to communicate across the non-transparent bridges). The 64-entry NT Mapping table is virtualized, such that each partition is assigned 16 entries in the table (i.e., partition 0 is assigned entries 0 to 15, partition 1 is assigned entries 16 to 31, etc.).

For example, the NTMTBLPROT0 register is configured to virtualize the NT Mapping table for partition 0 as follows:

- TBLBASE = 0x0
- TBLLIMIT = 0xE

NT Mapping table protection is also enabled to prevent agent(s) in a partition from configuring the NT Mapping table inappropriately. Specifically, an agent in a partition is only allowed to program the NT Mapping table with the device IDs of PCI Express devices in that partition.

For example, the NTMTBLPROT0 register is configured as follows, such that the root complex in partition 0 (i.e., RC0 in the figure above) is only allowed to program the NT Mapping table with the IDs of other PCI Express devices in partition 0.

- REG = 0b1111_1110

While the serial EEPROM loading executes, the switch is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2). To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset.

After serial EEPROM completes preconfiguration of the switch, the root-complex in each partition enumerates the partition (e.g., assigns device IDs to each device in the PCI Express hierarchy and allocates memory space for PCI Express functions with BAR registers).

- The root allocates 4 KB of memory space associated with the NT function's BAR 0.
- The root allocates 1 MB of memory space associated with the NT function's BAR 2.

The root-complex in each partition enables the NT function to issue Message Signaled Interrupts (MSI) on inbound message reception as well as outbound message transmission failure.

- The appropriate bits are cleared in the MSGSTSMSK register.
- The appropriate bit is cleared in the NTINTMSK register.
- MSI is enabled in the NT function as described in section Interrupts on page 14-20.

Using the NT messaging mechanism (see section Message Registers on page 14-17), the roots exchange messages with each other regarding the allowed translated addresses in each domain. For example, the root-complex in partition 0 (RC0) sends a message to all other roots indicating the memory addresses that the other roots can access within RC0's domain. It is trusted that the each of the other roots will follow this and configure the NT lookup table entries associated with BAR 2 appropriately.

Notes

Each root-complex configures the NT lookup table in their corresponding NT function appropriately. For example, assume RC0 had received messages from RC1 indicating that it is allowed to transfer TLPs to a 256 KB window in RC1's memory address range starting at address 0x4000_0000 (1 GB).

Since BAR 2 is configured with a lookup table of 16 entries and a range of 1 MB, each table entry covers a range of 64 KB (see Table 14.2). Therefore, in order to cover a destination range of 256 KB, four entries in the NT lookup table are programmed for translation into partition 1, as follows:

Table Entry #	Valid	Partition	Translated Base Address[1]
0	1	1	0x4000_0000
1	1	1	0x4001_0000
2	1	1	0x4002_0000
3	1	1	0x4003_0000

Table 26.1 Example NT Lookup Table Programming

1. DWord aligned value of the translated base address field in the NT lookup table.

The root-complex in each partition proceeds to configure the NT Mapping table using PCI Express device IDs assigned during enumeration. The NT Mapping table is configured using the mechanism in section NT Mapping Table on page 14-9. For each device in the partition that is allowed to communicate across the NTB, an entry in the NT Mapping table is written. For example, assume RC0 wishes to allow the following PCI Express functions in RC0's domain to access the NT function:

- Bus = 0, Device = 0, Function = 0 (e.g., root-port)
- Bus = 3, Device = 0, Function = 0 (e.g., intelligent endpoint device)

RC0 programs the NT Mapping table as follows.

Table Entry #	RNS	CNS	ATP	PARTITION	BUS	DEV	FUNC	VALID
0	0	0	0	0	0	0	0	1
1	0	0	0	0	3	0	0	1

Table 26.2 Example NT Mapping Table Programming

Note: The table entries programmed in the NT Mapping table need not be contiguous.

This completes the programming of the NT translation mechanism. Other bits in the NT function need to be configured prior to starting traffic across the NTB (e.g., Bus Master Enable bit in the PCICMD register, Completion Enable bit in the NTCTL register). At this point, the multi-processor system is ready to use switch's NTB capabilities for inter-domain communication.

For example, in partition 0, the root-port with PCI Express ID = 0/0/0 can communicate with partition 1 by issuing memory read/write TLPs that fall within the first 256 KB of the BAR 2 address window. If the address falls within the first 64 KB of this range, the translated TLP's address will map to address (0x4000_0000 + address[15:0]) and emerge out of the NT function in partition 1. If the address falls within the next 64 KB of BAR 2, the translated TLP's address will map to address (0x4001_0000 + address[15:0]), and so on.

Notes

NT Crosslink & NT Punch-Through

Goal

The purpose of this section is to:

- Describe a system configuration with two switches, each connected to a root and two endpoints. The PES32NT8xG2 switches are interconnected to each other via NT ports, forming a crosslink.
- Describe how to achieve the initial configuration on both switches using a serial EEPROM connected to one of the switches, by using NT punch-through configuration requests (see section Punch-Through Configuration Requests on page 14-18).

Assumptions

Switch #1 boots in switch mode “Multi-partition with Unattached ports and Serial EEPROM initialization”. While the serial EEPROM loading executes, switch #1 is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2).

- Switch #2 boots in switch mode “Multi-partition with Unattached ports”.
- Prior to serial EEPROM initialization, the system is as shown in Figure 26.8.
- All ports are configured in unattached mode, and all partitions are disabled.

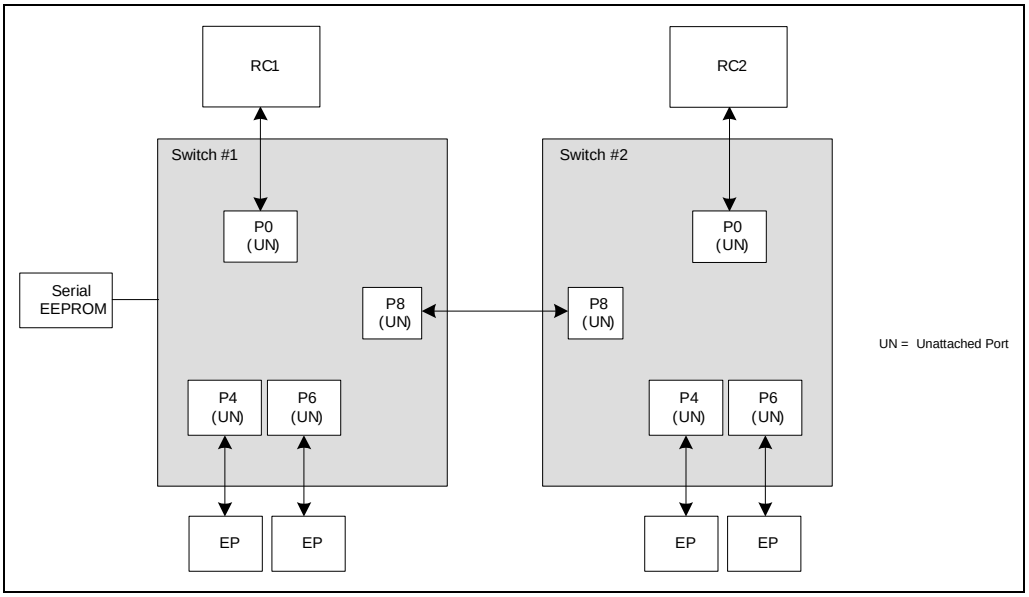

Figure 26.8 System Configuration immediately after Switch Fundamental Reset

Note: The link between the two switches automatically trains as a crosslink. The crosslink is formed by two ports operating in unattached mode (i.e., between port 8 in switch #1 and port 8 in switch #2). The ports are automatically configured to support crosslink.

The serial EEPROM connected to switch #1 is responsible for configuring both switches prior to enumeration by the roots. To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset. The target system configuration is shown in as shown in Figure 26.9.

