



# IDT™ 89HPES22H16 PCI Express® Switch

## User Manual

October 2008

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.  
©2008 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

## Introduction

This user manual includes hardware and software information on the 89HPES22H16, a member of IDT's PRECISE™ family of PCI Express® switching solutions offering the next-generation I/O interconnect standard.

## 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](http://www.idt.com)) as well as through your local IDT sales representative.

## Content Summary

**Chapter 1, "PES22H16 Device Overview,"** provides a complete introduction to the performance capabilities of the 89HPES22H16. Included in this chapter is a summary of features for the device as well as a system block diagram and pin description.

**Chapter 2, "Upstream Port Failover,"** describes upstream port failover mechanism in the PES22H16 that enables the construction of fault tolerant systems.

**Chapter 3, "Clocking, Reset, and Initialization,"** provides a description of the two differential reference clock inputs that are used internally to generate all of the clocks required by the internal switch logic and the SerDes.

**Chapter 4, "Link Operation,"** describes the operation of the link feature including polarity inversion, link width negotiation, and lane reversal.

**Chapter 5, "General Purpose I/O,"** describes how the 32 General Purpose I/O (GPIO) pins may be individually configured as general purpose inputs, general purpose outputs, or alternate functions.

**Chapter 6, "SMBus Interfaces,"** describes the operation of the 2 SMBus interfaces on the PES22H16.

**Chapter 7, "Power Management,"** describes the power management capability structure located in the configuration space of each PCI-PCI bridge in the PES22H16.

**Chapter 8, "Hot-Plug and Hot-Swap,"** describes the behavior of the hot-plug and hot-swap features in the PES22H16.

**Chapter 9, "Configuration Registers,"** discusses the base addresses, PCI configuration space, and registers associated with the PES22H16.

**Chapter 10, "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.

## 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.

## Notes

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.

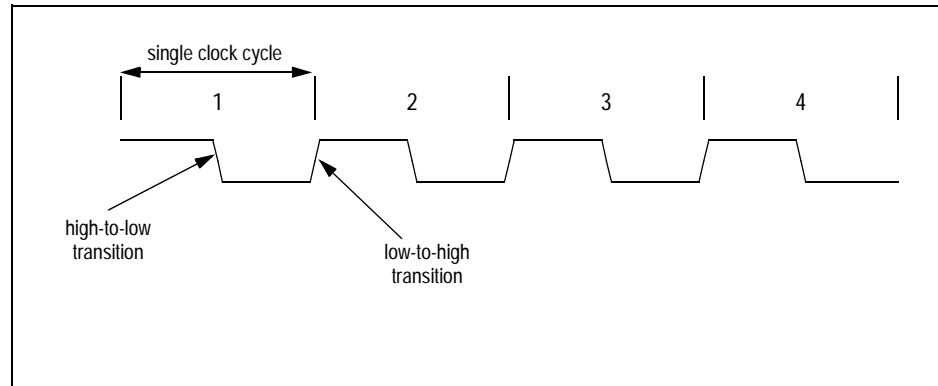


Figure 1 Signal Transitions

## 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[x..y]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

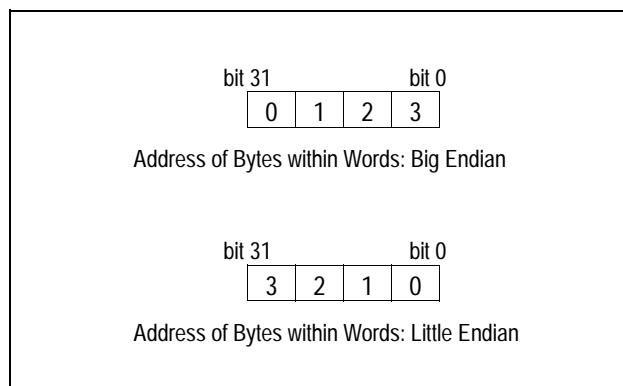


Figure 2 Example of Byte Ordering for “Big Endian” or “Little Endian” System Definition

## Register Terminology

Software in the context of this register terminology refers to modifications made by PCIe 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.
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.
Read and Write	RW	Software can both read and write bits with this attribute.

Table 2 Register Terminology (Sheet 1 of 2)

# Notes

Type	Abbreviation	Description
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	RWL	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
Write Transient	WT	The zero is always read from a bit/field of this type. Writing of a one is used to qualify the writing of other bits/fields in the same register.
Zero	Zero	A zero register or bit must be written with a value of zero and returns a value of zero when read.

Table 2 Register Terminology (Sheet 2 of 2)

## Use of Hypertext

In Chapter 9, Tables 9.2 and 9.3 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 the register section of the chapter. 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 1.1, PCI Special Interest Group.

PCI Power Management Interface Specification, Revision 1.1, PCI Special Interest Group.

PCI to PCI Bridge Architecture Specification, Revision 1.2, PCI Special Interest Group.

SMBus Specification, Revision 2.0.

## Revision History

**October 17, 2007:** Initial Publication.

**December 7, 2007:** Removed references to x8 and x4 port configurations from Chapter 2, Upstream Port Failover.

**December 14, 2007:** In Chapter 1, added second virtual channel capability and changed Device ID to 0x8034.

**December 26, 2007:** Changed device number from 22T16 to 22H16.

**February 7, 2008:** Removed references to x8 link width and port merging in Chapter 4.

**April 7, 2008:** Revised package type in Chapter 1 Features section to 484-ball Flip Chip BGA.

**April 15, 2008:** In Chapter 9, changed 0x0 definition for bit EEPE in SWPERCTL register from "time-out" to "end-to-end parity error".

**October 30, 2008:** Updated the following Description fields: LDIS in the PCIELCTL register, INTXD in PCICMD register, changed RO to RW for bits 10:9 in the HPCFGCTL register, SDOENERR in the AERUES register, DLLLA in both the PCIESTS and PCIELCAP registers, and added note in Description field for SWMODE field.



# Table of Contents

## Notes

### About This Manual

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

### PES22H16 Device Overview

Introduction .....	1-1
List of Features .....	1-1
Logic Diagram.....	1-3
System Identification.....	1-4
Vendor ID.....	1-4
Device ID .....	1-4
Revision ID .....	1-4
JTAG ID .....	1-4
SSID/SSVID.....	1-4
Device Serial Number Enhanced Capability.....	1-4
Pin Description.....	1-5
Pin Characteristics .....	1-12
Port Configuration .....	1-14
Disabled Ports .....	1-15

### Upstream Port Failover

Introduction .....	2-1
Failover .....	2-2
Static Upstream Port Failover.....	2-2
Dynamic Upstream Port Failover.....	2-3

### Clocking, Reset, and Initialization

Introduction .....	3-1
Initialization .....	3-3
Reset.....	3-4
Fundamental Reset .....	3-4
Hot Reset.....	3-6
Upstream Secondary Bus Reset .....	3-7
Downstream Secondary Bus Reset.....	3-8
Downstream Port Reset Outputs .....	3-8
Power Enable Controlled Reset Output.....	3-9
Power Good Controlled Reset Output .....	3-9

# Notes

## Link Operation

Introduction .....	4-1
Polarity Inversion .....	4-1
Link Width Negotiation .....	4-1
Link Retraining .....	4-1
Link Down .....	4-1
Slot Power Limit Support .....	4-2
Upstream Port .....	4-2
Downstream Port .....	4-2
Link States .....	4-2
Active State Power Management .....	4-3
Link Status .....	4-3

## General Purpose I/O

Introduction .....	5-1
GPIO Configuration .....	5-2
GPIO Pin Configured as an Input .....	5-2
GPIO Pin Configured as an Output .....	5-2
GPIO Pin Configured as an Alternate Function .....	5-2

## SMBus Interfaces

Introduction .....	6-1
Master SMBus Interface .....	6-2
Initialization .....	6-2
Serial EEPROM .....	6-2
I/O Expanders .....	6-6
Slave SMBus Interface .....	6-17
Initialization .....	6-17
SMBus Transactions .....	6-18

## Power Management

Introduction .....	7-1
PME Messages .....	7-2
Power Express Power Management Fence Protocol .....	7-2
Power Budgeting Capability .....	7-3

## Hot-Plug and Hot-Swap

Introduction .....	8-1
Hot-Plug I/O Expander .....	8-4
Hot-Plug Interrupts and Wake-up .....	8-4
Legacy System Hot-Plug Support .....	8-4
Hot-Swap .....	8-6

## Configuration Registers

Configuration Space Organization .....	9-1
Upstream Port (Port 0) .....	9-3
Downstream Ports (Ports 1 through 15) .....	9-8
Register Definitions .....	9-11
Type 1 Configuration Header Registers .....	9-11
PCI Express Capability Structure .....	9-22
Power Management Capability Structure .....	9-35

# Notes

Message Signaled Interrupt Capability Structure .....	9-36
Subsystem ID and Subsystem Vendor ID .....	9-38
Extended Configuration Space Access Registers .....	9-38
Advanced Error Reporting (AER) Enhanced Capability .....	9-39
Device Serial Number Enhanced Capability .....	9-45
PCI Express Virtual Channel Capability .....	9-46
Power Budgeting Enhanced Capability .....	9-57
Switch Control and Status Registers .....	9-59
Internal Switch Error Control and Status Registers .....	9-76

## JTAG Boundary Scan

Introduction .....	10-1
Test Access Point .....	10-1
Signal Definitions .....	10-1
Boundary Scan Chain .....	10-3
Test Data Register (DR) .....	10-5
Boundary Scan Registers .....	10-5
Instruction Register (IR) .....	10-7
EXTEST .....	10-8
SAMPLE/PRELOAD .....	10-8
BYPASS .....	10-8
CLAMP .....	10-9
IDCODE .....	10-9
VALIDATE .....	10-9
RESERVED .....	10-9
Usage Considerations .....	10-10

Notes



# List of Tables

## Notes

Table 1.1	PES22H16 Device IDs.....	1-4
Table 1.2	PES22H16 Revision ID.....	1-4
Table 1.3	PCI Express Interface Pins.....	1-5
Table 1.4	SMBus Interface Pins.....	1-7
Table 1.5	General Purpose I/O Pins.....	1-7
Table 1.6	System Pins.....	1-10
Table 1.7	Test Pins.....	1-11
Table 1.8	Power and Ground Pins.....	1-11
Table 1.9	Pin Characteristics.....	1-12
Table 3.1	Reference Clock Mode Encoding.....	3-1
Table 3.2	Boot Configuration Vector Signals.....	3-3
Table 5.1	General Purpose I/O Pin Alternate Function.....	5-1
Table 5.2	GPIO Pin Configuration.....	5-2
Table 6.1	Serial EEPROM SMBus Address.....	6-2
Table 6.2	PES22H16 Compatible Serial EEPROMs.....	6-3
Table 6.3	I/O Expander 0 Signals.....	6-10
Table 6.4	I/O Expander 1 Signals.....	6-11
Table 6.5	I/O Expander 2 Signals.....	6-11
Table 6.6	I/O Expander 3 Signals.....	6-12
Table 6.7	I/O Expander 4 Signals.....	6-13
Table 6.8	I/O Expander 5 Signals.....	6-13
Table 6.9	I/O Expander 6 Signals.....	6-14
Table 6.10	I/O Expander 7 Signals.....	6-15
Table 6.11	I/O Expander 8 Signals.....	6-15
Table 6.12	I/O Expander 9 Signals.....	6-16
Table 6.13	I/O Expander 10 Signals.....	6-17
Table 6.14	Slave SMBus Address When a Static Address is Selected.....	6-17
Table 6.15	Slave SMBus Command Code Fields.....	6-18
Table 6.16	CSR Register Read or Write Operation Byte Sequence.....	6-19
Table 6.17	CSR Register Read or Write CMD Field Description.....	6-20
Table 6.18	Serial EEPROM Read or Write Operation Byte Sequence.....	6-21
Table 6.19	Serial EEPROM Read or Write CMD Field Description.....	6-21
Table 7.1	PES22H16 Power Management State Transition Diagram.....	7-2
Table 8.1	Downstream Port Hot-Plug Signals.....	8-3
Table 9.1	Base Addresses for Port Configuration Space Registers.....	9-1
Table 9.2	Upstream Port 0 Configuration Space Registers.....	9-3
Table 9.3	Downstream Ports 1 through 15 Configuration Space Registers.....	9-8
Table 10.1	JTAG Pin Descriptions.....	10-2
Table 10.2	Boundary Scan Chain.....	10-3
Table 10.3	Instructions Supported by PES22H16's JTAG Boundary Scan.....	10-8
Table 10.4	System Controller Device Identification Register.....	10-9

Notes



# List of Figures

## Notes

Figure 1.1	PES22H16 Architectural Block Diagram .....	1-2
Figure 1.2	PES22H16 Logic Diagram .....	1-3
Figure 1.3	Port Configuration .....	1-15
Figure 2.1	Upstream Port Failover Architecture .....	2-1
Figure 3.1	Common Clock on Upstream and Downstream (option to enable or disable Spread Spectrum Clock) .....	3-1
Figure 3.2	Non-Common Clock on Upstream; Common Clock on Downstream (must disable Spread Spectrum Clock) .....	3-2
Figure 3.3	Common Clock on Upstream; Non-Common Clock on Downstream (must disable Spread Spectrum Clock) .....	3-2
Figure 3.4	Non-Common Clock on Upstream and Downstream (must disable Spread Spectrum Clock) .....	3-3
Figure 3.5	Fundamental Reset in Transparent Mode with Serial EEPROM Initialization .....	3-6
Figure 3.6	Power Enable Controlled Reset Output Mode Operation .....	3-9
Figure 3.7	Power Good Controlled Reset Output Mode Operation .....	3-9
Figure 4.1	PES22H16 ASPM Link State Transitions .....	4-3
Figure 6.1	SMBus Interface Configuration Examples .....	6-1
Figure 6.2	Sequential Double Word Initialization Sequence Format .....	6-4
Figure 6.3	Configuration Done Sequence Format .....	6-4
Figure 6.4	Serial EEPROM Initialization Errors .....	6-5
Figure 6.5	I/O Expander Function Allocation .....	6-6
Figure 6.6	I/O Expander Default Output Signal Value .....	6-7
Figure 6.7	Slave SMBus Command Code Format .....	6-18
Figure 6.8	CSR Register Read or Write CMD Field Format .....	6-20
Figure 6.9	Serial EEPROM Read or Write CMD Field Format .....	6-21
Figure 6.10	CSR Register Read Using SMBus Block Write/Read Transactions with PEC Disabled .....	6-22
Figure 6.11	Serial EEPROM Read Using SMBus Block Write/Read Transactions with PEC Disabled .....	6-23
Figure 6.12	CSR Register Write Using SMBus Block Write Transactions with PEC Disabled .....	6-23
Figure 6.13	Serial EEPROM Write Using SMBus Block Write Transactions with PEC Disabled .....	6-23
Figure 6.14	Serial EEPROM Write Using SMBus Block Write Transactions with PEC Enabled .....	6-23
Figure 6.15	CSR Register Read Using SMBus Read and Write Transactions with PEC Disabled .....	6-24
Figure 7.1	PES22H16 Power Management State Transition Diagram .....	7-1
Figure 8.1	Hot-Plug on Switch Downstream Slots Application .....	8-1
Figure 8.2	Hot-Plug with Switch on Add-In Card Application .....	8-2
Figure 8.3	Hot-Plug with Carrier Card Application .....	8-2
Figure 8.4	PES22H16 Hot-Plug Event Signalling .....	8-5
Figure 9.1	Port Configuration Space Organization .....	9-2
Figure 10.1	Diagram of the JTAG Logic .....	10-1
Figure 10.2	State Diagram of PES22H16's TAP Controller .....	10-2
Figure 10.3	Diagram of Observe-only Input Cell .....	10-6
Figure 10.4	Diagram of Output Cell .....	10-6
Figure 10.5	Diagram of Output Enable Cell .....	10-7
Figure 10.6	Device ID Register Format .....	10-9

Notes



# Register List

## Notes

AERCAP - AER Capabilities (0x100) .....	9-39
AERCEM - AER Correctable Error Mask (0x114) .....	9-44
AERCES - AER Correctable Error Status (0x110) .....	9-43
AERCTL - AER Control (0x118) .....	9-44
AERHL1DW - AER Header Log 1st Doubleword (0x11C) .....	9-45
AERHL2DW - AER Header Log 2nd Doubleword (0x120) .....	9-45
AERHL3DW - AER Header Log 3rd Doubleword (0x124) .....	9-45
AERHL4DW - AER Header Log 4th Doubleword (0x128) .....	9-45
AERUEM - AER Uncorrectable Error Mask (0x108) .....	9-40
AERUES - AER Uncorrectable Error Status (0x104) .....	9-40
AERUESV - AER Uncorrectable Error Severity (0x10C) .....	9-42
BAR0 - Base Address Register 0 (0x010) .....	9-15
BAR1 - Base Address Register 1 (0x014) .....	9-15
BCTL - Bridge Control Register (0x03E) .....	9-21
BIST - Built-in Self Test Register (0x00F) .....	9-15
CAPPTR - Capabilities Pointer Register (0x034) .....	9-20
CCODE - Class Code Register (0x009) .....	9-14
CLS - Cache Line Size Register (0x00C) .....	9-14
DID - Device Identification Register (0x002) .....	9-12
ECFGADDR - Extended Configuration Space Access Address (0x0F8) .....	9-38
ECFGDATA - Extended Configuration Space Access Data (0x0FC) .....	9-39
EEPROMINTF - Serial EEPROM Interface (0x42C) .....	9-66
EROMBASE - Expansion ROM Base Address Register (0x038) .....	9-20
GPECTL - General Purpose Event Control (0x450) .....	9-70
GPESTS - General Purpose Event Status (0x454) .....	9-72
GPIOCFG - General Purpose I/O Configuration (0x41C) .....	9-63
GPIOD - General Purpose I/O Data (0x420) .....	9-64
GPIOFUNC - General Purpose I/O Control Function (0x418) .....	9-63
GPR - General Purpose Register (0x40C) .....	9-63
HDR - Header Type Register (0x00E) .....	9-15
HPCFGCTL - Hot-Plug Configuration Control (0x408) .....	9-62
INTRLINE - Interrupt Line Register (0x03C) .....	9-20
INTRPIN - Interrupt PIN Register (0x03D) .....	9-21
IOBASE - I/O Base Register (0x01C) .....	9-16
IOBASEU - I/O Base Upper Register (0x030) .....	9-19
IOEXPADDR0 - SMBus I/O Expander Address 0 (0x434) .....	9-68
IOEXPADDR1 - SMBus I/O Expander Address 1 (0x438) .....	9-69
IOEXPADDR2 - SMBus I/O Expander Address 2 (0x43C) .....	9-69
IOEXPINTF - I/O Expander Interface (0x430) .....	9-67
IOLIMIT - I/O Limit Register (0x01D) .....	9-17
IOLIMITU - I/O Limit Upper Register (0x032) .....	9-20
MBASE - Memory Base Register (0x020) .....	9-18
MLIMIT - Memory Limit Register (0x022) .....	9-18
MSIADDR - Message Signaled Interrupt Address (0x0D4) .....	9-37
MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) .....	9-36
MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) .....	9-38
MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) .....	9-37
PBUSN - Primary Bus Number Register (0x018) .....	9-15
PCICMD - PCI Command Register (0x004) .....	9-12