Notes

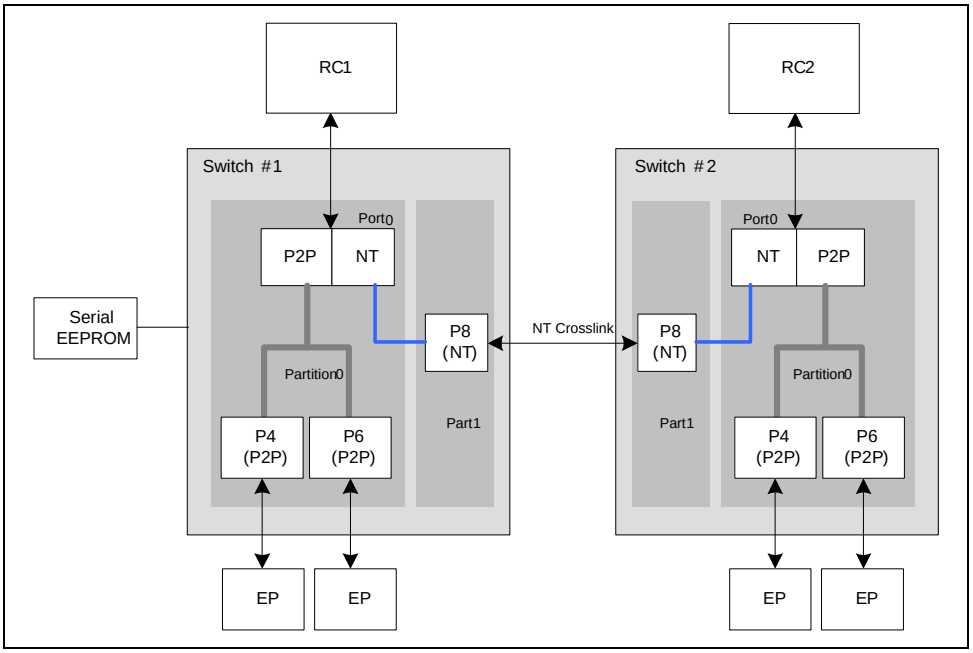

Figure 26.9 System Configuration after Serial EEPROM Initialization

Description

The serial EEPROM configures switch #1 as shown in Figure 26.9.

- Partitions 0 and 1 are placed in active state.
- Ports 4 and 6 are configured as downstream switch ports in partition 0.
- Port 8 is configured in NT function mode in partition 1.
- Port 0 is configured as an upstream switch port with NT endpoint in partition 0.
- This configuration is achieved by writing to Switch Partition Control (SWPARTxCTL) and Switch Port Control (SWPORTxCTL) registers in switch #1.

Note that the sequence shown above modifies the state of partitions 0 and 1 to active before adding ports to the partitions. This ensures that the links of these ports are not reset or disabled when the ports are migrated into the partition.

The serial EEPROM is also in charge of configuring switch #2. To do so, the serial EEPROM instruction sequence performs the following steps.

- The NT function in port 8 of switch #1 is directed to issue punch-through configuration requests. These configuration requests are sent from port 8 in switch #1 to port 8 in switch #2, via the cross-link formed between these devices.
- The punch-through configuration requests target function 0 of the port 8 in switch #2. Specifically, the punch-through configuration requests target the global space access registers (i.e., GASAADDR and GASADATA). Through these registers, any other register in the switch #2 may be accessed (e.g., the switch partition and control registers), thereby making it possible to configure partitions in the switch #2.

For example, to issue a punch-through type 0 configuration write request targeting the GASAADDR register in function 0 of port 8 in switch #2, the serial EEPROM instruction sequence performs the following actions:

- Write the following values to the Punch Through Configuration Control 0 (PTCTL0) register of the NT function in port 8 of switch #1. The PTCTL0 register is located at address 0x11510 in the switch's global address space[1]:

[1] This address is derived by summing the global address space base address of the NT function in port 8 (i.e., 0x11000) to the offset of the PTCTL0 register within the NT function's configuration space (i.e., 0x510).

Notes

- BUS = 0x0, DEV = 0x0, FUNC = 0x0, {EREG, REG} = 0xFF8

– Write the following values to the PTCTL1 register in the NT function in port 8 of switch #1:
  - CFGTYPE = 0x0 (i.e., type 0 configuration access)
  - OP = 0x1 (i.e., configuration write)
– Write the following value to the PTCDATA register in the NT function in port 8 of switch #1 switch:
  - DATA = 0x3E100 (i.e., the address of the SWPORT0CTL register, which will be accessed indirectly through the GASAADDR and GASADATA registers)
– The write to the PTCDATA causes switch #1 switch to send a type 0 configuration write request to port 8 in switch #2.
  - The requester ID is set to 0/0/4 (i.e., function 4 is used for punch-through requests).
  - The completer ID is set to 0/0/0 (i.e., as determined by the PTCTL0 register.
  - The configuration write request targets address 0xFF8 (i.e., as determined by the PTCTL0 register).
– Upon receiving the type 0 configuration write request, port 8 in switch #2 switch will write to the register located at address 0xFF8 (i.e., the GASAADDR register). The value written to this register is 0x3E100, which corresponds to the address of the SWPORT0CTL register in the global address space of switch #2.
– To complete the sequence, the serial EEPROM must wait until the Punch Through Status (PTCSTS) register reports that the punch through transfer has been completed correctly. To do so, the serial EEPROM uses a “Wait” configuration block (see section Initialization from Serial EEPROM on page 12-3) that stalls serial EEPROM execution until the DONE bit is set in the PTCSTS register. Once the DONE bit is set in the PTCSTS register, the serial EEPROM proceeds with the configuration sequence.

Using the same mechanism, the serial EEPROM can access the GASADATA register in the port 8 of switch #2. By using the GASAADDR and GASADATA registers to access any register in switch #2, the serial EEPROM can proceed to configure switch partitions in that switch.

The configuration of switch partitions in switch #2 could follow the sequence shown earlier for the partition configuration of switch #1.

Note that the root complex connected to switch #2 switch may start enumeration as soon as port 0 in that switch is configured to upstream switch port with NT function mode. Also, note that the sequence shown above configures the mode of port 0 (i.e., the port connected to the root complex) after all the other ports are configured. In this way, the root complex is guaranteed to find a fully configured switch when it enumerates.

Once the serial EEPROM configuration completes, switch #1 exits quasi-reset mode. Therefore, the root complex associated with switch #1 proceeds to enumerate the switch.

After enumeration, the roots can proceed to configure the NT functions (i.e., in port 0 and port 8 of their respective switches) and start communication across the NT crosslink. Since the NT function in port 8 of each switch is not directly visible to each root complex, configuring the NT function in port 8 must be done via the switch’s global address space (e.g., by accessing the GASAADDR and GASADATA registers in the NT function in port 0).

DMA Usage Models

High-Performance Multiprocessor System

Goal

Describe the usage of the switch’s DMA and NT functions to create a high-performance multiprocessor system.

Description

Fundamental reset is applied to the system. The switch in each processor node boots in switch-mode “Multi-partition with Unattached ports”. Immediately after fundamental reset is applied, the system is as shown in Figure 26.10.

Notes

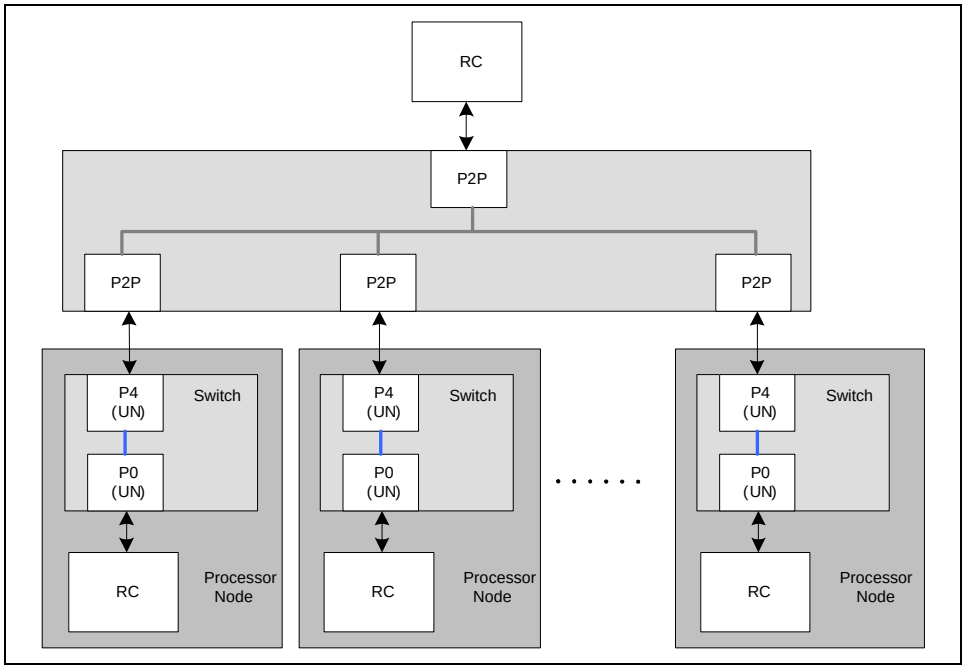

Figure 26.10 System Configuration Immediately after Switch Fundamental Reset

The roots in the system start the PCI Express hierarchy discovery process. The root-complex connected to the transparent switch starts enumeration. Enumeration will not complete as each processing node responds to received configuration requests with configuration-request-retry-completions (i.e., the switch ports are in unattached mode).

A customized BIOS in each processor node configures the switch using PCI Express configuration requests. The target configuration is shown in Figure 26.11.

- To meet PCI Express conventional reset requirements, this configuration completes within 1 second after the de-assertion of fundamental reset.
- The BIOS preconfigures the NT function BARSETUP registers, the NT function message registers, NT Mapping table protection, etc. (see example in section PES32NT8xG2 as a Multiprocessor System Interconnect on page 26-11).
- The BIOS preconfigures the DMA function BARSETUP registers. DMA BARs 0/1 are preconfigured to map DMA configuration space to 64-bit memory space.
- The BIOS configures the switch partitions and ports as shown in the figure, using the global address space registers in port 0.
- At this point, the NT ports are ready to receive and process configuration requests.

Notes

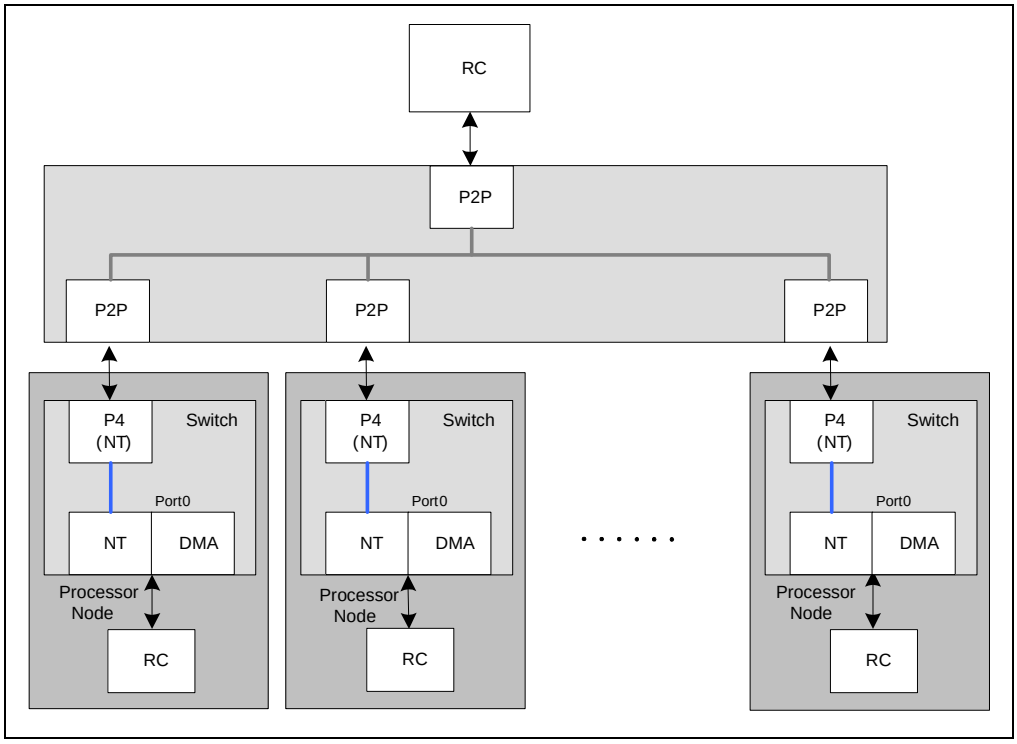

Figure 26.11 Target System Configuration

The root-complex connected to the transparent switch completes enumeration. This root-complex views each processor node as an endpoint device with a single function (i.e., the NT function). Enumeration software assigns a memory region to the BAR aperture(s) associated with the NT function.

In parallel with the previous step, the BIOS in each processor node proceeds to enumerate its PCI Express hierarchy. Enumeration software detects a hierarchy with a multi-function device containing two endpoint functions (i.e., NT function and DMA function) connected to a root-port. Enumeration software assigns memory regions to the BAR apertures associated with the NT function and DMA functions.

- An NT software driver will be associated with the NT function.
- A DMA software driver will be associated with the DMA function.

Software executing in the processing nodes and the transparent switch manager both use the NT messaging capability to exchange messages and coordinate the programming of the NTB windows used for communication.

- NTB windows for communication between the transparent switch's manager and any of the peer processors are configured.
- NTB windows for direct communication between the peer processors are configured.

The DMA function in the switch will be used for high-speed data transfer data among peer processors. The DMA offloads the CPU in each processing node from performing the data transfer. The DMA software driver operates by managing descriptor lists, controlling the DMA function channels, and processing DMA interrupts.

Software that wishes to use DMA services can issue a system call to the DMA driver to perform the desired data transfer. In particular, software layers in charge of multiprocessor communication need only issue a call to the DMA driver with system addresses that map to the appropriate NTB windows. The DMA driver builds the descriptor lists and programs the DMA function appropriately. The DMA hardware performs the transfer and issues an interrupt when the transfer completes.

Notes

Immediate Descriptor Usage

Goal

Describe the use of a DMA “immediate data transfer descriptor” in combination with NT doorbells as a mechanism to notify a target device of a completion of a DMA data transfer to the device (i.e., via interrupt generation).

Assumptions

- The DMA is configured to transfer data across the NTB, from a memory location in partition A to a memory location in partition B.
- The DMA is located in the upstream port of partition A.
- The user wishes that upon completion of the DMA data transfer, an interrupt be generated by the switch to notify both partitions.
- The DMA function in partition A is configured to generate an MSI upon completion of a DMA transfer operation.
- The NT function in the upstream port of partition A is configured to map its configuration space to BAR 0. This gives an agent in partition A, including the DMA, access to the memory-mapped NT doorbell registers.
- Outbound doorbell bit[0] in partition A has been preconfigured to set inbound doorbell bit[0] in partition B.

Description

The DMA descriptor list is set up by software. At the end of the list is an immediate data transfer descriptor (see section Immediate Data Transfer Descriptor on page 15-13). The immediate data transfer descriptor is configured such that:

- The address in the descriptor is that of the memory-mapped NT outbound doorbell register (e.g., OUTDBELLSET).
- The data in the descriptor sets outbound doorbell bit[0] in the OUTDBELLSET register.

The DMA operation is started by software (e.g., by setting the RUN bit in the DMACxSTS register). Upon completion of the DMA transfer, the following occurs:

- The DMA issues an MSI to the root complex in partition A.
- Inbound doorbell bit[0] is set in the NT function of partition B. This triggers the NT function to generate an interrupt to the root complex in partition B.

At this point, the root complex in both partitions have been notified of the DMA data transfer completion and can proceed appropriately. It is possible to use a similar mechanism with the NT messaging registers, instead of the NT doorbell registers. The NT messaging mechanism allows a message exchange in addition to the generation of the interrupt in partition B.

Failover

Active / Passive Failover Configuration

Goal

Describe an active/passive failover configuration, where the switch detects a failover condition (using a failover signal trigger) and automatically reconfigures switch partitions to swap the active/passive ports.

Notes

Description

Fundamental reset is applied to the system. The switch boots in switch-mode “Multi-partition with Disabled ports and Serial EEPROM Initialization”. The serial EEPROM configures the switch as shown in Figure 26.12.

- Port 0 is configured as an upstream switch port in partition 0.
- Port 8 is configured as an upstream switch port in partition 1.
- Partition 0 is placed in active state.
- Partition 1 is placed in disabled state. As a result, port 8 is implicitly disabled (see section Disabled on page 5-3).

To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset. While the serial EEPROM loading executes, the switch is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2).

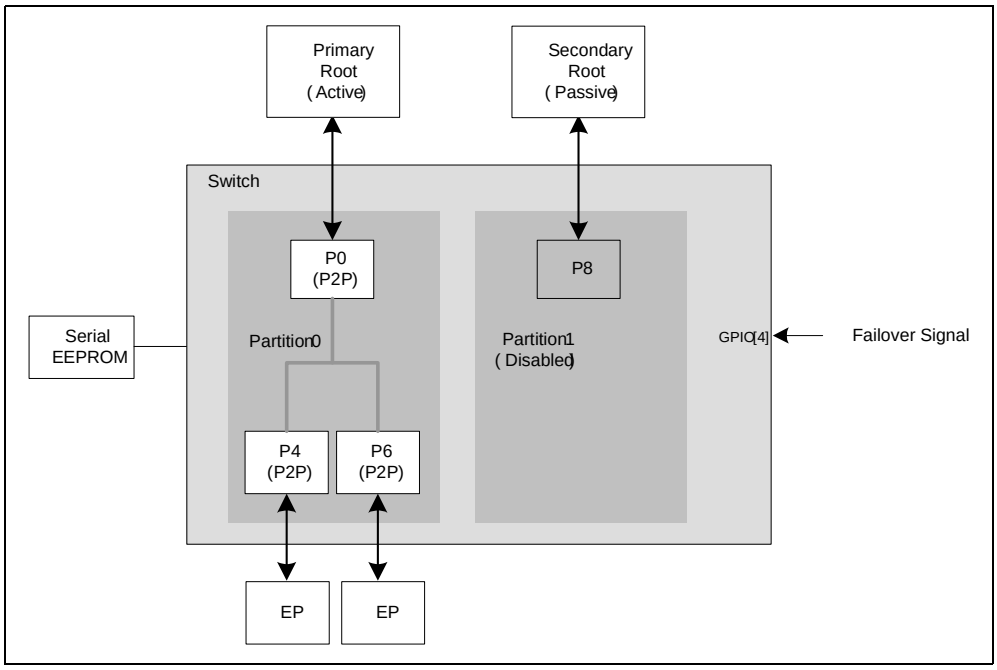

Figure 26.12 Active/Passive System Configuration Before Failover Event

The serial EEPROM configures the switch’s automatic failover mechanism as follows.