# Notes

PCIECAP - PCI Express Capability (0x040) .....	9-22
PCIEDCAP - PCI Express Device Capabilities (0x044) .....	9-23
PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) .....	9-33
PCIEDCTL - PCI Express Device Control (0x048) .....	9-24
PCIEDCTL2 - PCI Express Device Control 2 (0x068) .....	9-33
PCIEDSTS - PCI Express Device Status (0x04A) .....	9-25
PCIEDSTS2 - PCI Express Device Status 2 (0x06A) .....	9-34
PCIELCAP - PCI Express Link Capabilities (0x04C) .....	9-26
PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) .....	9-34
PCIELCTL - PCI Express Link Control (0x050) .....	9-27
PCIELCTL2 - PCI Express Link Control 2 (0x070) .....	9-34
PCIELSTS - PCI Express Link Status (0x052) .....	9-28
PCIELSTS2 - PCI Express Link Status 2 (0x072) .....	9-34
PCIESCAP - PCI Express Slot Capabilities (0x054) .....	9-29
PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074) .....	9-34
PCIESCTL - PCI Express Slot Control (0x058) .....	9-31
PCIESCTL2 - PCI Express Slot Control 2 (0x078) .....	9-34
PCIESSTS - PCI Express Slot Status (0x05A) .....	9-32
PCIESSTS2 - PCI Express Slot Status 2 (0x07A) .....	9-35
PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200) .....	9-46
PCISTS - PCI Status Register (0x006) .....	9-13
PLTIMER - Primary Latency Timer (0x00D) .....	9-15
PMBASE - Prefetchable Memory Base Register (0x024) .....	9-18
PMBASEU - Prefetchable Memory Base Upper Register (0x028) .....	9-19
PMCAP - PCI Power Management Capabilities (0x0C0) .....	9-35
PMCSR - PCI Power Management Control and Status (0x0C4) .....	9-36
PMLIMIT - Prefetchable Memory Limit Register (0x026) .....	9-19
PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) .....	9-19
PVCCAP1 - Port VC Capability 1 (0x204) .....	9-46
PVCCAP2 - Port VC Capability 2 (0x208) .....	9-47
PVCCCTL - Port VC Control (0x20C) .....	9-47
PVCSTS - Port VC Status (0x20E) .....	9-48
PWRBCAP - Power Budgeting Capabilities (0x280) .....	9-57
PWRBD - Power Budgeting Data (0x288) .....	9-57
PWRBDSEL - Power Budgeting Data Select (0x284) .....	9-57
PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) .....	9-58
PWRBPBC - Power Budgeting Power Budget Capability (0x28C) .....	9-58
RID - Revision Identification Register (0x008) .....	9-14
SBUSN - Secondary Bus Number Register (0x019) .....	9-16
SECSTS - Secondary Status Register (0x01E) .....	9-17
SLTIMER - Secondary Latency Timer Register (0x01B) .....	9-16
SMBUSCTL - SMBus Control (0x428) .....	9-65
SMBUSSTS - SMBus Status (0x424) .....	9-64
SNUMCAP - Serial Number Capabilities (0x180) .....	9-45
SNUMLDW - Serial Number Lower Doubleword (0x184) .....	9-46
SNUMUDW - Serial Number Upper Doubleword (0x188) .....	9-46
SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) .....	9-38
SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) .....	9-38
SUBUSN - Subordinate Bus Number Register (0x01A) .....	9-16
SWCTL - Switch Control (0x404) .....	9-61
SWPECNT - Switch Parity Error Count (0x74C) .....	9-77
SWPECTL - Switch Parity Error Control (0x740) .....	9-76
SWPERCTL - Switch Parity Error Reporting Control (0x748) .....	9-77
SWPESTS - Switch Parity Error Status (0x744) .....	9-76
SWSTS - Switch Status (0x400) .....	9-59

# Notes

SWTOCNT - Switch Time-Out Count (0x75C) .....	9-79
SWTORCTL - Switch Time-Out Reporting Control (0x758) .....	9-78
SWTOSTS - Switch Time-Out Status (0x754) .....	9-77
USPFCTL - Upstream Port Failover Control (0x474) .....	9-75
USPFSTS - Upstream Port Failover Status (0x470) .....	9-74
USPFTIMER - Upstream Port Failover Watchdog Timer (0x478) .....	9-75
VCR0CAP - VC Resource 0 Capability (0x210) .....	9-48
VCR0CTL - VC Resource 0 Control (0x214) .....	9-49
VCR0STS - VC Resource 0 Status (0x218) .....	9-50
VCR0TBL0 - VC Resource 0 Arbitration Table Entry 0 (0x230) .....	9-52
VCR0TBL1 - VC Resource 0 Arbitration Table Entry 1 (0x234) .....	9-53
VCR0TBL2 - VC Resource 0 Arbitration Table Entry 2 (0x238) .....	9-53
VCR0TBL3 - VC Resource 0 Arbitration Table Entry 3 (0x23C) .....	9-54
VCR1CAP - VC Resource 1 Capability (0x21C) .....	9-50
VCR1CTL - VC Resource 1 Control (0x220) .....	9-51
VCR1STS - VC Resource 1 Status (0x224) .....	9-51
VCR1TBL0 - VC Resource 1 Arbitration Table Entry 0 (0x240) .....	9-54
VCR1TBL1 - VC Resource 1 Arbitration Table Entry 1 (0x244) .....	9-55
VCR1TBL2 - VC Resource 1 Arbitration Table Entry 2 (0x248) .....	9-55
VCR1TBL3 - VC Resource 1 Arbitration Table Entry 3 (0x24C) .....	9-56
VID - Vendor Identification Register (0x000) .....	9-11

Notes



# PES22H16 Device Overview

## Notes

## Introduction

The 89HPES22H16 is a member of the IDT PRECISE™ family of PCI Express® switching solutions. The PES22H16 is a 22-lane, 16-port peripheral chip that performs PCI Express packet switching with a feature set optimized for high-performance applications such as servers, storage, and communications/networking. It provides connectivity and switching functions between a PCI Express upstream port and up to fifteen downstream ports and supports switching between downstream ports.

## List of Features

- ◆ **High Performance PCI Express Switch**
  - Sixteen maximum switch ports
    - Two x4 ports
    - Fourteen x1 ports
  - Twenty-two 2.5 Gbps embedded SerDes
    - Supports pre-emphasis and receive equalization on per-port basis
  - Delivers 88 Gbps (11 GBps) of aggregate switching capacity
  - Low-latency cut-through switch architecture
  - Support for Max Payload Size up to 2048 bytes
  - Supports two virtual channels and eight traffic classes
  - PCI Express Base Specification Revision 1.1 compliant
- ◆ **Flexible Architecture with Numerous Configuration Options**
  - Port arbitration schemes utilizing round robin algorithms
  - Automatic per port link width negotiation from x4 to x2 or x1
  - Automatic lane reversal on all ports
  - Automatic polarity inversion on all lanes
  - Supports locked transactions, allowing use with legacy software
  - Ability to load device configuration from serial EEPROM
  - Ability to control device via SMBus
- ◆ **Highly Integrated Solution**
  - Requires no external components
  - Incorporates on-chip internal memory for packet buffering and queueing
  - Integrates twenty-two 2.5 Gbps embedded full duplex SerDes, 8B/10B encoder/decoder (no separate transceivers needed)
- ◆ **Reliability, Availability, and Serviceability (RAS) Features**
  - Redundant upstream port failover capability
  - Supports optional PCI Express end-to-end CRC checking
  - Internal end-to-end parity protection on all TLPs ensures data integrity even in systems that do not implement end-to-end CRC (ECRC)
  - Supports optional PCI Express Advanced Error Reporting
  - Supports PCI Express Hot-Plug
    - Compatible with Hot-Plug I/O expanders used on PC motherboards
  - Supports Hot-Swap

◆ Power Management

- Supports PCI Power Management Interface specification, Revision 1.1 (PCI-PM)
  - Supports powerdown modes at the link level (L0, L0s, L1, L2/L3 Ready and L3) and at the device level (D0, D3<sub>hot</sub>)
- Unused SerDes disabled

◆ Testability and Debug Features

- Built in SerDes Pseudo-Random Bit Stream (PRBS) generator
- Ability to read and write any internal register via the SMBus
- Ability to bypass link training and force any link into any mode
- Provides statistics and performance counters

◆ Thirty-two General Purpose Input/Output pins

- Each pin may be individually configured as an input or output
- Each pin may be individually configured as an interrupt input
- Some pins have selectable alternate functions

◆ Packaged in a 23mm x 23mm 484-ball Flip Chip BGA with 1mm ball spacing

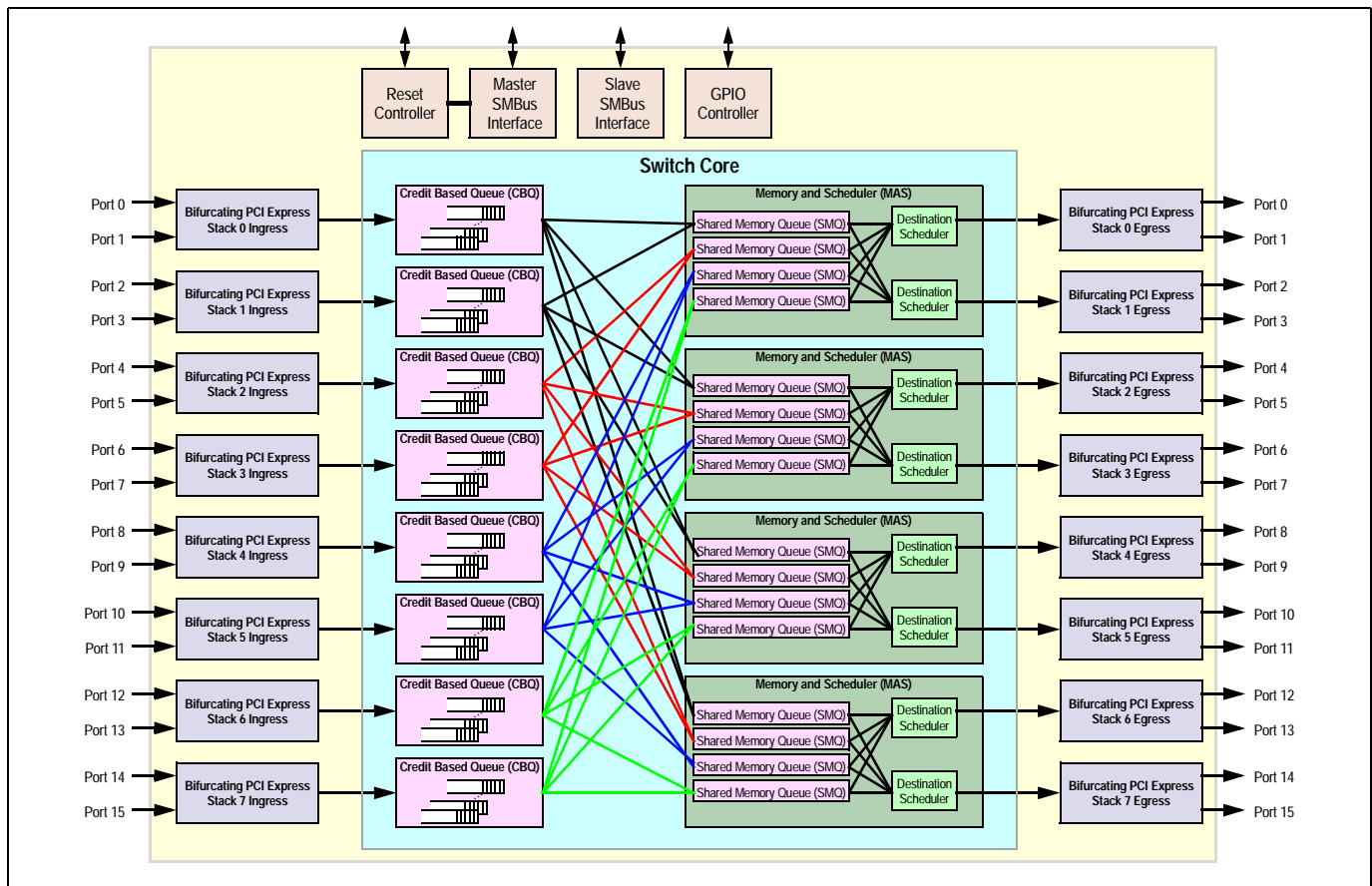


Figure 1.1 PES22H16 Architectural Block Diagram

# Logic Diagram

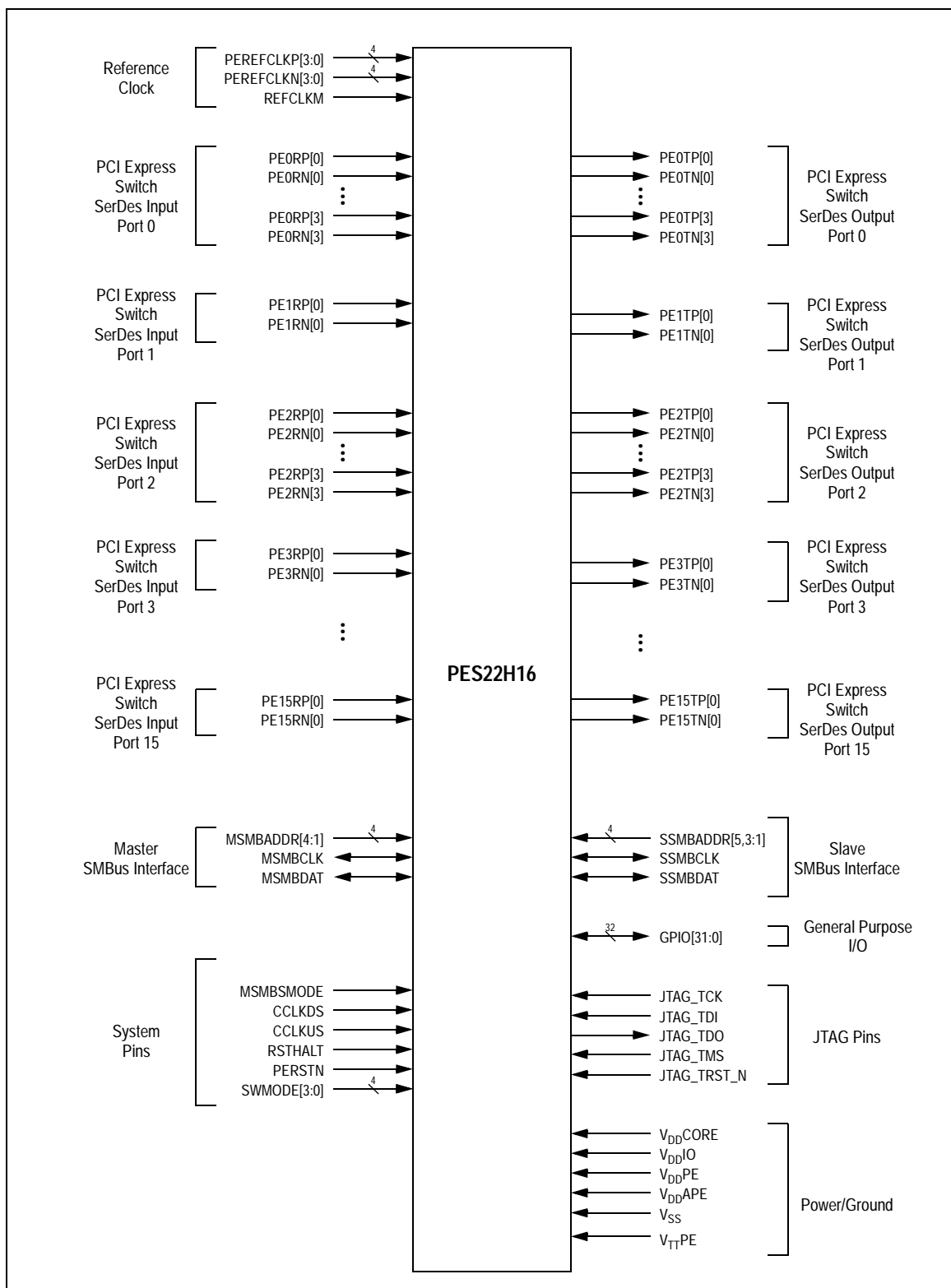


Figure 1.2 PES22H16 Logic Diagram

## Notes

## System Identification

### Vendor ID

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

### Device ID

The PES22H16 device ID is shown in Table 1.1.

PCIe Device	Device ID
0x4	0x8034

**Table 1.1 PES22H16 Device IDs**

### Revision ID

The revision ID in the PES22H16 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.

Revision ID	Description
0x0	Corresponds to ZA silicon

**Table 1.2 PES22H16 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 PES22H16 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 PES22H16 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.

## Notes

## Pin Description

The following tables lists the functions of the pins provided on the PES22H16. Some of the functions listed may be multiplexed onto the same pin. The active polarity of a signal is defined using a suffix. Signals ending with an "N" are defined 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. Differential signals end with a suffix "N" or "P." The differential signal ending in "P" is the positive portion of the differential pair and the differential signal ending in "N" is the negative portion of the differential pair.

Signal	Type	Name/Description
PE0RP[3:0] PE0RN[3:0]	I	<b>PCI Express Port 0 Serial Data Receive.</b> Differential PCI Express receive pairs for port 0. Port 0 is the upstream port.
PE0TP[3:0] PE0TN[3:0]	O	<b>PCI Express Port 0 Serial Data Transmit.</b> Differential PCI Express transmit pairs for port 0. Port 0 is the upstream port.
PE1RP[0] PE1RN[0]	I	<b>PCI Express Port 1 Serial Data Receive.</b> Differential PCI Express receive pair for port 1.
PE1TP[0] PE1TN[0]	O	<b>PCI Express Port 1 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 1.
PE2RP[3:0] PE2RN[3:0]	I	<b>PCI Express Port 2 Serial Data Receive.</b> Differential PCI Express receive pairs for port 2.
PE2TP[3:0] PE2TN[3:0]	O	<b>PCI Express Port 2 Serial Data Transmit.</b> Differential PCI Express transmit pairs for port 2.
PE3RP[0] PE3RN[0]	I	<b>PCI Express Port 3 Serial Data Receive.</b> Differential PCI Express receive pair for port 3.
PE3TP[0] PE3TN[0]	O	<b>PCI Express Port 3 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 3.
PE4RP[0] PE4RN[0]	I	<b>PCI Express Port 4 Serial Data Receive.</b> Differential PCI Express receive pair for port 4.
PE4TP[0] PE4TN[0]	O	<b>PCI Express Port 4 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 4.
PE5RP[0] PE5RN[0]	I	<b>PCI Express Port 5 Serial Data Receive.</b> Differential PCI Express receive pair for port 5.
PE5TP[0] PE5TN[0]	O	<b>PCI Express Port 5 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 5.
PE6RP[0] PE6RN[0]	I	<b>PCI Express Port 6 Serial Data Receive.</b> Differential PCI Express receive pair for port 6.
PE6TP[0] PE6TN[0]	O	<b>PCI Express Port 6 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 6.
PE7RP[0] PE7RN[0]	I	<b>PCI Express Port 7 Serial Data Receive.</b> Differential PCI Express receive pair for port 7.
PE7TP[0] PE7TN[0]	O	<b>PCI Express Port 7 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 7.
PE8RP[0] PE8RN[0]	I	<b>PCI Express Port 8 Serial Data Receive.</b> Differential PCI Express receive pair for port 8.
PE8TP[0] PE8TN[0]	O	<b>PCI Express Port 8 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 8.