- Partition 0 is enabled to respond to failover capability 0. The FCAPSEL field in the SWPART0CTL register is set to 0x0. The FEN field in the SWPART0CTL register is set to 0x1.
- Similarly, partition 1 is enabled to respond to failover capability 0.
- Partition 0 is configured to respond to primary and secondary failover events by programming fields in the SWPART0FCTL register as follows:
  - SFSTATE = 0x0 (i.e., secondary failover state is set to disabled)
  - PFSTATE = 0x1 (i.e., primary failover state is set to active)
- Partition 1 is configured to respond to primary and secondary failover events by programming fields in the SWPART1FCTL register as follows:
  - SFSTATE = 0x1 (i.e., secondary failover state is set to active)
  - PFSTATE = 0x0 (i.e., primary failover state is set to disabled)
- Ports 4 and 6 are configured to respond to failover capability 0.
  - The FCAPSEL field in the SWPORT4CTL and SWPORT6CTL registers is set to 0x0.

Notes

- The FEN field in the SWPART4CTL and SWPORT6CTL registers is set to 0x1.

– Ports 4 and 6 are configured to respond to primary and secondary failover events by programming fields in the SWPORT4FCTL and SWPORT6FCTL registers as follows:
  - SFMODE = 0x1 (i.e., secondary failover mode is set to downstream port)
  - SFSWPART = 0x1 (i.e., secondary failover partition is set to partition 1)
  - SFDEVNUM = 0x4 (for port 0) and 0x6 (for port 1)
  - PFMODE = 0x2 (i.e., primary failover mode is set to downstream port)
  - PFSWPART = 0x0 (i.e., primary failover partition is set to partition 0)
  - PFDEVNUM = 0x4 (for port 4) and 0x6 (for port 6)
– In addition, ports 4 and 6 are configured to be reset during the failover operation by programming the Operating Mode Change Action (OMA) field in the SWPORT4CTL and SWPORT6CTL registers to a value of 0x1.
  - Resetting these ports is desired in this scenario so that when the failover operation takes place, the root complex to which these ports are assigned finds the ports in a non configured state.
  - In addition, resetting the ports causes the port's links to be retrained from the Detect state, thereby causing a hot reset to any devices downstream.
– The failover capability 0 is enabled to respond to a failover signal trigger by setting the FSIGPOL and FSIGEN bits in the FCAP0CTL register.
– Since the failover signal trigger associated with failover capability 0 is an alternate function of GPIO[4], this GPIO is configured for alternate function operation by programming GPIOAFSEL and GPIOFUNC registers appropriately. Refer to Chapter 13, General Purpose I/O for details.
– At this point the failover mechanism is armed. A failover is triggered when the platform asserts the failover signal trigger. The failover signal has the polarity programmed in the FSIGPOL field in the FCAP0CTL register.

After EEPROM loading completes, the switch exits quasi-reset mode and the roots can proceed to configure the device.

– The BIOS executing in the active root will find a fully configured switch with one upstream port and two downstream ports. The two downstream ports have device numbers 4 and 6.
– The BIOS executing in the passive root will find that the link that connects the root port to the switch's port 8 is down. As a result, no PCI Express hierarchy is found below that root port.

If the platform asserts the failover signal trigger (e.g., after detecting a problem with the primary root complex), a failover event is automatically executed by the switch. The failover event causes the following actions (in the order listed below[1]):

1. Ports 4 and 6 are migrated from partition 0 to partition 1. During the migration process, the ports are reset.

   This causes the downstream links to retrain from the Detect state, thereby causing a hot reset to the endpoint devices.

2. Partition 0 becomes disabled and partition 1 becomes active.

   When partition 0 becomes disabled, port 0 becomes disabled.

   When partition 1 becomes active, port 8 becomes active as an upstream switch port in that partition.

Figure 26.13 shows the system configuration after the failover event.

[1] Refer to section Partition Reconfiguration and Failover on page 5-21 for details on the order of reconfiguration actions during failover.

Notes

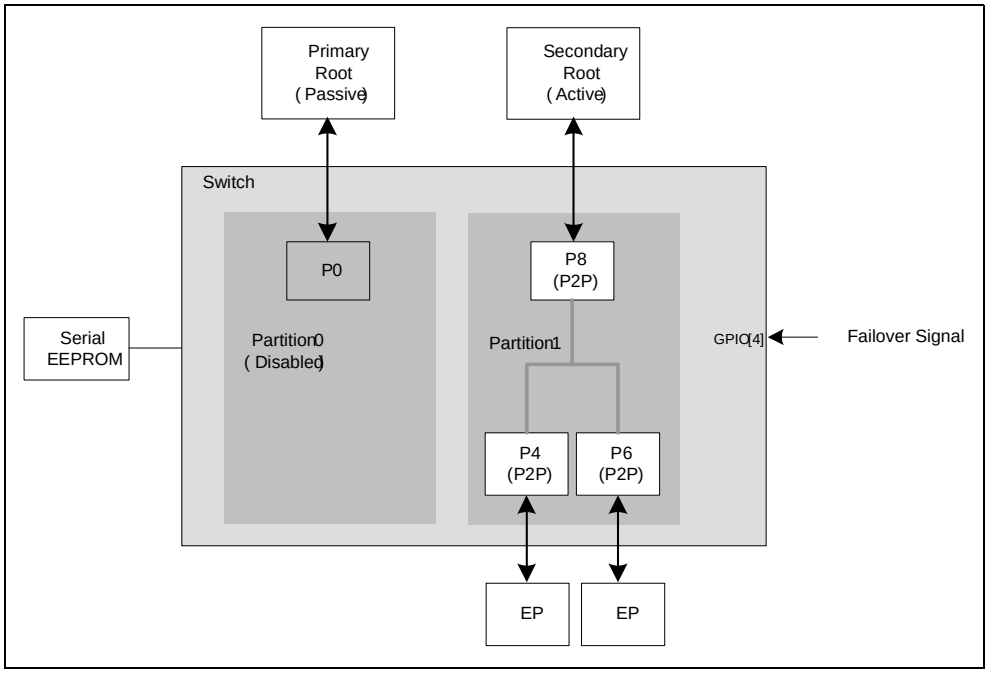

Figure 26.13 Active/Passive System Configuration after Failover Event

The root complex connected to port 8 can now configure the PCI Express hierarchy. It is assumed that the root complex is aware that the failover operation has occurred, so that it may proceed to configure the PCI Express hierarchy associated with the switch's port 8. In this example, the root complex can be made aware of the failover operation by a notification from the platform, in the same way that the switch is notified of the failover event through the assertion of the failover signal trigger.

Note that if the platform de-asserts the failover signal, the switch will execute another failover and automatically reconfigure itself as programmed (in this example, the switch will reconfigure itself back to the initial (i.e., primary) partition configuration shown in Figure 26.12).

Active / Active Failover Configuration

Goal

Describe an active/active failover configuration, where the switch is configured with two switch partitions, each connected to a root complex and some endpoints. The partitions are inter-connected via an NTB, which is used by the roots to exchange recovery point data. The PES32NT8xG2 is configured such that upon detection of a failover trigger (a software trigger in this case), the downstream ports from one partition are migrated to the other partition and the roots are notified of the event via an interrupt generated by the upstream switch port in the partition.

Description

Fundamental reset is applied to the system. The PES32NT8xG2 switch boots in switch-mode "Multi-partition with Unattached ports and Serial EEPROM Initialization". The serial EEPROM configures the switch as shown Figure 26.14.

To meet PCI Express conventional reset requirements, serial EEPROM configuration completes within 1 second after the de-assertion of fundamental reset. While the serial EEPROM loading executes, the switch is kept in quasi-reset mode (see section Switch Fundamental Reset on page 3-2).

Notes

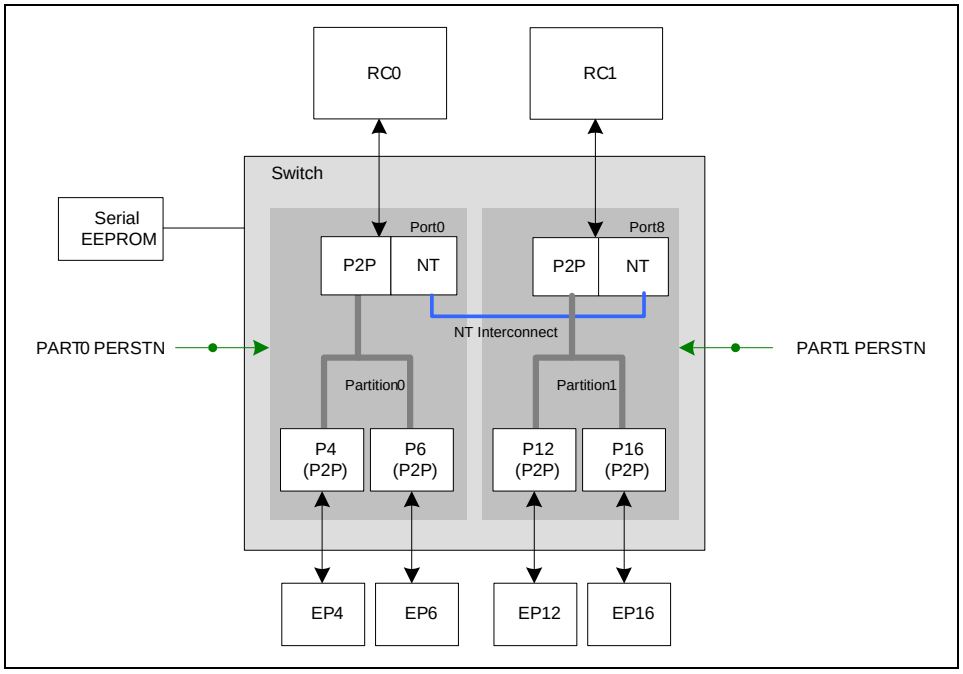

Figure 26.14 Active/Active System Configuration Before Failover Event

The serial EEPROM configures the GPIO pins 0 and 1 to operate in alternate function 0 mode.

- GPIO 0 acts as a partition fundamental reset input for partition 0 (i.e., PART0PERSTN).
- GPIO 1 acts as a partition fundamental reset input for partition 1 (i.e., PART1PERSTN).

The serial EEPROM configures the switch's event signaling mechanism such that partition fundamental reset events in partitions 0 or 1 are notified to the other partition. This allows a root complex to track the event of a partition fundamental reset occurring in the partition associated with the other root complex. The serial EEPROM configures the switch's automatic failover mechanism as follows:

- Failover capability 0 and 1 are configured to respond to a software initiated failover by setting the FSWTRIG bit in the FCAP0CTL and FCAP1CTL registers.
- Ports 0, 4 and 6 are configured to respond to failover capability 1.
- Port 0 is configured to respond to primary and secondary failover events by programming fields in the SWPORT0FCTL register as follows:
  - PFMODE = SFMODE = 0x5 (i.e., primary and secondary failover mode is set to unattached port)
- Ports 4 and 6 are configured to respond to primary and secondary failover events by programming fields in the SWPORT4FCTL and SWPORT6FCTL registers as follows:
  - PFMODE = SFMODE = 0x1 (i.e., primary and secondary failover mode is set to downstream port)
  - PFSWPART = SFSWPART = 0x1 (i.e., primary and secondary failover partition is set to partition 1)
  - PFDEVNUM = SFDEVNUM = 0x4 (for port 4) and 0x6 (for port 6)
- In addition, ports 0, 4 and 6 are configured to be reset during the failover operation by programming the OMA field in the corresponding SWPORTxCTL registers to a value of 0x1.

  Resetting these ports is desired in this scenario so that when the failover operation takes place, the root complex to which these ports are assigned finds the ports in a non-configured state.

  In addition, resetting the downstream ports causes the port's links to be retrained from the Detect state, thereby causing a hot reset to any devices downstream.
- Ports 8, 12 and 16 are configured to respond to failover capability 0.
- Port 8 is configured to respond to primary and secondary failover events by programming fields in the SWPORT8FCTL register as follows:

Notes

- PFMODE = SFMODE = 0x5 (i.e., primary and secondary failover mode is set to unattached port)

– Ports 12 and 16 are configured to respond to primary and secondary failover events by programming fields in the SWPORT12FCTL and SWPORT16FCTL registers as follows:
  - PFMODE = SFMODE = 0x1 (i.e., primary and secondary failover mode is set to downstream port)
  - PFSWPART = SFSWPART = 0x0 (i.e., primary and secondary failover partition is set to partition 0)
  - PFDEVNUM = SFDEVNUM = 0xC (for port 12) and 0x10 (for port 16)
– In addition, ports 12 and 16 are configured to be reset during the failover operation by programming the OMA field in the corresponding SWPORTxCTL registers to a value of 0x1.

Note: In this example, primary and secondary failover operations are configured identically. This is necessary because after a failover occurs and the switch reconfigures itself, restoring the system to its initial configuration is performed by software executing on the roots, and not through a subsequent failover operation.

The serial EEPROM configures port 0 so that the PCI-to-PCI bridge function in this port generates an interrupt when a failover completed event is detected in the failover capability structure associated with partition 0. To do this, the FMCC bit is cleared in the P2PINTMSK register of port 0.

Similarly, the serial EEPROM configures port 8 so that the PCI-to-PCI bridge function in this port generates an interrupt when a failover completed event is detected in the failover capability structure associated with partition 1. To do this, the FMCC bit is cleared in the P2PINTMSK register of port 8.

Note: The enabling of MSI/INTx is done at a later time by the operating system running on each of the roots.

At this point the failover mechanism is armed.

Failover capability 0 is associated with partition 0. When a failover is triggered, port 8 is reset and its operating mode is changed to unattached mode. In addition, ports 12 and 16 are reset and migrated to partition 0. Failover capability 1 is associated with partition 1. When a failover is triggered, port 0 is reset and its operating mode is changed to unattached mode. In addition, ports 4 and 6 are reset and migrated to partition 1.

After EEPROM loading completes, the switch exits quasi-reset mode and the root in each partition can proceed to enumerate the switch.

– RC0 enumerates partition 0.
– RC1 enumerates partition 1.

RC0 and RC1 enable the interrupt mechanism (i.e., INTx or MSI) in the PCI-to-PCI bridge function in the upstream switch port of their respective partitions, and bind the interrupt to a software interrupt handler. The interrupt handler is responsible for understanding and notifying other software layers of the re-configuration of the switch as a result of the failover event.

RC0 and RC1 proceed to configure the NT endpoint, use NT messaging to exchange NT window information, and configure the NT lookup and NT Mapping tables appropriately. During normal operation, RC0 and RC1 use the NTB to exchange recovery point information and issue “heart-beats” to each other.

When a root fails to issue a heart-beat, it is assumed to have failed. As a result, the other root triggers a software initiated failover by setting the FSWTRIG bit in the failover capability structure associated with its partition. Once the switch completes the failover operation, an interrupt is generated by the upstream port in the partition indicating that the failover change has completed.

For example, if RC0 fails to issue a heart-beat to RC1, RC1 triggers a failover by setting the FSWTRIG bit in the FCAP1CTL register. As a result, the switch automatically resets port 0 and changes its operating mode to unattached mode, and resets ports 4 and 6 and migrates them to partition 1 (see Figure 26.15).

When the switch completes the failover action, the upstream port in partition 1 generates an interrupt to RC1. The interrupt handler executes in RC1, and notifies other software layers that ports 4 and 6 are now available in the partition.