Table 1.3 PCI Express Interface Pins (Part 1 of 2)

# Notes

Signal	Type	Name/Description
PE9RP[0] PE9RN[0]	I	<b>PCI Express Port 9 Serial Data Receive.</b> Differential PCI Express receive pair for port 9.
PE9TP[0] PE9TN[0]	O	<b>PCI Express Port 9 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 9.
PE10RP[0] PE10RN[0]	I	<b>PCI Express Port 10 Serial Data Receive.</b> Differential PCI Express receive pair for port 10.
PE10TP[0] PE10TN[0]	O	<b>PCI Express Port 10 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 10.
PE11RP[0] PE11RN[0]	I	<b>PCI Express Port 11 Serial Data Receive.</b> Differential PCI Express receive pair for port 11.
PE11TP[0] PE11TN[0]	O	<b>PCI Express Port 11 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 11.
PE12RP[0] PE12RN[0]	I	<b>PCI Express Port 12 Serial Data Receive.</b> Differential PCI Express receive pair for port 12.
PE12TP[0] PE12TN[0]	O	<b>PCI Express Port 12 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 12.
PE13RP[0] PE13RN[0]	I	<b>PCI Express Port 13 Serial Data Receive.</b> Differential PCI Express receive pair for port 13.
PE13TP[0] PE13TN[0]	O	<b>PCI Express Port 13 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 13. W
PE14RP[0] PE14RN[0]	I	<b>PCI Express Port 14 Serial Data Receive.</b> Differential PCI Express receive pair for port 14.
PE14TP[0] PE14TN[0]	O	<b>PCI Express Port 14 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 14.
PE15RP[0] PE15RN[0]	I	<b>PCI Express Port 15 Serial Data Receive.</b> Differential PCI Express receive pair for port 15.
PE15TP[0] PE15TN[0]	O	<b>PCI Express Port 15 Serial Data Transmit.</b> Differential PCI Express transmit pair for port 15.
REFCLKM	I	<b>PCI Express Reference Clock Mode Select.</b> This signal selects the frequency of the reference clock input. 0x0 - 100 MHz 0x1 - 125 MHz
PEREFCLKP[3:0] PEREFCLKN[3:0]	I	<b>PCI Express Reference Clock.</b> Differential reference clock pair input. This clock is used as the reference clock by on-chip PLLs to generate the clocks required for the system logic and on-chip SerDes. The frequency of the differential reference clock is determined by the REFCLKM signal.

Table 1.3 PCI Express Interface Pins (Part 2 of 2)

# Notes

Signal	Type	Name/Description
MSMBADDR[4:1]	I	<b>Master SMBus Address.</b> These pins determine the SMBus address of the serial EEPROM from which configuration information is loaded.
MSMBCLK	I/O	<b>Master SMBus Clock.</b> This bidirectional signal is used to synchronize transfers on the master SMBus. It is active and generating the clock only when the EEPROM or I/O Expanders are being accessed.
MSMBDAT	I/O	<b>Master SMBus Data.</b> This bidirectional signal is used for data on the master SMBus.
SSMBADDR[5,3:1]	I	<b>Slave SMBus Address.</b> These pins determine the SMBus address to which the slave SMBus interface responds.
SSMBCLK	I/O	<b>Slave SMBus Clock.</b> This bidirectional signal is used to synchronize transfers on the slave SMBus.
SSMBDAT	I/O	<b>Slave SMBus Data.</b> This bidirectional signal is used for data on the slave SMBus.

**Table 1.4 SMBus Interface Pins**

Signal	Type	Name/Description
GPIO[0]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[1]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[2]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[3]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[4]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[5]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: GPEN Alternate function pin type: Output Alternate function: General Purpose Event (GPE) output
GPIO[6]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P1RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 1
GPIO[7]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P2RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 2
GPIO[8]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P3RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 3

**Table 1.5 General Purpose I/O Pins (Part 1 of 4)**

## Notes

Signal	Type	Name/Description
GPIO[9]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P4RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 4
GPIO[10]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P5RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 5
GPIO[11]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P6RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 6
GPIO[12]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P7RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 7
GPIO[13]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P8RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 8
GPIO[14]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P9RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 9
GPIO[15]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P10RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 10
GPIO[16]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P11RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 11
GPIO[17]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P12RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 12
GPIO[18]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P13RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 13

Table 1.5 General Purpose I/O Pins (Part 2 of 4)

# Notes

Signal	Type	Name/Description
GPIO[19]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P14RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 14
GPIO[20]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: P15RSTN Alternate function pin type: Output Alternate function: Reset output for downstream port 15
GPIO[21]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN0 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 0
GPIO[22]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN1 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 1
GPIO[23]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN2 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 2
GPIO[24]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN3 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 3
GPIO[25]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN4 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 4
GPIO[26]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN5 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 5
GPIO[27]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN6 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 6
GPIO[28]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN7 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 7

Table 1.5 General Purpose I/O Pins (Part 3 of 4)

## Notes

Signal	Type	Name/Description
GPIO[29]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[30]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin.
GPIO[31]	I/O	<b>General Purpose I/O.</b> This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN10 Alternate function pin type: Input Alternate function: SMBus I/O expander interrupt 10

**Table 1.5 General Purpose I/O Pins (Part 4 of 4)**

Signal	Type	Name/Description
CCLKDS	I	<b>Common Clock Downstream.</b> When the CCLKDS pin is asserted, it indicates that a common clock is being used between the downstream device and the downstream port.
CCLKUS	I	<b>Common Clock Upstream.</b> When the CCLKUS pin is asserted, it indicates that a common clock is being used between the upstream device and the upstream port.
MSMBSMODE	I	<b>Master SMBus Slow Mode.</b> The assertion of this pin indicates that the master SMBus should operate at 100 KHz instead of 400 KHz. This value may not be overridden.
PERSTN	I	<b>Fundamental Reset.</b> Assertion of this signal resets all logic inside the PES22H16 and initiates a PCI Express fundamental reset.
RSTHALT	I	<b>Reset Halt.</b> When this signal is asserted during a PCI Express fundamental reset, the PES22H16 executes the reset procedure and remains in a 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 reset state when the RSTHALT bit is cleared in the PA_SWCTL register by an SMBus master.
SWMODE[3:0]	I	<b>Switch Mode.</b> These configuration pins determine the PES22H16 switch operating mode. These pins should be static and not change following the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 through 0x7 - Reserved 0x8 - Normal switch mode with upstream port failover (port 0 selected as the upstream port) 0x9 - Normal switch mode with upstream port failover (port 2 selected as the upstream port) 0xA - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 0 selected as the upstream port) 0xB - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 2 selected as the upstream port) 0xC through 0xF - Reserved

**Table 1.6 System Pins**

# Notes

Signal	Type	Name/Description
JTAG_TCK	I	<b>JTAG Clock.</b> This is an input test clock used to clock the shifting of data into or out of the boundary scan logic or JTAG Controller. JTAG_TCK is independent of the system clock with a nominal 50% duty cycle.
JTAG_TDI	I	<b>JTAG Data Input.</b> This is the serial data input to the boundary scan logic or JTAG Controller.
JTAG_TDO	O	<b>JTAG Data Output.</b> This is the serial data shifted out from the boundary scan logic or JTAG Controller. When no data is being shifted out, this signal is tri-stated.
JTAG_TMS	I	<b>JTAG Mode.</b> The value on this signal controls the test mode select of the boundary scan logic or JTAG Controller.
JTAG_TRST_N	I	<b>JTAG Reset.</b> This active low signal asynchronously resets the boundary scan logic and JTAG TAP Controller. An external pull-up on the board is recommended to meet the JTAG specification in cases where the tester can access this signal. However, for systems running in functional mode, one of the following should occur: 1) actively drive this signal low with control logic 2) statically drive this signal low with an external pull-down on the board

**Table 1.7 Test Pins**

Signal	Type	Name/Description
V <sub>DD</sub> CORE	I	<b>Core VDD.</b> Power supply for core logic.
V <sub>DD</sub> I/O	I	<b>I/O VDD.</b> LVTTTL I/O buffer power supply.
V <sub>DD</sub> PE	I	<b>PCI Express Digital Power.</b> PCI Express digital power used by the digital power of the SerDes.
V <sub>DD</sub> APE	I	<b>PCI Express Analog Power.</b> PCI Express analog power used by the PLL and bias generator.
V <sub>SS</sub>	I	<b>Ground.</b>
V <sub>TT</sub> PE		<b>PCI Express Serial Data Transmit Termination Voltage.</b> This pin allows the driver termination voltage to be set, enabling the system designer to control the Common Mode Voltage and output voltage swing of the corresponding PCI Serial Data Transmit differential pair.

**Table 1.8 Power and Ground Pins**

# Notes

## Pin Characteristics

**Note:** Some input pads of the PES22H16 do not contain internal pull-ups or pull-downs. Unused inputs should be tied off to appropriate levels. This is especially critical for unused control signal inputs which, if left floating, could adversely affect operation. Also, any input pin left floating can cause a slight increase in power consumption.

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes
PCI Express Interface	PE0RN[3:0]	I	CML	Serial Link		
	PE0RP[3:0]	I				
	PE0TN[3:0]	O				
	PE0TP[3:0]	O				
	PE1RN[0]	I				
	PE1RP[0]	I				
	PE1TN[0]	O				
	PE1TP[0]	O				
	PE2RN[3:0]	I				
	PE2RP[3:0]	I				
	PE2TN[3:0]	O				
	PE2TP[3:0]	O				
	PE3RN[0]	I				
	PE3RP[0]	I				
	PE3TN[0]	O				
	PE3TP[0]	O				
	PE4RN[0]	I				
	PE4RP[0]	I				
	PE4TN[0]	O				
	PE4TP[0]	O				
	PE5RN[0]	I				
	PE5RP[0]	I				
	PE5TN[0]	O				
	PE5TP[0]	O				
	PE6RN[0]	I				
	PE6RP[0]	I				
	PE6TN[0]	O				
	PE6TP[0]	O				
	PE7RN[0]	I				
	PE7RP[0]	I				
	PE7TN[0]	O				
	PE7TP[0]	O				
	PE8RN[0]	I				

Table 1.9 Pin Characteristics (Part 1 of 3)

# Notes

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes
PCI Express Interface (cont.)	PE8RP[0]	I	CML	Serial Link		
	PE8TN[0]	O				
	PE8TP[0]	O				
	PE9RN[0]	I				
	PE9RP[0]	I				
	PE9TN[0]	O				
	PE9TP[0]	O				
	PE10RN[0]	I				
	PE10RP[0]	I				
	PE10TN[0]	O				
	PE10TP[0]	O				
	PE11RN[0]	I				
	PE11RP[0]	I				
	PE11TN[0]	O				
	PE11TP[0]	O				
	PE12RN[0]	I				
	PE12RP[0]	I				
	PE12TN[0]	O				
	PE12TP[0]	O				
	PE13RN[0]	I				
	PE13RP[0]	I				
	PE13TN[0]	O				
	PE13TP[0]	O				
	PE14RN[0]	I				
	PE14RP[0]	I				
	PE14TN[0]	O				
	PE14TP[0]	O				
	PE15RN[0]	I				
	PE15RP[0]	I				
	PE15TN[0]	O				
	PE15TP[0]	O				
	PEREF-CLKN[3:0]	I	LVPECL/CML	Diff. Clock Input		Refer to Table 9 in the PES22H16 Data Sheet
	PEREF-CLKP[3:0]	I				
	REFCLKM	I	LVTTL	Input	pull-down	

Table 1.9 Pin Characteristics (Part 2 of 3)

## Notes

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes
SMBus	MSMBADDR[4:1]	I	LVTTL		pull-up	
	MSMBCLK	I/O		STI <sup>1</sup>		
	MSMBDAT	I/O		STI		
	SSMBADDR[5,3:1]	I			pull-up	
	SSMBCLK	I/O		STI		
	SSMBDAT	I/O		STI		
General Purpose I/O	GPIO[31:0]	I/O	LVTTL		pull-up	
System Pins	CCLKDS	I	LVTTL	Input	pull-up	
	CCLKUS	I			pull-up	
	MSMBSMODE	I			pull-down	
	PERSTN	I				
	RSTHALT	I			pull-down	
	SWMODE[3:0]	I			pull-down	
EJTAG / JTAG	JTAG_TCK	I	LVTTL	STI	pull-up	
	JTAG_TDI	I		STI	pull-up	
	JTAG_TDO	O				
	JTAG_TMS	I		STI	pull-up	
	JTAG_TRST_N	I		STI	pull-up	External pull-down

**Table 1.9 Pin Characteristics (Part 3 of 3)**

<sup>1</sup> Schmitt Trigger Input (STI).

## Port Configuration

The PES22H16 contains two x4 ports and fourteen x1 ports labeled 0 through 15. Port 0 is always the upstream port and ports 1 through 15 are always downstream ports. Figure 1.3 illustrates the port configuration for the PES22H16.

# Notes

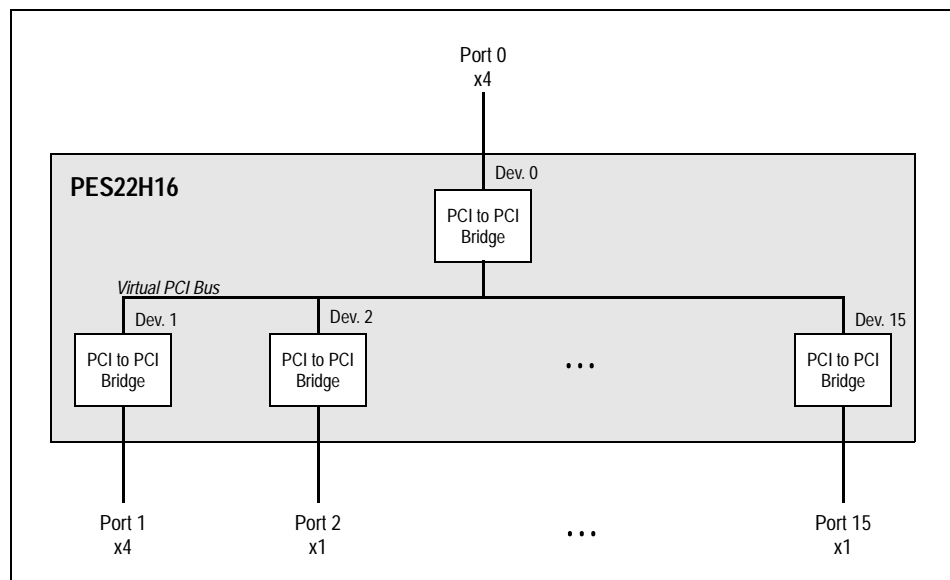


Figure 1.3 Port Configuration

## Disabled Ports

The PES22H16 may be configured to disable one or more ports. Disabling a port has the following effect.

- All of the output signals associated with a disabled port remain in a negated state (e.g., hot plug outputs, link status signals, port reset output, etc.)
- All input signals associated with a disabled port are ignored by the PES22H16 and have no effect on its operation.
- Configuration read or write transactions to a device that corresponds to a disabled port on the PES22H16's virtual PCI bus are treated by the upstream port (port 0) as an unsupported request (i.e., the device no longer exists).

All registers associated with a disabled port become inaccessible via the SMBus. Reading or writing a disabled port's register has an undefined effect.

Notes



# Upstream Port Failover

## Notes

## Introduction

The PES22H16 supports an upstream port failover mechanism that enables the construction of fault tolerant systems. Upstream port failover allows port 0 or port 2 to be selected as the upstream switch port.

The failover feature is disabled by default. To enable upstream port failover, an upstream port failover switch mode must be selected in the boot configuration vector during a fundamental reset. The following switch modes (SWMODE[3:0]) enable this feature.

- Normal switch mode with upstream port failover (port 0 selected as the upstream port)
- Normal switch mode with upstream port failover (port 2 selected as the upstream port)
- Normal switch mode with Serial EEPROM initialization and upstream port failover (port 0 selected as the upstream port)
- Normal switch mode with Serial EEPROM initialization and upstream port failover (port 2 selected as the upstream port)

A graphical representation of the upstream port failover architecture is shown in Figure 2.1.

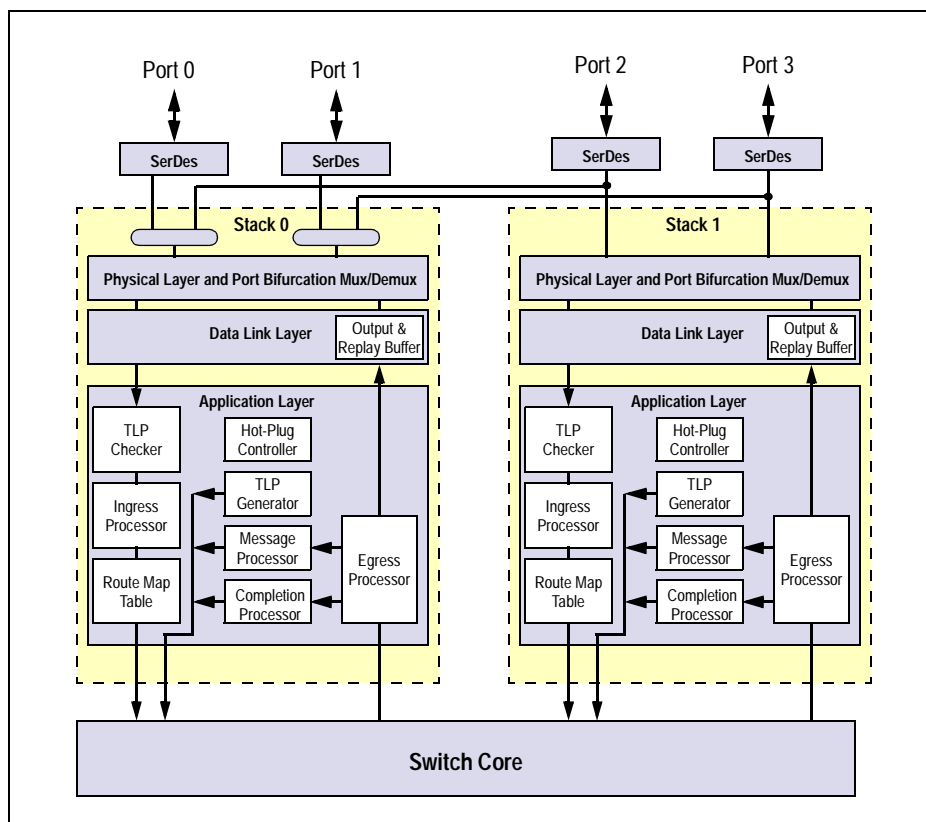


Figure 2.1 Upstream Port Failover Architecture

Stack 0 is always associated with the upstream port. In normal mode, SerDes lanes associated with port 0 are fed into stack zero. In failover mode, the SerDes lanes associated with port 0 or port 2 may be fed into stack zero. Thus, from an external perspective, it appears as though the upstream port can be moved from port 0 to port 2; however, in reality, all that is occurring is that the SerDes lanes associated with port 0 or port 2 are multiplexed into stack zero (i.e., the stack associated with the upstream port).

## Notes

When the PES22H16 is configured to operate in an upstream port failover switch mode, port 2 is always disabled, but there is no effect on the operation of ports 1 and 3 which operate as normal. The behavior of a disabled port is described in the section entitled section Disabled Ports on page 1-15. Upstream port resources (e.g., link status control LED outputs) are always associated with port 0 regardless of the external PES22H16 port associated with the upstream port.

## Failover

An upstream port failover may be initiated statically through a fundamental reset or dynamically while the system is running.

At a system level, a dynamic upstream port failover appears as a full link retrain of the upstream port, i.e., the Link State Sequence State Machine (LTSSM) transitions to the Detect state, and the data link layer transitions to a DL\_Down state. This typically results in a hot-reset of the PES22H16 and devices below the PES22H16 in the PCIe hierarchy. A hot-reset due to a DL\_Down state may be disabled by setting the Disable Link Down Hot Reset (DLDHRST) bit in the Switch Control (SWCTL) register. With this bit set, an upstream port failover does not reset the PES22H16 or the PCIe hierarchy and system state is preserved.

When a dynamic upstream port failover occurs, upstream port data queued in the switch, data being transmitted, and data in the replay buffer may be lost. Thus, some interruption of PCIe traffic should be expected with an upstream port failover. When the switch is configured to operate in a mode that supports upstream port failover, the Upstream Port Failover Enabled (USPFEN) bit is set in the Upstream Port Failover Status (USPFSTS) register.

In all switch modes, the current external PES22H16 switch port associated with the upstream port may be determined by reading the Current Upstream Port (CUSP) field in the USPFSTS register. Whenever a dynamic upstream port failover occurs, the Upstream Port Change (USPC) bit is set in the USPFSTS register. This bit is sticky and thus its status is preserved across a hot-reset.

The operation of the upstream port failover mechanism is unaffected by a hot-reset. Fields in the USPFEN register have no effect on system operation when an upstream port failover switch mode has not been selected during a fundamental reset.

## Static Upstream Port Failover

A static upstream port failover requires a fundamental reset to be initiated whenever the upstream port is changed. Since the initial upstream port is selected by the switch mode in the boot configuration vector, the static upstream port failover feature may be viewed as nothing more than the ability to select the upstream port during a fundamental reset.

A static upstream port failover consists of the following steps:

- Assert the PCIe fundamental reset signal (PERSTN)
- Modify the switch mode (SWMODE) signals to select the external PES22H16 port associated with the upstream port.
- Negate the PCIe fundamental reset signal (PERSTN).

The following switch modes select port 0 as the upstream port during a fundamental reset:

- Normal switch mode
- Normal switch mode with Serial EEPROM initialization
- Normal switch mode with upstream port failover (port 0 selected as the upstream port)
- Normal switch mode with Serial EEPROM initialization and upstream port failover (port 0 selected as the upstream port).

The following switch modes select port 2 as the upstream port during a fundamental reset:

- Normal switch mode with upstream port failover (port 2 selected as the upstream port)
- Normal switch mode with Serial EEPROM initialization and upstream port failover (port 2 selected as the upstream port)

Since initiation of an upstream port failover requires a fundamental reset of the PCIe hierarchy, many systems may require the use of dynamic upstream port failover.

## Notes

### Dynamic Upstream Port Failover

Dynamic upstream port failover allows the external PES22H16 port associated with the upstream port to be modified while the system is live and in a manner which preserves the system state.

When a dynamic upstream port failover is initiated, the PES22H16 takes the following actions:

- The LTSSM associated with the upstream port immediately transitions to the Detect state and the data link layer transitions to the DL\_Down state. This causes data in the replay buffer associated with the upstream port and upstream port data queued in the switch core to be discarded.
- The state of the SerDes multiplexors are modified to select the new upstream port.
- The LTSSM initiates link training with the upstream port.

The following sections describe the manner in which a dynamic upstream port failover may be initiated. In most systems it is expected that only one upstream failover mechanism will be enabled at a time. If a failover of the same type (i.e., software, signal, or watchdog timer) is initiated while one is already in progress, then the second initiation will be lost. If a failover of a different type is initiated while one is in progress, then the second failover will be performed when the one in progress is completed. Software may utilize the Upstream Failover Change (USPC) and Upstream Failover Change Initiated (USPCI) bits in the Upstream Port Failover Status (USPFSTS) register to avoid failover race conditions.

### Software Initiated Failover

An upstream port failover may be initiated by modifying the state of the Upstream Port Software Select (USPSEL) field in the Upstream Port Failover Control (USPFCTL) register when the switch is selected to operate in an upstream failover mode. A software initiated failover may be instituted by software running on the root or software running on the device that writes to the USPSEL field via the SMBus. The USPSEL field should not be modified during an upstream port failover (i.e., failover requests are not queued).

### Signal Initiated Failover

An upstream port failover may be initiated by a change in the state of the Upstream Port Select (USPSEL) signal. Such a failover is initiated when:

1. the Upstream Port Signal Failover Enable (SIGFEN) bit is set in the Upstream Port Failover Control (USPFCTL) register
2. the switch is selected to operate in an upstream failover mode, and
3. the upstream port selected by the USPSEL signal differs from the current upstream port.

The USPSEL signal is an alternate function of GPIO[4]. The state of the USPSEL signal always reflects the state of the GPIO[4] pin regardless of whether or not GPIO[4] is configured to operate as an alternate function. When USPSEL is negated (low), port 0 is selected as the upstream port. When USPSEL is asserted (high), port 2 is selected as the upstream port. The state of the USPSEL signal should not be modified more frequently than once per second. The behavior of the PES22H16 is undefined when the USPSEL signal is modified more frequently than this rate.

### Watchdog Timer Initiated Failover

An upstream port failover may be initiated as the result of an expiration of a watchdog timer. Such a failover is initiated when:

1. the Upstream Port Timer Failover Enable (TIMFEN) bit is set in the Upstream Port Failover Control (USPFCTL) register
2. the switch is selected to operate in an upstream failover mode, and
3. the Watchdog Timer Count (COUNT) field in the Upstream Port Failover Watchdog Timer (USPFTIMER) transitions from a one to a zero.

When non-zero, the COUNT field in the USPFTIMER is decremented once per microsecond (1  $\mu$ S). This provides a maximum watchdog timer interval 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 a hot-reset.

## Notes

When a watchdog timer failover is initiated, the new upstream port becomes the one that is not selected by the CUSP field in the USPFSTS register. For example, if the current upstream port is port 0, then the new upstream port following the failover is port 2.

When the count reaches zero, it remains at zero and no switchover will occur. When the count is updated with a non-zero value, switchover is again enabled.



# Clocking, Reset, and Initialization

## Notes

## Introduction

The PES22H16 has four differential reference clock inputs that are used internally to generate all of the clocks required by the internal switch logic and the SerDes. It is recommended that all reference clock input pairs be driven from a common clock source. The frequency of the reference clock inputs may be selected by the Reference Clock Mode Select (REFCLKM) input.

REFCLKM	Description
0	100 MHz reference clock input.
1	125 MHz reference clock input.

Table 3.1 Reference Clock Mode Encoding

Each PES22H16 port has an associated PLL. Each of the reference clock differential inputs feeds four on-chip PLLs (i.e., one quarter of the ports). Each PLL generates a 2.5 GHz clock which is used by four SerDes lanes and produces a 250 MHz core clock. The 250 MHz core clock output from the upstream port (i.e., port 0) is used as the system clock for internal switch logic<sup>1</sup>. When the switch is placed in PLL Bypass test mode via the SWMODE pins, the 250 MHz clock generated by the PLL is bypassed and the reference clock input on REFCLKP[0]/REFCLKN[0] is used for the core logic.

## Clock Operation

When the CCLKUS and CCLKDS pins are asserted, they indicate that a common clock is being used between the upstream device and the upstream port, as well as between the downstream devices and the downstream ports. The Spread Spectrum Clock (SSC) must be disabled when the non-common clock is used on either the upstream port or downstream port. Figures 3.1 through 3.4 illustrate the operation of the CCLKUS and CCLKDS clocks using a common clock and a non-common clock.

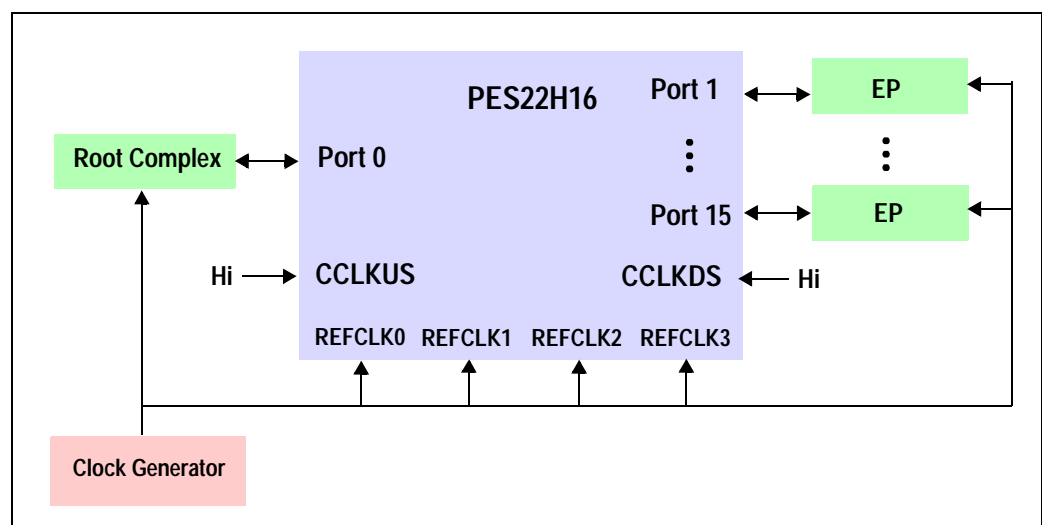


Figure 3.1 Common Clock on Upstream and Downstream (option to enable or disable Spread Spectrum Clock)

<sup>1</sup> The port 0 PLL is used to generate the 250 MHz core clock even when upstream port failover is enabled and regardless of the selected upstream port.

Notes

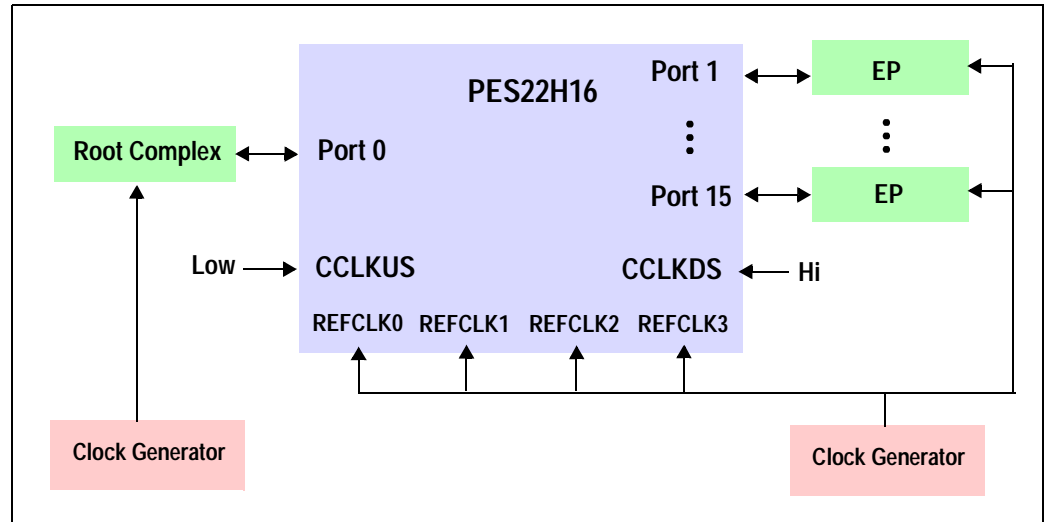


Figure 3.2 Non-Common Clock on Upstream; Common Clock on Downstream (must disable Spread Spectrum Clock)

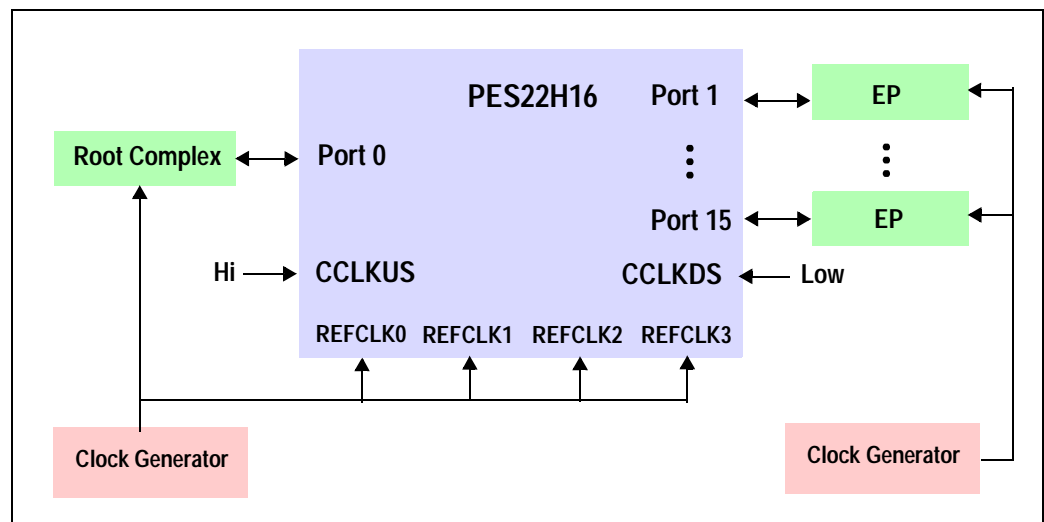


Figure 3.3 Common Clock on Upstream; Non-Common Clock on Downstream (must disable Spread Spectrum Clock)

# Notes

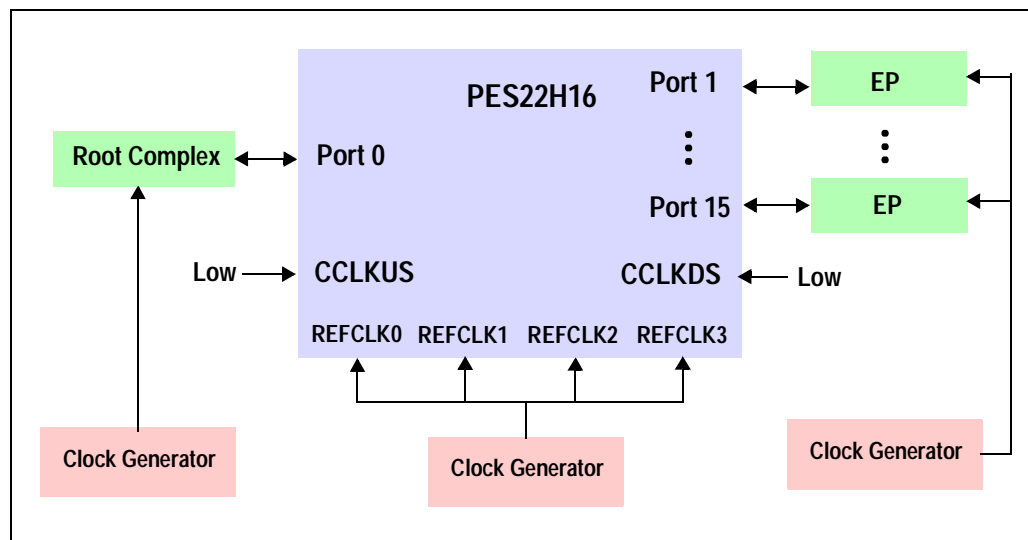


Figure 3.4 Non-Common Clock on Upstream and Downstream (must disable Spread Spectrum Clock)

## Initialization

A boot configuration vector consisting of the signals listed in Table 3.2 is sampled by the PES22H16 during a fundamental reset when PERSTN is negated. The boot configuration vector defines essential parameters for switch operation. Since the boot configuration vector is sampled only during a fundamental reset sequence, the value of signals which make up the boot configuration vector is ignored during other times and their state outside of a fundamental reset has no effect on the operation of the PES22H16.

While basic switch operation may be configured using signals in the boot configuration vector, advanced switch features require configuration via an external serial EEPROM. The external serial EEPROM allows modification of any bit in any software visible register. See Chapter 6, SMBus Interfaces, for more information on the serial EEPROM.

The external serial EEPROM and slave SMBus interface may be used to override the function of some of the signals in the boot configuration vector during a fundamental reset. The signals that may be overridden are noted in Table 3.2. The state of all of the boot configuration signals in Table 3.2 sampled during the most recent cold reset may be determined by reading the SWSTS register.

Signal	May Be Overridden	Description
CCLKDS	Y	<b>Common Clock Downstream.</b> When the CCLKDS pin is asserted, it indicates that a common clock is being used between the downstream device and the downstream port.
CCLKUS	Y	<b>Common Clock Upstream.</b> When the CCLKUS pin is asserted, it indicates that a common clock is being used between the upstream device and the upstream port.
MSMBSMODE	N	<b>Master SMBus Slow Mode.</b> The assertion of this pin indicates that the master SMBus should operate at 100 KHz instead of 400 KHz. This value may not be overridden.

Table 3.2 Boot Configuration Vector Signals

# Notes

Signal	May Be Overridden	Description
PERSTN	I	<b>Fundamental Reset.</b> Assertion of this signal resets all logic inside the PES22H16 and initiates a PCI Express fundamental reset.
RSTHALT	Y	<b>Reset Halt.</b> When this signal is asserted during a PCI Express fundamental reset, the PES22H16 executes the reset procedure and remains in a 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 reset state when the RSTHALT bit is cleared in the PA_SWCTL register by an SMBus master.
SWMODE[3:0]	N	<b>Switch Mode.</b> These configuration pins determine the PES22H16 switch operating mode. These pins should be static and not change following the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 through 0x7 - Reserved 0x8 - Normal switch mode with upstream port failover (port 0 selected as the upstream port) 0x9 - Normal switch mode with upstream port failover (port 2 selected as the upstream port) 0xA - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 0 selected as the upstream port) 0xB - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 2 selected as the upstream port) 0xC through 0xF - Reserved

Table 3.2 Boot Configuration Vector Signals

## Reset

The PES22H16 defines four reset categories: fundamental reset, hot reset, upstream secondary bus reset, and downstream secondary bus reset.

- A fundamental reset causes all logic in the PES22H16 to be returned to an initial state.
- A hot reset causes all logic in the PES22H16 to be returned to an initial state, but does not cause the state of register fields denoted as “sticky” to be modified.
- An upstream secondary bus reset causes all devices on the virtual PCI bus to be hot reset except the upstream port (i.e., upstream PCI to PCI bridge).
- A downstream secondary bus reset causes a hot reset to be propagated on the corresponding external secondary bus link.