Notes

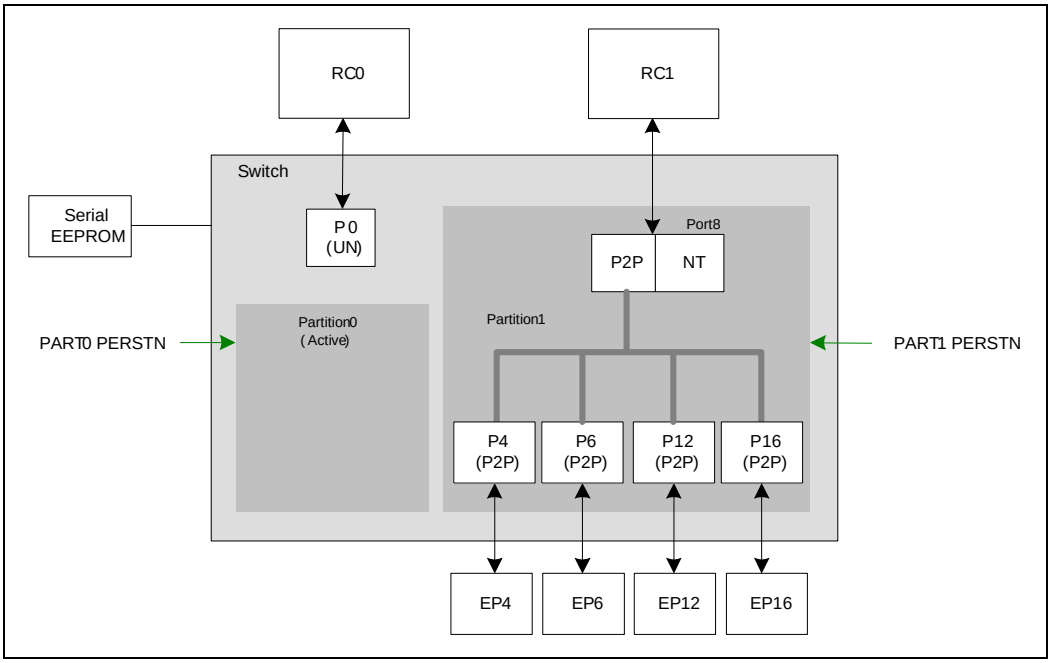

Figure 26.15 Active/Active System Configuration Before Failover Event

If at a later timer the failed root is repaired or replaced, a partition fundamental reset in the partition connected to the failed root complex is expected. The platform is responsible for asserting this fundamental reset to the switch (via the partition fundamental reset pin associated with the appropriate GPIO alternate function pin).

The switch's event signaling mechanism notifies the partition fundamental reset event to partitions 0 and 1. The notification results in an interrupt being generated by the upstream port in the partition associated with the root which did not fail. Continuing with the previous example, if RC0 had failed and is later replaced, the platform asserts the fundamental reset associated with partition 0 (i.e., via GPIO[0]). As a result, the switch's event signaling mechanism detects a partition fundamental reset event in partition 0. The switch's event signaling mechanism notifies this event to partitions 0 and 1. This in turn causes the PCI-to-PCI bridge function in port 8 to generate and send an interrupt to RC1. Note that since partition 0 is empty (i.e., port 0 is unattached and ports 4 and 6 are in partition 1 as a result of the failover), no further action occurs in this partition.

Upon receiving the interrupt, the root checks the source of this interrupt by inspecting the PCI-to-PCI bridge Interrupt Status (P2PINTSTS), Switch Event status (SESTS), and Switch Event Link Up Status (SELINKUPSTS) registers. In this way, the root determines that the interrupt is due to a partition fundamental reset in a partition associated with another root complex. The root complex clears the event status bits during the inspection to re-arm the event signaling mechanism.

Continuing with the previous example, after RC1 receives the interrupt, it inspects the P2PINTSTS, SESTS, and SELINKUPSTS registers and determines that partition 0 has experiences a fundamental reset. RC1 clears the bits in the SELINKUPSTS and P2PINTSTS registers to re-arm the event signaling mechanism.

In order to restore the system to its initial configuration (shown in Figure 26.14 above), the root complex that received the interrupt proceeds to de-allocate any system resources associated with the ports that it will migrate out of the partition, and then reconfigures the switch by programming the SWPORTxCTL registers corresponding to the ports that will be migrated.

Continuing with the previous example, RC1 de-allocates system resources associated with ports 4 and 6, and then it resets and migrates these ports to partition 0. In addition, RC1 resets and modifies the operating mode of port 0 to upstream switch port with NT function mode, placing it in partition 0. To ensure

Notes

correct system operation, RC1 migrates the downstream ports 4 and 6 before modifying the operating mode of the upstream port 0. In this way, the root connected to the upstream port 0 will find a fully established partition during enumeration.

Note: To meet PCI Express conventional reset requirements, RC1 must reconfigure the switch within 1 second after the de-assertion of the partition fundamental reset to partition 0. In this way, RC0 will be able to enumerate partition 0 correctly.

Prior to this point, the root complex that failed and was restored could not configure its PCI Express hierarchy because it was connected to an unattached port. After the partition reconfiguration in the prior step, the upstream port connected to this root is configured in upstream switch port with NT function mode. Therefore, after the platform de-asserts the partition fundamental reset, the root complex can now enumerate its PCI Express hierarchy normally, configure the endpoints, configure the NT function, and re-establish communication with the other root. At this point the system has returned to its initial configuration. If a new failover occurs, the process is repeated.

To finalize the example, after the reconfiguration initiated by RC1, RC0 will find that port 0 (which was previously unattached) operates in upstream switch port with NT function mode. In addition, it will find that downstream ports 4 and 6 are connected to partition 0's virtual PCI bus. Therefore, RC0 can enumerate the hierarchy normally, configure the NT function, and re-establish communication with RC1.

Failover with Two Crosslinked PES32NT8xG2 Switches

Goal

The purpose of this section is to describe:

- A system configuration with two switches, each connected to a root complex and two endpoints. The switches are connected to each other via NT ports, forming a crosslink.
- A failover scenario in which one of the roots fails and the other root configures both switches such that all endpoints in the system are under the control of the active (i.e., non-failed) root.

Assumptions

The system is pre-configured as shown in Figure 26.16.

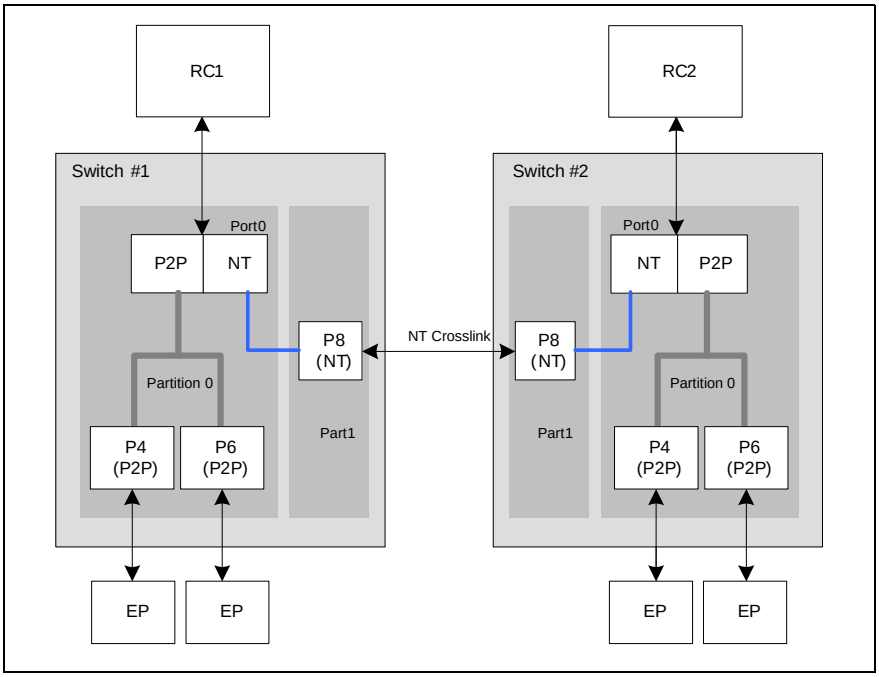

Figure 26.16 High Availability System Configuration with Redundant PCI Express Switches

Notes

Description

In each switch, the event signaling mechanism is configured to report link down events that occur on ports 0 and 8 to switch partitions 0 and 1. In switch #1, a link down event on port 0 is configured to be reported to partitions 0 and 1.

- In partition 0, port 0 has its link down. Therefore, partition 0 is hot reset and the reporting of the event to this partition has no effect.
- In partition 1, the NT function in port 8 generates an MSI towards the #2 switch.

In switch #2, a link down event on port 0 is configured to be reported to partitions 0 and 1.

- In partition 0, port 0 has its link down. Therefore, partition 0 is hot reset and the reporting of the event to this partition has no effect.
- In partition 1, the NT function in port 8 sends an MSI towards the #1 switch.

In either switch, a link down event on port 8 is reported to partitions 0 and 1.

- In partition 0, the NT function in port 0 sends an MSI towards the corresponding root complex.
- In partition 1, port 8 has its link down. Therefore, partition 1 is hot reset and the reporting of the event to this partition has no effect.

The configuration of the event signaling mechanism described above allows each root to detect a failure in the link that connects the switches together (i.e., the NT crosslink) as well as a failure in the link that connects the other root to its corresponding switch. Assume that the link that connects root complex 1 (RC1) to switch #1 fails (i.e., the link is down). As a result, the following occurs:

- The event signaling mechanism in switch #1 detects the link down event on port 0. As a result, port 8 in switch #1 sends an MSI to the #2 switch. Assume the MSI's address has been configured to map into BAR 2 of the NT function in port 8 of the #2 switch.
- The NT function in port 8 in switch #2 switch receives the MSI and translates it across the NTB. As a result, the TLP is emitted by the NT function in port 0 of the #2 switch (i.e., towards root complex 2 (RC2)).
- RC2 receives the MSI, processes the interrupt, and determines that the link that connects RC1 to the #1 switch has failed.