There are two sub-categories of fundamental reset: cold reset and warm reset. A cold reset occurs following a device being powered on and assertion of PERSTN. A warm reset is a fundamental reset that occurs without removal of power.

### Fundamental Reset

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

- A cold reset initiated by a power-on and the assertion of the PCI Express Reset (PERSTN) input pin.
- A warm reset initiated by the assertion of the PCI Express Reset (PERSTN) input pin while power is on.
- A warm reset initiated by the writing of a one to the Fundamental Reset (FRST) bit in the Switch Control (SWCTL) register.

The following reset sequence is executed.

1. Wait for the fundamental reset condition to clear (e.g., negation of PERSTN).

# Notes

2. On negation of PERSTN, sample the boot configuration signals listed in Table 3.2. If PERSTN was not asserted, use the previously sampled boot configuration signal values (e.g., when a fundamental reset is the result of a one being written to the FRST bit in the SWCTL register).
  - Examine the state of the sampled SWMODE[3:0] signals to determine the switch operating mode.
3. The PLL and SerDes are initialized.
4. Link training begins. While link training is in progress, proceed to step 5.
5. If the Reset Halt (RSTHALT) pin is asserted, the RSTHALT bit in the SWSTS register is set.
6. If the switch operating mode is not a test mode, then the reset signal to the PCI Express stacks and associated logic is negated but they are held in a quasi-reset state in which the following actions occur.
  - All links enter an active link training state within 20ms of the clearing of the fundamental reset condition.
  - Within 100ms of the clearing of the fundamental reset condition, all of the stacks are able to process configuration transactions and respond to these transactions with a configuration request retry status completion. All other transactions are ignored.
7. The master SMBus operating frequency is determined.
  - The state of the MSMBSMODE signal is examined. If it is asserted, then the master SMBus is initialized to operate at 100 KHz rather than 400 KHz.
8. The slave SMBus is taken out of reset and initialized. The slave SMBus address specified by the SSMBADDR[5,3:1] pins is used.
9. The master SMBus is taken out of reset and initialized.
10. If the selected switch operating mode is one that requires initialization from the serial EEPROM, then the contents of the serial EEPROM are read and the appropriate PES22H16 registers are updated.
  - 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.
  - When serial EEPROM initialization completes or when an error is detected, the EEPROMDONE bit in the SMBUSSTS register is set.
11. If the Reset Halt (RSTHALT) bit is set in the SWCTL register, all of the logic is held in a reset state except the master and slave SMBuses, the control/status registers, and the stacks which continue to be held in a quasi-reset state and respond to configuration transactions with a retry. The device remains in this state until the RSTHALT bit is cleared via the slave SMBus. In this mode, an external agent may read and write any internal control and status registers and may access the external serial EEPROM via the EEPROMINTF register.
12. Normal device operation begins.

The PCIe base specification indicates that normal operation should begin within 1.0 second after a fundamental reset of a device. The reset sequence above guarantees that normal operation will begin within this 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 even with a Master SMBus operating frequency of 100 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.

A warm reset initiated by the writing of a one to the Fundamental Reset (FRST) bit in the Switch Control (SWCTL) register always results in the PES22H16 returning a completion to the requester before the warm reset process begins.

# Notes

The PES22H16 provides a reset output signal for each downstream port implemented as a GPIO alternate function. When a fundamental reset occurs, all of the GPIO pins default to GPIO inputs. Therefore, the downstream port resets are tri-stated. A system designer should use a pull-down on these signals if they are used as reset outputs.

The operation of a fundamental reset with serial EEPROM initialization (i.e., SWMODE[3:0] = 0x1) is illustrated in Figure 3.5.

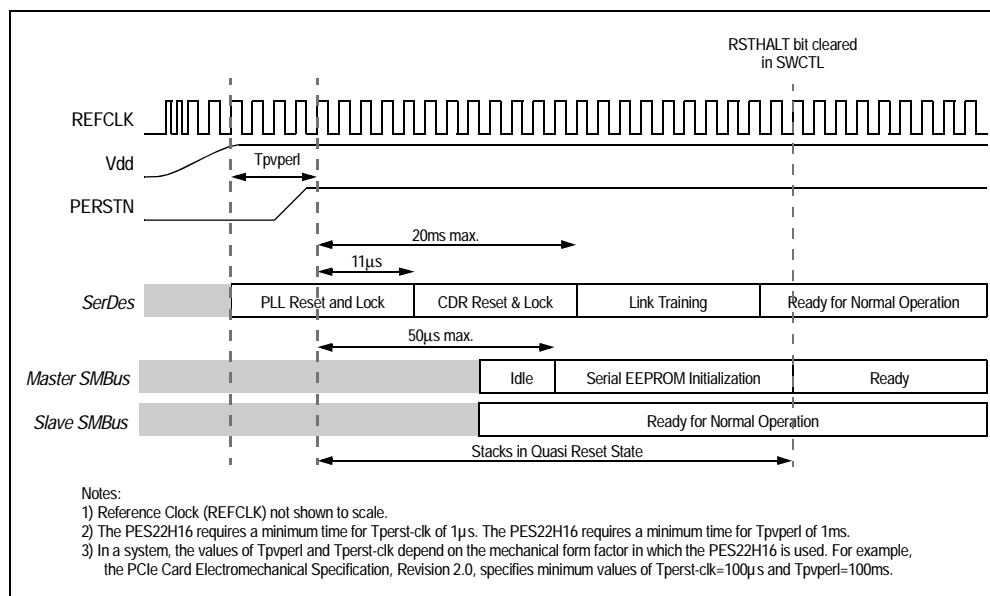


Figure 3.5 Fundamental Reset in Transparent Mode with Serial EEPROM Initialization

## Hot Reset

A hot reset may be initiated by any of the following conditions:

- Reception of TS1 ordered-sets on the upstream port indicating a hot reset.
- Data link layer of the upstream port transitions to the DL\_Down state.
- Writing a one to the Hot Reset (HRST) bit in the System Control (SWCTL) register.

# Notes

The initiation of a hot reset due to the data link layer of the upstream port transitioning to the DL\_Down state may be disabled by setting the Disable Link Down Hot Reset (DLDHRST) bit in the Switch Control (SWCTL) register. Other hot reset conditions are unaffected by this bit. When a hot reset occurs, the following sequence is executed.

1. Each downstream port whose link is up propagates the hot reset by transmitting TS1 ordered sets with the hot reset bit set.
2. All of the logic associated with the PES22H16 except the PLLs, SerDes, master SMBus interface, and slave SMBus interface is reset.
3. All registers fields in all registers, except those denoted as "sticky" or Read and Write when Unlocked (i.e., RWL), are reset to their initial value. The value of fields denoted as "sticky" or RWL is preserved across a hot reset.
4. Link training begins. While link training is in progress, proceed to step 5.
5. The PCI Express stacks and associated logic are held in a quasi-reset state in which the following actions occur.
  - All links enter an active link training state within 20ms of the clearing of the hot reset condition.
  - Within 100ms of the clearing of the fundamental reset condition, all of the stacks are able to process configuration transactions and respond to these transactions with a configuration request retry status completion. All other transactions are ignored.
6. If the selected switch operating mode is one that requires initialization from the serial EEPROM and the Disable Hot Reset Serial EEPROM Initialization (DHRSTSEI) bit is not set in the Switch Control (SWCTL) register, then the contents of the serial EEPROM are read and the appropriate PES22H16 registers are updated. In addition, if the Reset Halt (RSTHALT) pin is asserted, the RSTHALT bit in the SWSTS register is set.
  - 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.
  - When serial EEPROM initialization completes or when an error is detected, the DONE bit in the SMBUSSTS register is set.
7. If the Reset Halt (RSTHALT) bit is set in the SWCTL register, all of the logic is held in a reset state except the master and slave SMBuses. The RSTHALT bit is set only under the following two conditions:
  - serial EEPROM initialization is enabled in step 6 and an error is detected during loading of the serial EEPROM or
  - the user intentionally sets this bit through the EEPROM code.
8. Normal device operation begins.

The operation of the slave SMBus interface is unaffected by a hot reset. Using the slave SMBus to access a register that is reset by a hot reset causes zero to be returned on a read and written data to be ignored on writes.

A hot reset initiated by the writing of a one to the Hot Reset (HRST) bit in the Switch Control (SWCTL) register always results in the PES22H16 returning a completion to the requester before the hot reset process begins.

## Upstream Secondary Bus Reset

An upstream secondary bus reset may be initiated by the following condition:

- A one is written to the Secondary Bus Reset (SRESET) bit in the upstream port's (i.e., port 0) Bridge Control Register (BRCTL).

## Notes

When an upstream secondary bus reset occurs, the following sequence is executed.

1. Each downstream port whose link is up propagates the reset by transmitting TS1 ordered sets with the hot reset bit set.
2. All registers fields in all registers associated with downstream ports, except those denoted as “sticky” or Read and Write when Unlocked (i.e., RWL), are reset to their initial value. The value of fields denoted as “sticky” or RWL is unaffected by an upstream secondary bus reset.
3. All TLPs received from downstream ports and queued in the PES22H16 are discarded.
4. Logic in the stack, application layer and switch core associated with the downstream ports are gracefully reset.
5. Wait for the Secondary Bus Reset (SRESET) bit in the upstream port’s Switch Control Register (SWCTRL) to clear.
6. Normal downstream port operation begins.

The operation of the upstream 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 complete normally. During an upstream secondary bus reset, all TLPs destined to the secondary side of the upstream port’s PCI-to-PCI bridge are treated as unsupported requests. The operation of the slave SMBus interface is unaffected by an upstream secondary bus reset. Using the slave SMBus to access a register that is reset by an upstream secondary bus reset causes zero to be returned on a read and written data to be ignored on writes.

### Downstream Secondary Bus Reset

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

- A one is written to the Secondary Bus Reset (SRESET) bit in a downstream port’s (i.e., port 0) Bridge Control Register (BCTRL).

When a downstream secondary bus reset occurs, the following sequence is executed.

1. If the corresponding downstream port’s link is up, TS1 ordered sets with the hot reset bit set are transmitted
2. All TLPs received from corresponding downstream port and queued in the PES22H16 are discarded.
3. Wait for the Secondary Bus Reset (SRESET) bit in the upstream port’s Switch Control Register (SWCTRL) to clear.
4. Normal downstream port operation begins.

The operation of the upstream port is unaffected by a downstream secondary bus reset. The operation of other downstream ports is unaffected by a downstream secondary bus reset. During a downstream secondary bus reset, Type 0 configuration read and write transactions that target the downstream port complete normally. During a downstream secondary bus reset, all TLPs destined to the secondary side of the downstream port’s PCI-to-PCI bridge are treated as unsupported requests. The operation of the slave SMBus interface is unaffected by a downstream secondary bus reset.

### Downstream Port Reset Outputs

Individual downstream port reset outputs (P1RSTN through P15RSTN) are provided as GPIO pin alternate functions. Following a fundamental reset, all of the GPIO pins default to GPIO inputs. Therefore, the downstream port resets are tri-stated. A system designer should use a pull-down on these signals if they are used as reset outputs. The PES22H16 ensures through hardware that the minimum PxRSTN assertion pulse width is no less than 200  $\mu$ S.

Downstream port reset outputs can be configured to operate in one of three modes. These modes are: power enable controlled reset output, power good controlled reset output, and hot reset controlled output. The downstream port reset output mode is determined by the Reset Mode (RSTMODE) field in the Hot-Plug Configuration Control (HPCFGCTL) register.

# Notes

## Power Enable Controlled Reset Output

In this mode a downstream 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 3.6. A downstream port's slot power is controlled by the Power Controller Control (PCC) bit in the PCI Express Slot Control (PCIESCTL) register

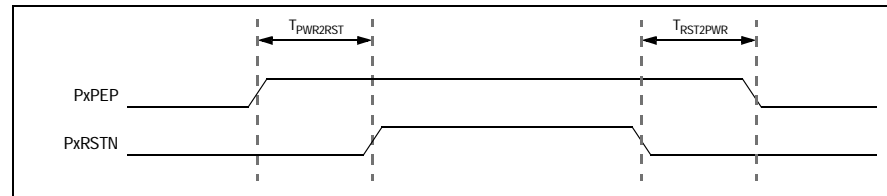


Figure 3.6 Power Enable Controlled Reset Output Mode Operation

While slot power is disabled, the corresponding downstream 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 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 port reset output is negated.

When slot power is disabled by writing a one to the PCC bit, the corresponding downstream 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 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 3.7.

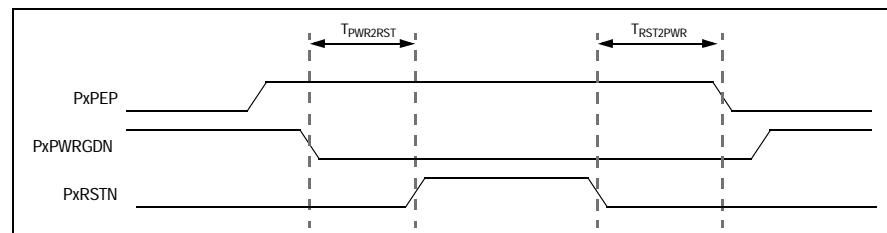


Figure 3.7 Power Good Controlled Reset Output Mode Operation

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 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 port is not being reset (i.e., PxRSTN is negated) a power fault is detected (i.e., PxPWRGDN is negated), then the corresponding port reset output is immediately asserted.

## Notes

Since the PxPWRGDN signal is an I/O expander input, it may not be 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 PES22H16.



# Link Operation

## Notes

### Introduction

The PES22H16 contains two x4 ports and fourteen x1 ports. The default link width is x4 for ports 0 and 2 and x1 for the remaining ports. SerDes lanes are statically assigned to a port.

### Polarity Inversion

Each port of the PES22H16 supports automatic polarity inversion as required by the PCIe 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 six through 16 of the TS1 and TS2 ordered sets for inversion of the PEXAP[n] and PEXAN[n] signals. If an inversion is detected, then logic for the receiving lane automatically inverts received data. Polarity inversion is a lane and not a link function. Therefore, it is possible for some lanes of link to be inverted and for others to not be inverted.

### Link Width Negotiation

The PES22H16 supports the optional link variable width negotiation feature outlined in the PCIe specification. During link training, each x4 port is capable of negotiating to a x4, x2 or x1 link width.

The negotiated width of each link may be determined from the Link Width (LW) field in the corresponding port's PCI Express Link Status (PCIELSTS) register. The Maximum Link Width (MAXLNKWDTH) field in a port's PCI Express Link Capabilities (PCIELCAP) register contains the maximum link width of the port. This field is of RWL type and may be modified 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 link training occurs.

The initial value of the MAXLNKWDTH field defaults to x4 or x1mode. 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.

When a x4 port negotiates to a width less than x4, the unused SerDes lanes are put in a low power state (i.e., L1 state). When a port is disabled, all SerDes lanes associated with that port are powered down.

### Link Retraining

Link retraining should not cause either a downstream component or an upstream component to reset or revert to default values. Writing a one to the Link Retrain (LRET) bit in the upstream port's PCI Express Link Control (PCIELCTL) register when the REGUNLOCK bit is set in the SYSCTL register forces the upstream PCIe link to retrain. **When this occurs the LTSSM transitions directly to the Recovery state.** Writing a one to the Link Retrain (LRET) bit in a downstream port's PCI Express Link Control (PCIELCTL) register regardless of the REGUNLOCK bit state in the SYSCTL register forces the downstream PCIe link to retrain. **When this occurs the LTSSM transitions directly to the Recovery state.**

### Link Down

When a link goes down, all TLPs received by that port and queued in the switch are discarded and all TLPs received by other ports and destined to the port whose link is down are treated as Unsupported Requests (UR). While a downstream link is down, it is possible to perform configuration read and write operations to the PCI-PCI bridge associated with that link. When a link comes up, flow control credits for the configured size of the input buffer queues are advertised.

## Notes

### 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 the upstream switch port, then the fields in the message are written to the PCI Express Device Capabilities (PCIEDCAP) register of that port.

- 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.

#### Downstream Port

A Set\_Slot\_Power\_Limit message is sent 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 port is up.
- A link associated with the downstream port transitions from a non-operational state to an operational (i.e., up) state.

### Link States

The PES22H16 supports 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
  - May be automatically entered or directed by software by placing the device in the D3<sub>hot</sub> state
- L2/L3 Ready
  - The L2/L3 state is entered after the acknowledgement of a PM\_Turn\_Off Message.
  - There is no TLP or DLLP communications over a link in this state.
- L3
  - Link is completely unpowered and off

## Notes

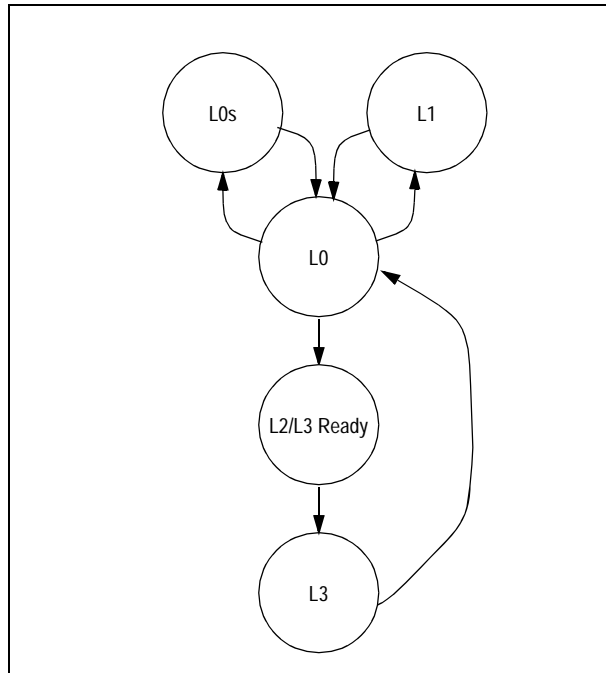


Figure 4.1 PES22H16 ASPM Link State Transitions

## Active State Power Management

The operation of Active State Power Management (ASPM) is orthogonal to power management. Once enabled by the ASPM field in the PCI Express Link Control (PCIELCTL) register, ASPM link state transitions are initiated by hardware without software involvement. The PES22H16 ASPM supports the required L0s state as well as the optional L1 state.

The L0s Entry Timer (L0ET) field in the PCI Power Management Proprietary Control (PMPC) register controls the amount of time L0s entry conditions must be met before the hardware transitions the link to the L0s state. The L1 Entry Timer (L1ET) field in the PCI Power Management Proprietary Control (PMPC) register controls the amount of time L1 entry conditions must be met before the hardware 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. Note that L1 entry requests are only made by the PES22H16 upstream port. If the link partner acknowledges the transition, then the L1 state is entered. Otherwise the L0s state is entered.

## Link Status

Associated with each port is a Port Link Up (PxLINKUPN) status output and a Port Activity (PxACTIVEN) status output. These outputs are provided on I/O expander tbd. See section I/O Expanders on page 6-6 for the operation of the I/O expander and the mapping of these status outputs to I/O expander pins. 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 PCI Express 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.

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.

Notes



# General Purpose I/O

## Notes

## Introduction

The PES22H16 has 32 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), and General Purpose I/O Data (GPIOD) registers in the upstream port's PCI configuration space. As shown in Table 5.1, many GPIO pins are shared with other on-chip functions. The GPIO Function (GPIOFUNC) register whether a GPIO bit operates as a general purpose I/O or as the specified alternate function.

GPIO Pin	Alternate Function Pin Name	Alternate Function Description	Alternate Function Pin Type
5	GPEN	General purpose event output	Output
6	P1RSTN	Reset output for downstream port 1	Output
7	P2RSTN	Reset output for downstream port 2	Output
8	P3RSTN	Reset output for downstream port 3	Output
9	P4RSTN	Reset output for downstream port 4	Output
10	P5RSTN	Reset output for downstream port 5	Output
11	P6RSTN	Reset output for downstream port 6	Output
12	P7RSTN	Reset output for downstream port 7	Output
13	P8RSTN	Reset output for downstream port 8	Output
14	P9RSTN	Reset output for downstream port 9	Output
15	P10RSTN	Reset output for downstream port 10	Output
16	P11RSTN	Reset output for downstream port 11	Output
17	P12RSTN	Reset output for downstream port 12	Output
18	P13RSTN	Reset output for downstream port 13	Output
19	P14RSTN	Reset output for downstream port 14	Output
20	P15RSTN	Reset output for downstream port 15	Output
21	IOEXPINTN0	SMBus I/O expander interrupt 0	Input
22	IOEXPINTN1	SMBus I/O expander interrupt 1	Input
23	IOEXPINTN2	SMBus I/O expander interrupt 2	Input
24	IOEXPINTN3	SMBus I/O expander interrupt 3	Input
25	IOEXPINTN4	SMBus I/O expander interrupt 4	Input
26	IOEXPINTN5	SMBus I/O expander interrupt 5	Input

Table 5.1 General Purpose I/O Pin Alternate Function (Part 1 of 2)

## Notes

GPIO Pin	Alternate Function Pin Name	Alternate Function Description	Alternate Function Pin Type
27	IOEXPINTN6	SMBus I/O expander interrupt 6	Input
28	IOEXPINTN7	SMBus I/O expander interrupt 7	Input
31	IOEXPINTN10	SMBus I/O expander interrupt 10	Input

**Table 5.1 General Purpose I/O Pin Alternate Function (Part 2 of 2)**

After reset, all GPIO pins default to the GPIO input function. GPIO pins configured as GPIO inputs are sampled no more frequently than once every 128ns and may be treated as asynchronous inputs. When a GPIO pin is configured to use the GPIO function, the unneeded alternate function associated with the pin is held in an inactive state by internal logic. Care should be exercised when configuring the GPIO pins as outputs since an incorrect configuration could cause damage to external components as well as the PES22H16.

## GPIO Configuration

Associated with each GPIO pin is a bit in the GPIOFUNC, GPIOCFG and GPIOD registers. Table 5.2 summarizes the configuration of GPIO pins.

GPIOFUNC	GPIOCFG	Pin Function
0	0	GPIO input
0	1	GPIO output
1	don't care	Alternate function

**Table 5.2 GPIO Pin Configuration**

### GPIO Pin Configured as an 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. Note that 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.

### GPIO Pin Configured as an 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 can be determined by reading the GPIOD register.

### GPIO Pin Configured as an Alternate Function

When configured as an alternate function in the GPIOFUNC register, the pin behaves as described by the section associated with that function. The value of the alternate function pin can be determined at any time by reading the GPIOD register.



# SMBus Interfaces

## Notes

## Introduction

The PES22H16 contains two SMBus interfaces. The slave SMBus interface provides full access to all software visible registers in the PES22H16, allowing every register in the device to be read or written by an external SMBus master. The slave SMBus may also be used to initialize 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. Six pins make up each of the two SMBus interfaces. These pins consist of an SMBus clock pin, an SMBus data pin, and 4 SMBus address pins. As shown in Figure 6.1, the master and slave SMBuses may be used in a unified or split configuration.

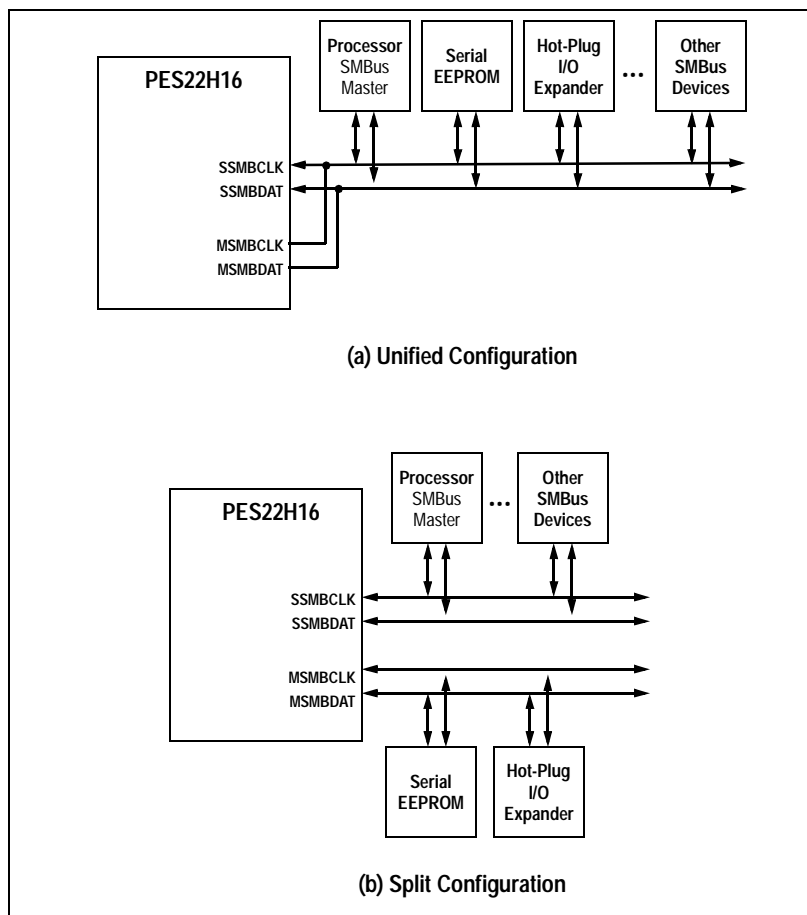


Figure 6.1 SMBus Interface Configuration Examples

In the unified configuration, shown in Figure 6.1(a), the master and slave SMBuses are tied together and the PES22H16 acts both as an SMBus master as well as an SMBus slave on this bus. This requires that the external SMBus master or processor that has access to the PES22H16 registers support SMBus arbitration. In some systems, this external SMBus master interface may be implemented using general purpose I/O pins on a processor or microcontroller, and thus may not support SMBus arbitration. To support these systems, the PES22H16 may be configured to operate in a split configuration as shown in Figure 6.1(b).

## Notes

In the split configuration, the master and slave SMBuses operate as two independent buses and thus multi-master arbitration is not required.

## Master SMBus Interface

The master SMBus interface is used during a fundamental reset to load configuration values from an optional serial EEPROM. It is also used to support optional I/O expanders used for hot-plug and other status signals.

### Initialization

Master SMBus initialization occurs during a fundamental reset (see section Fundamental Reset on page 3-4).

During a fundamental reset initialization sequence, the state of the Master SMBus Slow Mode (MSMB-SMODE) signal is examined. If this signal is asserted, then the Master SMBus Clock Prescaler (MSMB-BCP) field in the SMBus Control (SMBUSCTL) register is initialized to support 100 KHz SMBus operation. If the signal is negated, then the MSMB-BCP field is initialized for 400 KHz SMBus operation.

### Serial EEPROM

During a fundamental or hot 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[3:0]) field selects an operating mode that performs serial EEPROM initialization. The address used by the SMBus interface to access the serial EEPROM is specified by the MSMBADDR[4:1] signals as shown in Table 6.1.

Address Bit	Address Bit Value
1	MSMBADDR[1]
2	MSMBADDR[2]
3	MSMBADDR[3]
4	MSMBADDR[4]
5	1
6	0
7	1

Table 6.1 Serial EEPROM SMBus Address

### Device Initialization from a 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 PES22H16. Any PES22H16 software visible register in any port may be initialized with values stored in the serial EEPROM.

Each software visible register in the PES22H16 has a CSR system address which is formed by adding the PCI configuration space offset value of the register to the base address of the configuration space in which the register is located. Configuration blocks stored in the serial EEPROM use this CSR system address shifted right two bits (i.e., configuration blocks in the serial EEPROM use doubleword CSR system addresses and not byte CSR system addresses).

Base addresses for the PCI configuration spaces in the PES22H16 are listed in Table 9.1. 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.

# Notes

Any serial EEPROM compatible with those listed in Table 6.2 may be used to store the PES22H16 initialization values. Some of these devices are larger than the total size of all of the PCI configuration spaces in the PES22H16 that may be initialized and thus may not be fully utilized.

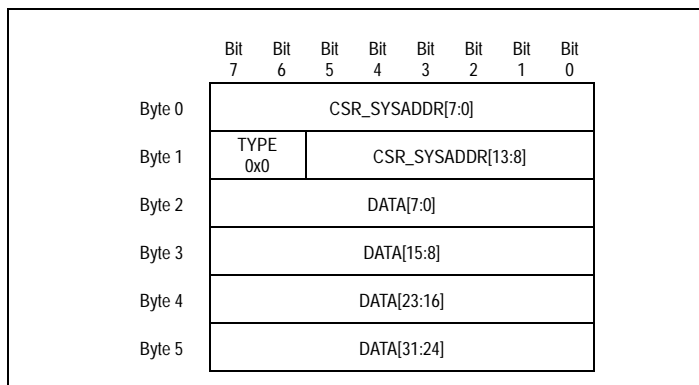
Serial EEPROM	Size
24C32	4 KB
24C64	8 KB
24C128	16 KB
24C256	32 KB
24C512	64 KB

**Table 6.2 PES22H16 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.

All register initialization performed by the serial EEPROM is performed in double word quantities. There are three configuration block types that may be stored in the serial EEPROM. The first type is a single double word initialization sequence. A double word initialization sequence occupies six bytes in the serial EEPROM and is used to initialize a single double word quantity in the PES22H16.

A single double word initialization sequence consists of three fields and its format is shown in Figure 6.1. The CSR\_SYSADDR field contains the double word CSR system address of the double word to be initialized. The actual CSR system address, which is a byte address, equals this value with two lower zero bits appended. The next field is the TYPE field that indicates the type of the configuration block. For single double word initialization sequence, this value is always 0x0. The final DATA field contains the double word initialization value.



**Figure 6.1 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 6.2. The CSR\_SYSADDR field contains the starting double word CSR system address to be initialized. The next field is the TYPE field that indicates the type of the configuration block. For sequential double word initialization sequences, this value is always 0x1. 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

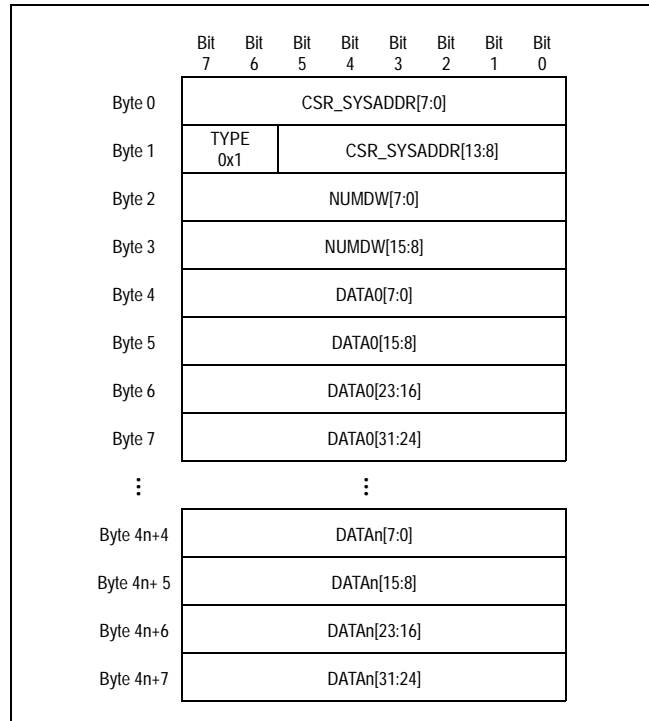


Figure 6.2 Sequential Double Word Initialization Sequence Format

The final type of configuration block is the configuration done sequence which is used to signify the end of a serial EEPROM initialization sequence.

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 9), then the Unmapped Register Initialization Attempt (URIA) bit is set in the SMBUSSTS register and the write is ignored.

The configuration done sequence consists of two fields and its format is shown in Figure 6.3. 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. The second field is the TYPE field which is always 0x3 for configuration done sequences.

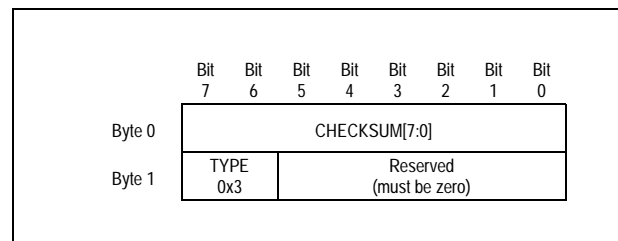


Figure 6.3 Configuration Done Sequence Format

The checksum in the configuration done sequence enables the integrity of the serial EEPROM initialization to be verified. Since uninitialized EEPROMs typically have a value of all ones, initialization from an uninitialized serial EEPROM will result in a checksum mismatch.

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.<sup>1</sup> The 1's complement of this

<sup>1</sup> This includes the byte containing the TYPE field.

# Notes

sum is placed in the checksum field. 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.<sup>1</sup> 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. 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. This allows debugging of the error condition via the slave SMBus interface but prevents normal system operation with a potentially incorrectly initialized device. Error information is recorded in the SMBUSSTS register.

Once serial EEPROM initialization completes, or when an error is detected, the EEPROM Done (EEPROMDONE) bit is set in the SMBus Status (SMBSTS) register. A summary of possible errors during serial EEPROM initialization and specific action taken when detected is summarized in Figure 6.4.

Error	Action Taken
Configuration Done Sequence checksum mismatch with that computed by the PES22H16	<ul style="list-style-type: none"> <li>- Set RSTHALT bit in SWCTL register</li> <li>- ICSERR bit is set in the SMBUSSTS register</li> <li>- Abort initialization, set DONE bit in the SMBUSSTS register</li> </ul>
Invalid configuration block type (only invalid type is 0x2)	<ul style="list-style-type: none"> <li>- Set RSTHALT bit in SWCTL register</li> <li>- ICSERR bit is set in the SMBUSSTS register</li> <li>- Abort initialization, set DONE bit in the SMBUSSTS register</li> </ul>
An unexpected NACK is observed during a master SMBus transaction	<ul style="list-style-type: none"> <li>- Set RSTHALT bit in SWCTL register</li> <li>- NAERR bit is set in the SMBUSSTS register</li> <li>- Abort initialization, set DONE bit in the SMBUSSTS register</li> </ul>
Master SMBus interface loses 16 consecutive arbitration attempts	<ul style="list-style-type: none"> <li>- Set RSTHALT bit in SWCTL register</li> <li>- LAERR bit is set in the SMBUSSTS register</li> <li>- Abort initialization, set DONE bit in the SMBUSSTS register</li> </ul>
A misplaced START or STOP condition is detected by the master SMBus interface	<ul style="list-style-type: none"> <li>- Set RSTHALT bit in SWCTL register</li> <li>- OTHERERR bit is set in the SMBUSSTS register</li> <li>- Abort initialization, set DONE bit in the SMBUSSTS register</li> </ul>

Figure 6.4 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 PCIe root. Programming the serial EEPROM via the slave SMBus is described in section Serial EEPROM Read or Write Operation on page 6-20. A PCIe 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.

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.

<sup>1</sup> This includes the checksum byte as well as the byte that contains the type and reserved field.

# Notes

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 PES22H16 utilizes external SMBus/I2C-bus I/O expanders connected to the master SMBus interface for hot-plug and port status signals.

The PES22H16 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. An external SMBus I/O expander provides 16 bit I/O pins that may be configured as inputs or outputs.

PES22H16 supports up to 11 external I/O expanders. Figure 6.5 summarizes the allocation of functions to I/O expanders. I/O expanders zero through seven are used to provide general hot-plug I/O signals. I/O expander eight provides link status outputs while I/O expander nine provides link activity LED status outputs. I/O expander ten provides hot-plug power good status inputs.

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	Function
0	Lower	Port 1 hot-plug
	Upper	Port 3 hot-plug
1	Lower	Port 2 hot-plug
	Upper	Port 4 hot-plug
2	Lower	Port 5 hot-plug
	Upper	Port 7 hot-plug
3	Lower	Port 6 hot-plug
	Upper	Port 8 hot-plug
4	Lower	Port 9 hot-plug
	Upper	Port 11 hot-plug
5	Lower	Port 10 hot-plug
	Upper	Port 12 hot-plug
6	Lower	Port 13 hot-plug
	Upper	Port 15 hot-plug
7	Lower	Port 14 hot-plug
	Upper	reserved
8	Lower	Link status (ports 0 through 7)
	Upper	Link status (ports 8 through 15)

Figure 6.5 I/O Expander Function Allocation (Part 1 of 2)

# Notes

SMBus I/O Expander	Section	Function
9	Lower	Link activity (ports 0 through 7)
	Upper	Link activity (ports 8 through 15)
10	Lower	Power good inputs (ports 0 through 7)
	Upper	Power good inputs (ports 8 through 15)

**Figure 6.5 I/O Expander Function Allocation (Part 2 of 2)**

During the PES22H16 initialization, the SMBus/I2C-bus address allocated each I/O expander used in that system configuration should be written to the corresponding IO Expander Address (IOE[0:10]ADDR) field. The IOE[0:3]ADDR fields are contained in the I/O Expander Address 0 (IOEXPADDR0) register, the IOE[4:7]ADDR fields are contained in the SMBus I/O Expander Address 1 (IOEXPADDR1) register, and the IOE[8:10]ADDR fields are contained in the SMBus I/O Expander Address 2 (IOEXPADDR2) register.

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 -> 1 -> 0 transition or a 1 -> 0 -> 1 transition).

Whenever the value of an IOEXPADDR field is written, SMBus write transactions are issued to the corresponding I/O expander by the PES22H16 to configure the device. This configuration initializes the direction of each I/O expander signal and sets outputs to their default value.

Outputs for ports that are disabled or are not implemented in that configuration or bond option, are set to their negated value (e.g., the power indicator is turned off, the link is down, there is no activity, etc.).

The default value of I/O expander outputs is shown in Figure 6.6. Note that this default value may be modified via serial EEPROM or SMBus configuration prior to SMBus initialization by changing the state of the PCI Express Slot Control Register (PCIESCTL) or Hot-Plug Configuration Control (HPCFGCTL).