As a result of the failure, RC2 decides to reconfigure the switches to take ownership of the endpoints previously associated with RC1 (i.e., the endpoints connected to ports 4 and 6 in the #1 switch). To do so, RC2 performs the following steps:

- RC2 disables the event signaling mechanism in the #2 switch. This ensures that no link down event is reported when RC2 proceeds to modify the operating mode of port 8 (see below).
- RC2 accesses the switch control registers in the #2 switch (e.g., via the GASAADDR and GASADATA registers located in the NT function in port 0) and modifies the operating mode of port 8 to unattached. This operating mode change is required as direct transitions from the NT function mode to downstream switch port mode are not supported (refer to section Port Operating Mode Change on page 5-13).
- RC2 waits for the operating mode change to complete (i.e., by polling the SWPORT8STS register).
- RC2 modifies the operating mode and partition association of port 8 in switch #2 as follows:
  - The operating mode is set to downstream switch port.
  - The partition is set to 0x0 (i.e., partition 0)
  - In addition, the OMA field in the SWPORT8CTL register is set to reset, to ensure that the downstream switch PCI-to-PCI bridge function is reset.

  As a result of the port operating mode change action being set to reset, the link associated with port 8 retrains from the Detect state. This in turn causes the NT function in port 8 of the #1 switch to hot reset.

  The newly formed link is no longer a crosslink, since it is formed between a downstream switch port (i.e., port 8 in switch #2) and an NT function port (i.e., port 8 in switch #1).
- RC2 waits for the operating mode change to complete (i.e., by polling the SWPORT8STS register).

Notes

At this point, port 8 is part of the PCI Express hierarchy in partition 0 of the #2 switch (i.e., a downstream switch port located in the virtual PCI bus of partition 0). Figure 26.17 shows the system configuration. RC2 can now send configuration request TLPs to the PCI-to-PCI bridge function in port 8 without having to access the switch's global address space.

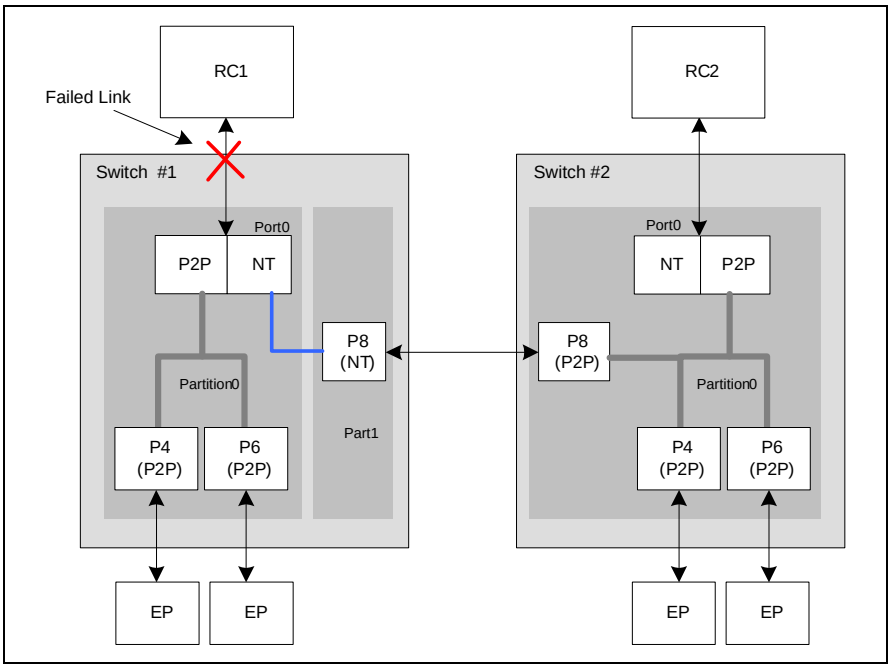

Figure 26.17 System Configuration after RC2 Modifies Port 8 in Switch #2 to Downstream Switch Port Mode in Partition 0

RC2 assigns a requester ID to the PCI-to-PCI bridge function in port 8, and programs the primary, secondary, and subordinate bus information appropriately. This primary, secondary, and subordinate bus information is programmed in such a way that it does not conflict with the existing bus IDs in the rest of the PCI Express hierarchy in partition 0.

- RC2 waits for the data link associated with port 8 to be active (i.e., by polling the DLLLA bit in the PCIELSTS register of port 8). Once the data link of port 8 is active, RC2 sends configuration write requests across port 8's link, to the NT function in port 8 of switch #1. The configuration write requests target the GASAADDR and GASADATA registers in this NT function (i.e., function 0), such that RC2 has access to the global address space in switch #1. This in effect allows RC2 to configure the #1 switch.
- RC2 modifies the operating mode of port 0 in switch #1 to unattached. This causes port 0 to not be associated with any partitions.
- RC2 waits for the operating mode change to complete (i.e., by polling the SWPORT0STS register in switch #1).
- RC2 modifies the operating mode of port 8 in switch #1 as follows:
  - The operating mode is set to upstream switch port.
  - The partition is set to 0x0 (i.e., partition 0)
  - The OMA field is set to 'No Action'.
- RC2 waits for the operating mode change to complete (i.e., by polling the SWPORT8STS register in switch #1).
  - The polling of the SWPORT8STS register is done using the GASAADDR and GASADATA registers in function 0 of port 8 in the #1 switch.
  - Note that function 0 of port 8 in switch #1 is in the process of transitioning from an NT function to a PCI-to-PCI bridge function as a result of the port operating mode change.

Notes

- Given that the port operating mode action is set to 'No Action', the port operating mode change process does not affect the link between the two switches or the ability of RC2 to access the GASAADDR and GASADATA registers in function 0 of port 8 in switch #1.

At this point, the PES32NT8xG2 switches are configured as shown in Figure 26.18. RC2 can now send configuration requests to the PCI-to-PCI bridge function in port 8 of switch #1.

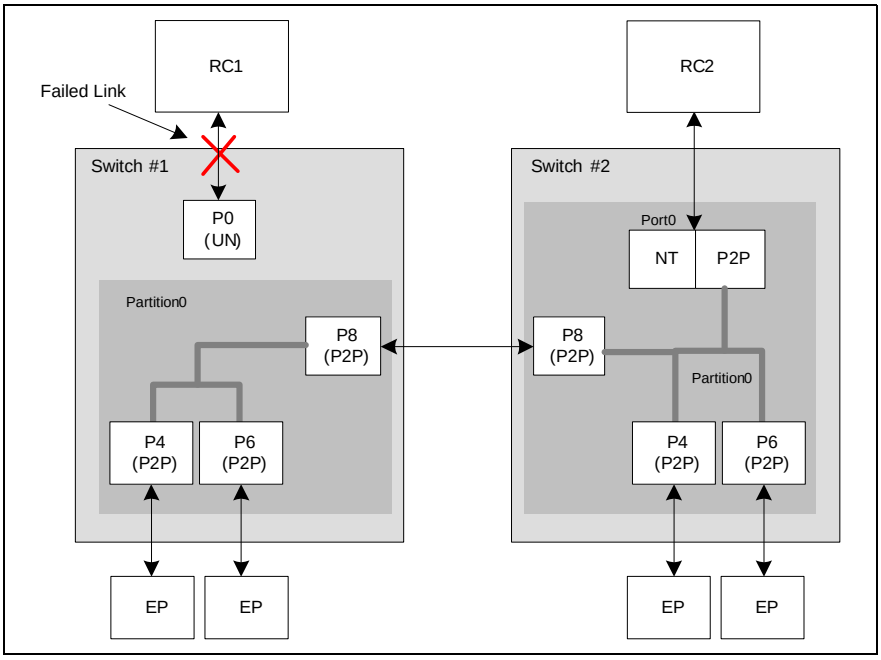

Figure 26.18 System Configuration after RC2 Modifies Port 8 in Switch #1 to Upstream Switch Port Mode in Partition 0

RC2 creates an upstream secondary hot reset in switch #1 by setting the SRESET bit in the Bridge Control (BCTL) register of the PCI-to-PCI bridge function in port 8 of switch #1. This causes a hot reset to the endpoints connected to switch #1. The hot reset continues until the SRESET bit is cleared. The SRESET bit in the BCTL register is cleared after an amount of time that guarantees that the conventional reset requirements in the PCI Express specification are met.

RC2 proceeds to enumerate the downstream ports in switch #1. These ports are now part of the PCI Express hierarchy associated with RC2. To monitor the status of the link between RC1 and switch #1, RC2 can enable the event signaling mechanism in switch #1 such that link up events are reported to partition 0 (i.e., the partition associated with ports 4, 6, and 8).

- Switch #1 can be configured such that a link up event on port 0 causes port 8 to generate an MSI towards RC2.
- If the data link between RC1 and switch #1 is later restored (i.e., the data link is up), port 8 generates an MSI to RC2.
- RC2 can then proceed to re-configure the PES32NT8xG2 switches to achieve the original configuration (see Figure 26.16), thereby restoring the system back to its pre-failure state.

NT Multicasting

Goal

This section describes a system configuration in which the switch's NT multicasting feature is used in conjunction with the DMA to multicast data from a switch partition to several other partitions. Refer to section Non-Transparent Multicast Operation on page 17-6 for a detailed description of NT Multicast operation.

Notes

Assumptions

The system is configured as shown in Figure 26.19. The data is to be NT multicasted from system memory in the root complex associated with switch partition 0 (i.e., RC0) to memory in the root complex associated with switch partitions 1 to 3. The DMA in the upstream switch port of partition 0 is used for this purpose.

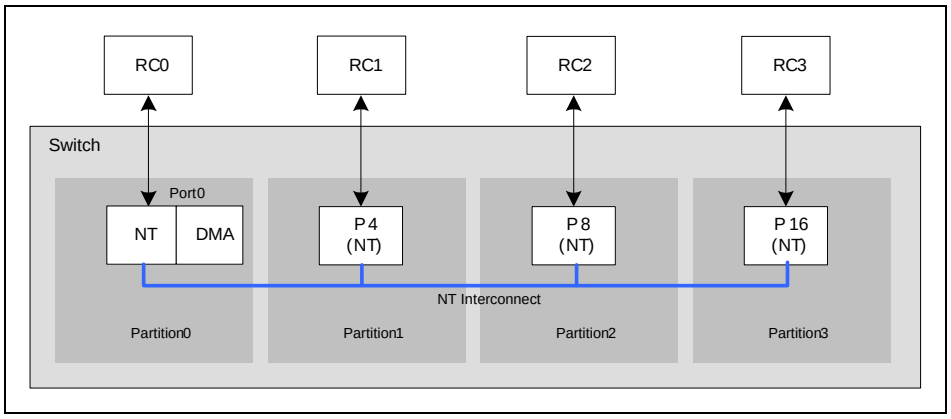

Figure 26.19 PES32NT8xG2 with Port 0 Configured in NT Function with DMA Mode and Ports 4, 8, and 16 in NT Function Mode

Description

In preparation to receive an NT multicast TLP (i.e., a TLP received on another partition that is multicasted to several destination partitions), the roots in partitions 1, 2, and 3 (i.e., RC1, RC2, and RC3) program NT multicast egress processing in the switch's NT port to which they are connected. In this way, when the NT port emits an NT multicast TLP, the TLP will have the desired address and requester ID.

For example, RC1 programs NT multicast egress processing in port 4. RC1 desires that when port 4 emits an NT multicast TLP, the requester ID in the TLP be that of the NT function in the switch's port to which RC1 is connected. Further, RC1 desires that the base address of the emitted NT multicast TLP be 0x00A0_0000. RC1 programs NT multicast egress processing in port 4 as follows:

- NTMCOVR0C.PART = 0x1 (i.e., NT multicast overlay register set 0 is associated with partition 0 only).
- NTMCOVR0C.GROUP = 0xF (i.e., NT multicast overlay register set 0 is associated with all 4 multicast groups in partition 0).

The above two settings imply that when an multicast TLP is received in partition 0, hits any of the first 4 multicast groups (i.e., groups 0 to 3), and the TLP is NT multicasted to port 4, NT multicast overlay register set 0 will control the manner in which port 4 performs address and requester ID overlay occur).

NTMCOVR0C.OVRREQID = <Requester ID of the NT function in port 4>

NTMCOVR0BARL.OVRSIZE = 0x14 (i.e., overlay windows size = 2^{20} = 1MB).

NTMCOVR0BARL.MCBARL = 0x0AA0_0000 (i.e., 32-bit overlay address)

NTMCC.NTMCAOE = 0x1 (i.e., enable NT multicast address overlay)

NTMCC.NTMCRIDOE = 0x1 (i.e., enable NT multicast requester ID overlay)

NTMCC.NTMCTEN = 0x1 (i.e., enable NT multicast transmission in this port)

Similarly, RC2 and RC3 program the NT multicast egress processing in ports 8 and 16 respectively. Each root programs NT multicast egress overlay processing independently and the programming may be done differently for each root.

Notes

For example, RC2 uses NT multicast overlay register set 0, programs the overlay address to base 0x0510_0000, and programs the overlay requester ID to match that of the NT function in port 8. Also, RC3 uses NT multicast overlay register set 0, programs the overlay address to base 0x0820_0000, and programs the overlay requester ID to match that of the NT function in port 16.

RC0, in preparation for the multicast operation, programs the PCI Express multicast capability structure in the NT endpoint of port 0. This structure is programmed to support a multicast aperture of 1 MB with 4 multicast groups (i.e., groups 0 to 3, the maximum number of groups supported for NT multicast). The multicast window starts at base address 0x0110_0000.

In addition, RC0 programs the NT Multicast Group x Port Association (NTMCG[3:0]PA) register to indicate that multicast groups 0 to 3 are all associated with ports 4, 8, and 16. In this way, a posted address-routed TLP received by the NT function in port 0 that falls within any of multicast groups 0 to 3 is NT multicasted to ports 4, 8, and 16.

NTMCG[3:0]PA.PORTVEC = 0x010088 (i.e., bits corresponding to ports 4, 8, and 16 are set).

At this point, the NT multicast mechanism is ready. To perform an NT multicast transfer, RC0 need only issue a posted address-routed TLP (e.g., memory write TLP) that falls within the multicast window associated with the NT function in port 0 (i.e., from address 0x0110_0000 to 0x011F_FFFF).

In this example, RC0 wishes to use the DMA in the switch's port 0 to transfer data from host memory to the memory associated with RC1, RC2, and RC3. To do this, RC0 sets up the DMA descriptors in its memory. The data transfer descriptors are programmed such that the source address points to the location in RC0's memory where the data to be transferred is located, while the destination address points to a location within the multicast window associated with the NT function in port 0.

When the DMA starts transferring the data, it will read the data from RC0's memory, convert the received completion TLPs to memory write TLPs, and transfer these to the destination address programmed in the descriptor. Since the destination address falls within the NT function's multicast window in partition 0, the switch will NT multicast the data to the programmed ports (i.e., ports 4, 8, and 16). Figure 26.20 shows the transfer.

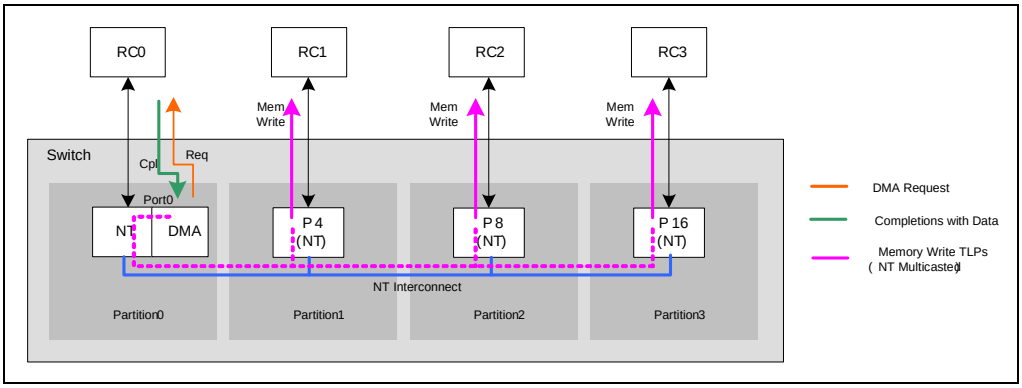

Figure 26.20 PES32NT8xG2 with Port 0 Configured in NT Function with DMA Mode and Ports 4, 8, and 16 in NT Function Mode

When the TLPs emerge on ports 4, 8, and 16, they will have the requester ID and address programmed in the NT multicast egress processing registers of each of these ports. For example, NT multicast TLPs emitted by port 4 will have:

- Requester ID = <Requester ID of the NT function in port 4> (i.e., as programmed in the OVRREQID field in port 4's NTMCOVR0C register).
- Address = { 0x0AA, DMA Destination Address[19:0] }

When the DMA finishes the transfer, it notifies RC0 via an interrupt. In order to notify RC1, RC2, and RC3, DMA immediate descriptors may be used in conjunction with the NT doorbell mechanism. Refer to section Immediate Descriptor Usage on page 26-20 for an example of this.

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View 89H32NT8AG2ZCHLG8 on WIN SOURCE
- IDT, Integrated Device Technology Inc Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management