SMBus I/O Expander Bit	Signal	Description	Default Value
(I/O-x.4)	P2AIN	Attention indicator output (off)	1
(I/O-x.5)	P2PIN	Power indicator output (on)	0
(I/O-x.6)	P2PEP	Power enable output (on)	1
(I/O-x.7)	P2ILOCKP	Electromechanical interlock (negated - off)	0

**Figure 6.6 I/O Expander Default Output Signal Value**

# Notes

The following I/O expander configuration sequence is issued by the PES22H16 to I/O expanders zero through seven (i.e., the ones that contain hot-plug signals).

- 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.
- 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.
- write value 0x0 to I/O expander register 4 (no inversion in IO-0)
- write value 0x0 to I/O expander register 5 (no inversion in IO-1)
- 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.
- 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.
- 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)
- 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 PES22H16 to I/O expanders eight and nine (i.e., the one that contains link up and link activity status).

- Write link up or activity status for ports 0 through 7 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.
- Write link up or activity status for ports 8 through 15 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.
- Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
- Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
- 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.
- 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 PES22H16 to I/O expander ten (i.e., the one that contains power good inputs).

- Write value 0x0 to I/O expander register 4 (no inversion in IO-0)
- Write value 0x0 to I/O expander register 5 (no inversion in IO-1)
- 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.
- 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.
- 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)
- 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)

While the I/O expander is enabled, the PES22H16 maintains the I/O bus expander signals and the PES22H16 internal view of the hot-plug signals in a consistent state. This means that whenever that I/O bus expander state and the PES22H16 internal view of the signal state differs, an SMBus transaction is initiated by the PES22H16 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 PES22H16 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 PES22H16 will initiate an SMBus transaction to modify the state of the I/O expander hot-plug outputs.

## Notes

Each I/O expander has an open drain interrupt output that is asserted when a pin configured as an input changes state from the value previously read. Each interrupt output from an I/O expander should be connected to the corresponding PES22H16 I/O expander interrupt input. Since the PES22H16 I/O expander interrupt inputs are GPIO alternate functions, the corresponding GPIOs should be initialized during configuration to operate in alternate function mode. See Chapter 5, General Purpose I/O.

Whenever the PES22H16 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 PES22H16 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 PES22H16 to issue a master SMBus transaction to read the updated state of the I/O expander inputs. Regardless of the state of the interrupt output of the I/O expander, the PES22H16 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 PES22H16 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 errors detected during I/O expander SMBus read or write transactions is reflected in the status bits of the SMBus Status (SMBUSSTS) register.

The I/O Expander Interface (IOEXPINTF) register allows direct testing and debugging of the I/O expander functionality. The Select (SEL) field in the IOEXPINTF register selects the I/O expander number on which other fields in the register operate. The I/O Expander Data field in the IOEXPINTF register reflect the current state, as viewed by the PES22H16, of the I/O expander inputs and outputs selected by the SEL field.

Writing a one to the Reload I/O Expander Signals (RELOADIOEX) bit in the IOEXPINTF register causes the PES22H16 to generate SMBus write and read transactions to the I/O expander number selected in the SEL field. This results in the value of the IOEDATA field being updated to reflect the current state of the corresponding I/O expander signals. This feature may be used to aid in debugging I/O expander operation. For example, a user who neglects to configure a GPIO as an alternate function may use this feature to determine that master SMBus transactions to the I/O expander function properly and that the issue is with the interrupt logic.

The IO Expander Test Mode (IOEXTM) bit in the IOEXPINTF register allows an I/O expander test mode to be entered. When this bit is set, the PES22H16 core logic outputs are ignored and the values written to the I/O expander for output bits are the values in the IOEDATA field. In this mode, the PES22H16 issues a transaction to update the state of the I/O expander whenever a bit corresponding to an I/O expander output changes state due to a write to the IOEDATA field and the RELOADIOEX bit is set. Bits in the IOEDATA field that correspond to outputs are dependent on the I/O expander number selected in the SEL field in the IOEXPINTF register. The outputs for each I/O expander number are shown in Table 6.3 through 6.13.

# Notes

## 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 PES22H16 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 upstream secondary bus resets and hot-resets.
- The standard PCA9555 I/O expander is not able to support the number of unique addresses required by all of the PES22H16 hot-plug I/O expanders. Therefore, it is recommended that a MAX7313 be used instead. The MAX7313 is software and pin compatible with the PCA9555
- 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, an I/O expander reset circuit may be used to reset the I/O expanders. Contact IDT for more information.

## I/O Expander 0

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P1APN	Port 1 attention push button input
1 (I/O-0.1)	I	P1PDN	Port 1 presence detect input
2 (I/O-0.2)	I	P1PFN	Port 1 power fault input
3 (I/O-0.3)	I	P1MRLN	Port 1 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P1AIN	Port 1 attention indicator output
5 (I/O-0.5)	O	P1PIN	Port 1 power indicator output
6 (I/O-0.6)	O	P1PEP	Port 1 power enable output
7 (I/O-0.7)	O	P1ILOCKP	Port 1 electromechanical interlock
8 (I/O-1.0)	I	P3APN	Port 3 attention push button input
9 (I/O-1.1)	I	P3PDN	Port 3 presence detect input
10 (I/O-1.2)	I	P3PFN	Port 3 power fault input
11 (I/O-1.3)	I	P3MRLN	Port 3 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P3AIN	Port 3 attention indicator output
13 (I/O-1.5)	O	P3PIN	Port 3 power indicator output
14 (I/O-1.6)	O	P3PEP	Port 3 power enable output
15 (I/O-1.7)	O	P3ILOCKP	Port 3 electromechanical interlock

**Table 6.3 I/O Expander 0 Signals**

<sup>1</sup> I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

# Notes

## I/O Expander 1

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P2APN	Port 2 attention push button input
1 (I/O-0.1)	I	P2PDN	Port 2 presence detect input
2 (I/O-0.2)	I	P2PFN	Port 2 power fault input
3 (I/O-0.3)	I	P2MRLN	Port 2 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P2AIN	Port 2 attention indicator output
5 (I/O-0.5)	O	P2PIN	Port 2 power indicator output
6 (I/O-0.6)	O	P2PEP	Port 2 power enable output
7 (I/O-0.7)	O	P2ILOCKP	Port 2 electromechanical interlock
8 (I/O-1.0)	I	P4APN	Port 4 attention push button input
9 (I/O-1.1)	I	P4PDN	Port 4 presence detect input
10 (I/O-1.2)	I	P4PFN	Port 4 power fault input
11 (I/O-1.3)	I	P4MRLN	Port 4 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P4AIN	Port 4 attention indicator output
13 (I/O-1.5)	O	P4PIN	Port 4 power indicator output
14 (I/O-1.6)	O	P4PEP	Port 4 power enable output
15 (I/O-1.7)	O	P4ILOCKP	Port 4 electromechanical interlock

**Table 6.4 I/O Expander 1 Signals**

<sup>1</sup>. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

## I/O Expander 2

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P5APN	Port 5 attention push button input
1 (I/O-0.1)	I	P5PDN	Port 5 presence detect input
2 (I/O-0.2)	I	P5PFN	Port 5 power fault input
3 (I/O-0.3)	I	P5MRLN	Port 5 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P5AIN	Port 5 attention indicator output
5 (I/O-0.5)	O	P5PIN	Port 5 power indicator output
6 (I/O-0.6)	O	P5PEP	Port 5 power enable output
7 (I/O-0.7)	O	P5ILOCKP	Port 5 electromechanical interlock

**Table 6.5 I/O Expander 2 Signals (Part 1 of 2)**

## Notes

SMBus I/O Expander Bit	Type	Signal	Description
8 (I/O-1.0)	I	P7APN	Port 7 attention push button input
9 (I/O-1.1)	I	P7PDN	Port 7 presence detect input
10 (I/O-1.2)	I	P7PFN	Port 7 power fault input
11 (I/O-1.3)	I	P7MRLN	Port 7 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P7AIN	Port 7 attention indicator output
13 (I/O-1.5)	O	P7PIN	Port 7 power indicator output
14 (I/O-1.6)	O	P7PEP	Port 7 power enable output
15 (I/O-1.7)	O	P7ILOCKP	Port 7 electromechanical interlock

**Table 6.5 I/O Expander 2 Signals (Part 2 of 2)**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

### I/O Expander 3

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P6APN	Port 6 attention push button input
1 (I/O-0.1)	I	P6PDN	Port 6 presence detect input
2 (I/O-0.2)	I	P6PFN	Port 6 power fault input
3 (I/O-0.3)	I	P6MRLN	Port 6 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P6AIN	Port 6 attention indicator output
5 (I/O-0.5)	O	P6PIN	Port 6 power indicator output
6 (I/O-0.6)	O	P6PEP	Port 6 power enable output
7 (I/O-0.7)	O	P6ILOCKP	Port 6 electromechanical interlock
8 (I/O-1.0)	I	P8APN	Port 8 attention push button input
9 (I/O-1.1)	I	P8PDN	Port 8 presence detect input
10 (I/O-1.2)	I	P8PFN	Port 8 power fault input
11 (I/O-1.3)	I	P8MRLN	Port 8 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P8AIN	Port 8 attention indicator output
13 (I/O-1.5)	O	P8PIN	Port 8 power indicator output
14 (I/O-1.6)	O	P8PEP	Port 8 power enable output
15 (I/O-1.7)	O	P8ILOCKP	Port 8 electromechanical interlock

**Table 6.6 I/O Expander 3 Signals**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

# Notes

## I/O Expander 4

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P9APN	Port 9 attention push button input
1 (I/O-0.1)	I	P9PDN	Port 9 presence detect input
2 (I/O-0.2)	I	P9PFN	Port 9 power fault input
3 (I/O-0.3)	I	P9MRLN	Port 9 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P9AIN	Port 9 attention indicator output
5 (I/O-0.5)	O	P9PIN	Port 9 power indicator output
6 (I/O-0.6)	O	P9PEP	Port 9 power enable output
7 (I/O-0.7)	O	P9ILOCKP	Port 9 electromechanical interlock
8 (I/O-1.0)	I	P11APN	Port 11 attention push button input
9 (I/O-1.1)	I	P11PDN	Port 11 presence detect input
10 (I/O-1.2)	I	P11PFN	Port 11 power fault input
11 (I/O-1.3)	I	P11MRLN	Port 11 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P11AIN	Port 11 attention indicator output
13 (I/O-1.5)	O	P11PIN	Port 11 power indicator output
14 (I/O-1.6)	O	P11PEP	Port 11 power enable output
15 (I/O-1.7)	O	P11ILOCKP	Port 11 electromechanical interlock

**Table 6.7 I/O Expander 4 Signals**

<sup>1</sup>. I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

## I/O Expander 5

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P10APN	Port 10 attention push button input
1 (I/O-0.1)	I	P10PDN	Port 10 presence detect input
2 (I/O-0.2)	I	P10PFN	Port 10 power fault input
3 (I/O-0.3)	I	P10MRLN	Port 10 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P10AIN	Port 10 attention indicator output
5 (I/O-0.5)	O	P10PIN	Port 10 power indicator output
6 (I/O-0.6)	O	P10PEP	Port 10 power enable output
7 (I/O-0.7)	O	P10ILOCKP	Port 10 electromechanical interlock

**Table 6.8 I/O Expander 5 Signals (Part 1 of 2)**

## Notes

SMBus I/O Expander Bit	Type	Signal	Description
8 (I/O-1.0)	I	P12APN	Port 12 attention push button input
9 (I/O-1.1)	I	P12PDN	Port 12 presence detect input
10 (I/O-1.2)	I	P12PFN	Port 12 power fault input
11 (I/O-1.3)	I	P12MRLN	Port 12 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P12AIN	Port 12 attention indicator output
13 (I/O-1.5)	O	P12PIN	Port 12 power indicator output
14 (I/O-1.6)	O	P12PEP	Port 12 power enable output
15 (I/O-1.7)	O	P12ILOCKP	Port 12 electromechanical interlock

**Table 6.8 I/O Expander 5 Signals (Part 2 of 2)**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

### I/O Expander 6

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P13APN	Port 13 attention push button input
1 (I/O-0.1)	I	P13PDN	Port 13 presence detect input
2 (I/O-0.2)	I	P13PFN	Port 13 power fault input
3 (I/O-0.3)	I	P13MRLN	Port 13 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P13AIN	Port 13 attention indicator output
5 (I/O-0.5)	O	P13PIN	Port 13 power indicator output
6 (I/O-0.6)	O	P13PEP	Port 13 power enable output
7 (I/O-0.7)	O	P13ILOCKP	Port 13 electromechanical interlock
8 (I/O-1.0)	I	P15APN	Port 15 attention push button input
9 (I/O-1.1)	I	P15PDN	Port 15 presence detect input
10 (I/O-1.2)	I	P15PFN	Port 15 power fault input
11 (I/O-1.3)	I	P15MRLN	Port 15 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P15AIN	Port 15 attention indicator output
13 (I/O-1.5)	O	P15PIN	Port 15 power indicator output
14 (I/O-1.6)	O	P15PEP	Port 15 power enable output
15 (I/O-1.7)	O	P15ILOCKP	Port 15 electromechanical interlock

**Table 6.9 I/O Expander 6 Signals**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

# Notes

## I/O Expander 7

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	I	P14APN	Port 14 attention push button input
1 (I/O-0.1)	I	P14PDN	Port 14 presence detect input
2 (I/O-0.2)	I	P14PFN	Port 14 power fault input
3 (I/O-0.3)	I	P14MRLN	Port 14 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P14AIN	Port 14 attention indicator output
5 (I/O-0.5)	O	P14PIN	Port 14 power indicator output
6 (I/O-0.6)	O	P14PEP	Port 14 power enable output
7 (I/O-0.7)	O	P14ILOCKP	Port 14 electromechanical interlock
8 (I/O-1.0)	I	-	Unused
9 (I/O-1.1)	I	-	Unused
10 (I/O-1.2)	I	-	Unused
11 (I/O-1.3)	I	-	Unused
12 (I/O-1.4)	O	-	Unused
13 (I/O-1.5)	O	-	Unused
14 (I/O-1.6)	O	-	Unused
15 (I/O-1.7)	O	-	Unused

**Table 6.10 I/O Expander 7 Signals**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

## I/O Expander 8

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	O	P0LINKUPN	Port 0 link up status output
1 (I/O-0.1)	O	P1LINKUPN	Port 1 link up status output
2 (I/O-0.2)	O	P2LINKUPN	Port 2 link up status output
3 (I/O-0.3)	O	P3LINKUPN	Port 3 link up status output
4 (I/O-0.4)	O	P4LINKUPN	Port 4 link up status output
5 (I/O-0.5)	O	P5LINKUPN	Port 5 link up status output
6 (I/O-0.6)	O	P6LINKUPN	Port 6 link up status output
7 (I/O-0.7)	O	P7LINKUPN	Port 7 link up status output
8 (I/O-1.0)	O	P8LINKUPN	Port 8 link up status output

**Table 6.11 I/O Expander 8 Signals (Part 1 of 2)**

# Notes

SMBus I/O Expander Bit	Type	Signal	Description
9 (I/O-1.1)	O	P9LINKUPN	Port 9 link up status output
10 (I/O-1.2)	O	P10LINKUPN	Port 10 link up status output
11 (I/O-1.3)	O	P11LINKUPN	Port 11 link up status output
12 (I/O-1.4)	O	P12LINKUPN	Port 12 link up status output
13 (I/O-1.5)	O	P13LINKUPN	Port 13 link up status output
14 (I/O-1.6)	O	P14LINKUPN	Port 14 link up status output
15 (I/O-1.7)	O	P15LINKUPN	Port 15 link up status output

**Table 6.11 I/O Expander 8 Signals (Part 2 of 2)**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

## I/O Expander 9

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>	O	P0ACTIVENN	Port 0 active status output
1 (I/O-0.1)	O	P1ACTIVENN	Port 1 active status output
2 (I/O-0.2)	O	P2ACTIVENN	Port 2 active status output
3 (I/O-0.3)	O	P3ACTIVENN	Port 3 active status output
4 (I/O-0.4)	O	P4ACTIVENN	Port 4 active status output
5 (I/O-0.5)	O	P5ACTIVENN	Port 5 active status output
6 (I/O-0.6)	O	P6ACTIVENN	Port 6 active status output
7 (I/O-0.7)	O	P7ACTIVENN	Port 7 active status output
8 (I/O-1.0)	O	P8ACTIVENN	Port 8 active status output
9 (I/O-1.1)	O	P9ACTIVENN	Port 9 active status output
10 (I/O-1.2)	O	P10ACTIVENN	Port 10 active status output
11 (I/O-1.3)	O	P11ACTIVENN	Port 11 active status output
12 (I/O-1.4)	O	P12ACTIVENN	Port 12 active status output
13 (I/O-1.5)	O	P13ACTIVENN	Port 13 active status output
14 (I/O-1.6)	O	P14ACTIVENN	Port 14 active status output
15 (I/O-1.7)	O	P15ACTIVENN	Port 15 active status output

**Table 6.12 I/O Expander 9 Signals**

<sup>1</sup>: I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

## Notes

### I/O Expander 10

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0) <sup>1</sup>		-	Unused
1 (I/O-0.1)	I	P1PWRGDN	Port 1 power good input
2 (I/O-0.2)	I	P2PWRGDN	Port 2 power good input
3 (I/O-0.3)	I	P3PWRGDN	Port 3 power good input
4 (I/O-0.4)	I	P4PWRGDN	Port 4 power good input
5 (I/O-0.5)	I	P5PWRGDN	Port 5 power good input
6 (I/O-0.6)	I	P6PWRGDN	Port 6 power good input
7 (I/O-0.7)	I	P7PWRGDN	Port 7 power good input
8 (I/O-1.0)	I	P8PWRGDN	Port 8 power good input
9 (I/O-1.1)	I	P9PWRGDN	Port 9 power good input
10 (I/O-1.2)	I	P10PWRGDN	Port 10 power good input
11 (I/O-1.3)	I	P11PWRGDN	Port 11 power good input
12 (I/O-1.4)	I	P12PWRGDN	Port 12 power good input
13 (I/O-1.5)	I	P13PWRGDN	Port 13 power good input
14 (I/O-1.6)	I	P14PWRGDN	Port 14 power good input
15 (I/O-1.7)	I	P15PWRGDN	Port 15 power good input

**Table 6.13 I/O Expander 10 Signals**

<sup>1</sup>: 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 PES22H16 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 fundamental reset (see section Fundamental Reset on page 3-4).

During the fundamental reset initialization sequence, the slave SMBus address is initialized. The address is specified by the SSMBADDR[5:3:1] signals as shown in Table 6.14.

Address Bit	Address Bit Value
1	SSMBADDR[1]
2	SSMBADDR[2]
3	SSMBADDR[3]

**Table 6.14 Slave SMBus Address When a Static Address is Selected (Part 1 of 2)**

# Notes

Address Bit	Address Bit Value
4	0
5	SSMBADDR[5]
6	1
7	1

**Table 6.14 Slave SMBus Address When a Static Address is Selected (Part 2 of 2)**

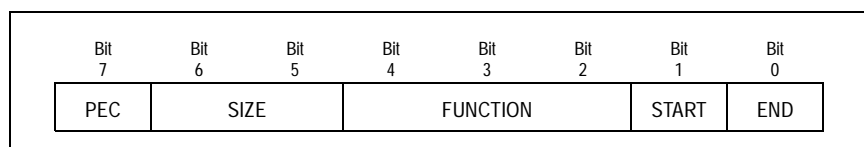
## SMBus Transactions

The slave SMBus interface responds to the following SMBus transactions initiated by an SMBus master. See the SMBus 2.0 specification 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 6.7 and described in Table 6.15.



**Figure 6.7 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.

**Table 6.15 Slave SMBus Command Code Fields (Part 1 of 2)**

## Notes

Bit Field	Name	Description
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 6.15 Slave SMBus Command Code Fields (Part 2 of 2)**

The FUNCTION field in the command code indicates if the SMBus operation is a CSR 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.

### CSR Register Read or Write Operation

Table 6.16 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	<b>Command Code.</b> Slave Command Code field described in Table 6.15.
1	BYTCNT	<b>Byte Count.</b> 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). <i>Note that the byte count field does not include the PEC byte if PEC is enabled.</i>
2	CMD	<b>Command.</b> This field encodes fields related to the CSR register read or write operation.
3	ADDRL	<b>Address Low.</b> Lower 8-bits of the doubleword CSR system address of register to access.
4	ADDRU	<b>Address Upper.</b> Upper 6-bits of the doubleword CSR system address of register to access. Bits 6 and 7 in the byte must be zero and are ignored by the hardware.
5	DATALL	<b>Data Lower.</b> Bits [7:0] of data doubleword.

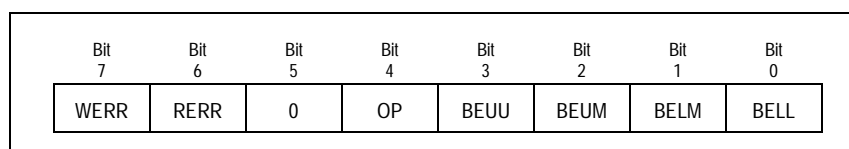
**Table 6.16 CSR Register Read or Write Operation Byte Sequence (Part 1 of 2)**

# Notes

Byte Position	Field Name	Description
6	DATA LM	<b>Data Lower Middle.</b> Bits [15:8] of data doubleword.
7	DATA UM	<b>Data Upper Middle.</b> Bits [23:16] of data doubleword.
8	DATA UU	<b>Data Upper.</b> Bits [31:24] of data doubleword.

**Table 6.16** CSR Register Read or Write Operation Byte Sequence (Part 2 of 2)

The format of the CMD field is shown in Figure 6.8 and described in Table 6.17.



**Figure 6.8** CSR Register Read or Write CMD Field Format

Bit Field	Name	Type	Description
0	BELL	Read/Write	<b>Byte Enable Lower.</b> When set, the byte enable for bits [7:0] of the data word is enabled.
1	BELM	Read/Write	<b>Byte Enable Lower Middle.</b> When set, the byte enable for bits [15:8] of the data word is enabled.
2	BEUM	Read/Write	<b>Byte Enable Upper Middle.</b> When set, the byte enable for bits [23:16] of the data word is enabled.
3	BEUU	Read/Write	<b>Byte Enable Upper.</b> 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	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 6.17** CSR Register Read or Write CMD Field Description

## Serial EEPROM Read or Write Operation

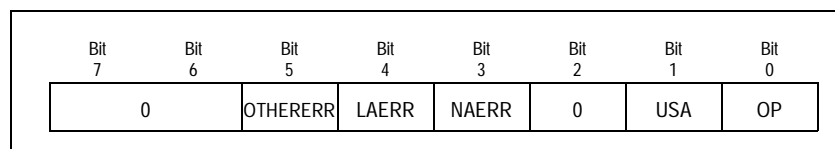
Table 6.17 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

# Notes

Byte Position	Field Name	Description
0	CCODE	<b>Command Code.</b> Slave Command Code field described in Table 6.15.
1	BYTCNT	<b>Byte Count.</b> 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).
2	CMD	<b>Command.</b> This field contains information related to the serial EEPROM transaction
3	EEADDR	<b>Serial EEPROM Address.</b> 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	<b>Address Low.</b> Lower 8-bits of the Serial EEPROM byte to access.
5	ADDRU	<b>Address Upper.</b> Upper 8-bits of the Serial EEPROM byte to access.
6	DATA	<b>Data.</b> Serial EEPROM value read or to be written.

**Table 6.18 Serial EEPROM Read or Write Operation Byte Sequence**

The format of the CMD field is shown in Figure 6.9 and described in Table 6.19.



**Figure 6.9 Serial EEPROM Read or Write CMD Field Format**

Bit Field	Name	Type <sup>1</sup>	Description
0	OP	RW	<b>Serial EEPROM Operation.</b> This field encodes the serial EEPROM operation to be performed. 0 - Serial EEPROM write 1 - Serial EEPROM read
1	USA	RW	<b>Use Specified Address.</b> When this bit is set the serial EEPROM SMBus address specified in the EEADDR is used instead of that specified in the ADDR field in the EEPROMINTF register.  When this bit is set the serial EEPROM SMBus address specified in the EEADDR is used instead of that specified in the MSMBADDR field in the SMBUSSTS register.
2	Reserved		

**Table 6.19 Serial EEPROM Read or Write CMD Field Description (Part 1 of 2)**

# Notes

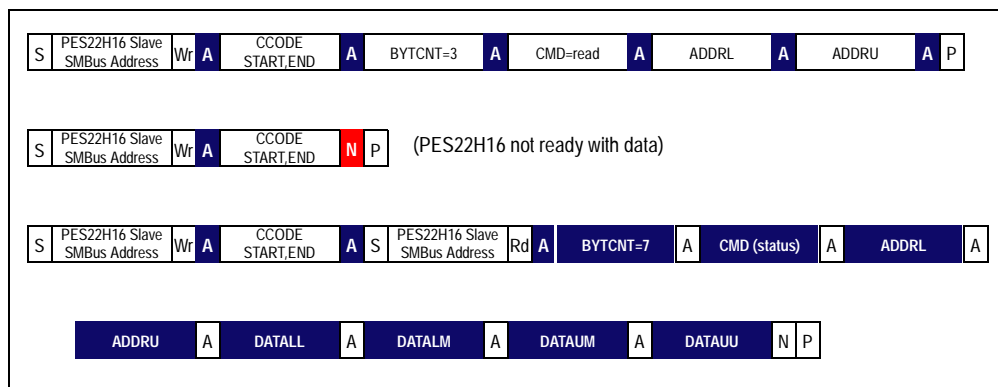
Bit Field	Name	Type <sup>1</sup>	Description
3	NAERR	RC	No Acknowledge Error. This bit is set if an unexpected NACK is observed during a master SMBus transaction when accessing the serial EEPROM. This bit has the same function as the NAERR bit in the SMBUSSTS register.  The setting of this bit may indicate 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, invalid data was detected by the slave.
4	LAERR	RC	Lost Arbitration Error. This bit is set if the master SMBus interface loses 16 consecutive arbitration attempts when accessing the serial EEPROM. This bit has the same function as the LAERR bit in the SMBUSSTS register.
5	OTHERERR	RC	Other Error. This bit is set if a misplaced START or STOP condition is detected by the master SMBus interface when accessing the serial EEPROM. This bit has the same function as the OTHERERR bit in the SMBUSSTS register.
7:6	Reserved	0	Reserved. Must be zero

**Table 6.19 Serial EEPROM Read or Write CMD Field Description (Part 2 of 2)**

<sup>1</sup> See Table 2 in the About This Manual chapter for a definition of these abbreviations.

## Sample Slave SMBus Operation

This section illustrates sample Slave SMBus operations. Shaded items are driven by the PES22H16's slave SMBus interface and non-shaded items are driven by an SMBus host.



**Figure 6.10 CSR Register Read Using SMBus Block Write/Read Transactions with PEC Disabled**

# Notes

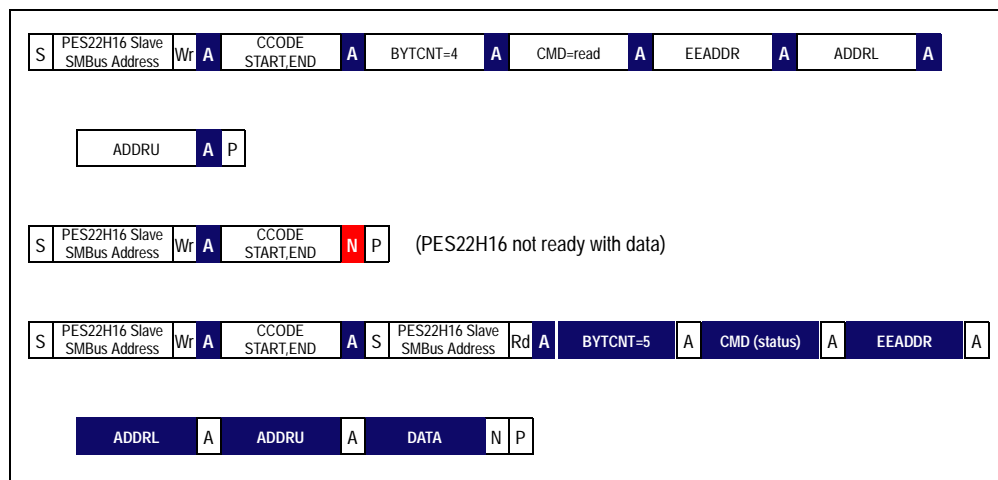


Figure 6.11 Serial EEPROM Read Using SMBus Block Write/Read Transactions with PEC Disabled

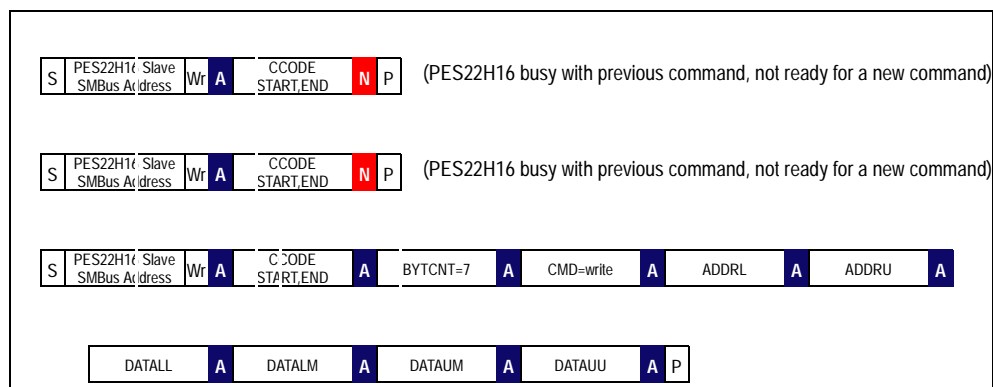


Figure 6.12 CSR Register Write Using SMBus Block Write Transactions with PEC Disabled

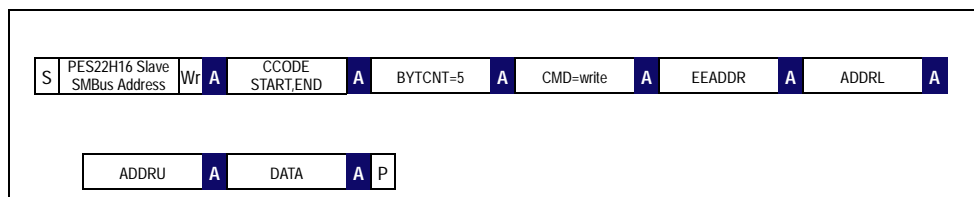


Figure 6.13 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Disabled

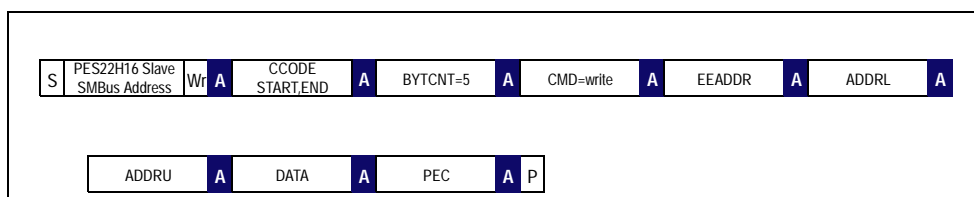


Figure 6.14 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Enabled

S	PES22H16 Slave SMBus Address	Wr	A	CCODE START, Word	A	CMD=read	A	ADDRL	A	P
---	---------------------------------	----	---	----------------------	---	----------	---	-------	---	---

S	PES22H16 Slave SMBus Address	Wr	A	CCODE END, Byte	A	ADDRU	A	P
---	---------------------------------	----	---	--------------------	---	-------	---	---

S	PES22H16 Slave SMBus Address	Wr	A	CCODE START, Word	N	P	(PES22H16 not ready with data)
---	---------------------------------	----	---	----------------------	---	---	--------------------------------

S	PES22H16 Slave SMBus Address	Wr	A	CCODE START, Word	A
---	---------------------------------	----	---	----------------------	---

S	PES22H16 Slave SMBus Address	Rd	A	CMD (status)	A	ADDRL	N	P
---	---------------------------------	----	---	--------------	---	-------	---	---

S	PES22H16 Slave SMBus Address	Wr	A	CCODE Byte	A
---	---------------------------------	----	---	---------------	---

S	PES22H16 Slave SMBus Address	Rd	A	ADDRU	A	P
---	---------------------------------	----	---	-------	---	---

S	PES22H16 Slave SMBus Address	Wr	A	CCODE Word	A
---	---------------------------------	----	---	---------------	---

S	PES22H16 Slave SMBus Address	Rd	A	DATALL	A	DATALM	N	P
---	---------------------------------	----	---	--------	---	--------	---	---

S	PES22H16 Slave SMBus Address	Wr	A	CCODE END, Word	A
---	---------------------------------	----	---	--------------------	---

S	PES22H16 Slave SMBus Address	Rd	A	DATAUM	A	DATAUU	N	P
---	---------------------------------	----	---	--------	---	--------	---	---

Figure 6.15 CSR Register Read Using SMBus Read and Write Transactions with PEC Disabled



# Power Management

## Notes

## Introduction

Located in configuration space of each PCI-PCI bridge in the PES22H16 is a power management capability structure. The power management capability structure associated with a PCI-PCI bridge of a downstream port only affects that port. Entering the D3<sub>Hot</sub> state allows the link associated with the bridge to enter the L1 state. The power management capability structure associated with the upstream port (i.e., port 0) affects the entire device. When the upstream port enters a low power state and the PME\_TO\_Ack messages are received, then the entire device is placed into a low power state.

The PES22H16 supports the following device power management states: D0 Uninitialized, D0 Active, D3<sub>Hot</sub>, and D3<sub>Cold</sub>. A power management state transition diagram for the states supported by the PES22H16 is provided in Figure 7.1 and described in Figure 7.1.

Transitioning a port's power management state from D3<sub>hot</sub> to D0<sub>uninitialized</sub> does not result in any logic being reset or re-initialization of register values. However, the default value of the No Soft Reset (NOSOFTRST) bit in the PCI Power Management Control and Status (PMCSR) register corresponds to the functional context being maintained in the D3<sub>hot</sub> state.

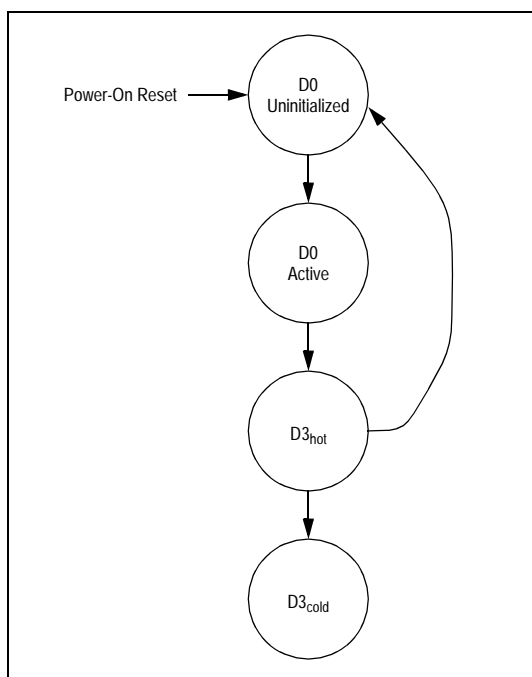


Figure 7.1 PES22H16 Power Management State Transition Diagram

## Notes

From State	To State	Description
any	D0 Uninitialized	Power-on fundamental reset.
D0 Uninitialized	D0 Active	PCI-PCI bridge configured by software
D0 Active	D3 <sub>hot</sub>	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 <sub>hot</sub> state.
D3 <sub>hot</sub>	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 <sub>hot</sub>	D3 <sub>cold</sub>	Power is removed from the device.

**Table 7.1 PES22H16 Power Management State Transition Diagram**

The PES22H16 PCI-to-PCI bridges (i.e., ports) have the following behavior when in the D3<sub>hot</sub> power management state:

- A bridge accepts, processes and completes all type 0 configuration read and write requests.
- A bridge accepts and processes all message requests that target the bridge.
- All requests received by the bridge on the primary interface, 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 bridge is not in D3<sub>hot</sub> (e.g, generation of an ERR\_NONFATAL message to the root).
- Error messages resulting from any event other than the receipt of a TLP are discarded (i.e., no error message is generated).
- All 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.
- All request TLPs received on the secondary interface are treated as unsupported requests (UR).

## PME Messages

The PES22H16 does not support generation of PME messages from the D3<sub>cold</sub> state. Downstream ports (i.e., PCI-PCI bridges associated with downstream ports) support the generation of hot-plug PME events (i.e., a PM\_PME power management message) from the D3<sub>hot</sub> state. This includes both the case when the downstream port is in the D3<sub>hot</sub> state or the entire switch is in the D3<sub>hot</sub> state.

The generation of a PME message by downstream ports necessitates the implementation of a PME service time-out mechanism to ensure that PME messages are not lost. If the PME Status (PMES) bit in the a downstream port's PCI Power Management Control and Status (PMCSR) register is not cleared within the time-out period specified in the PM\_PME Time-Out (PMPMETO) field in the ports PM\_PME Timer (PMPMETIMER) register after a PM\_PME message is transmitted, then the PM\_PME message is retransmitted and the timer is restarted.

## Power Express Power Management Fence Protocol

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.
- Devices acknowledge the PME\_Turn\_Off message by returning a PME\_TO\_ACK message

## Notes

When the PES22H16 receives a PME\_Turn\_Off message it broadcasts the PME\_Turn\_Off message on all active downstream ports. The PES22H16 transmits a PME\_TO\_Ack message on its upstream port and transitions its link state to L2/L3 Ready after it has received a PME\_TO\_Ack message on each of its downstream ports.

In order to avoid deadlock, a downstream port that does not receive a PME\_TO\_Ack message in the time-out period specified in the PME\_TO\_Ack Time-Out (PMETOATO) field in its corresponding PME\_TO\_Ack Timer (PMETOATIMER) register, declares a time-out, transitions its link to L2/L3 Ready and signals to the upstream port that a PME\_TO\_Ack message has been received. If instead of being transitioned to the D3<sub>cold</sub> state, the PES22H16 is transitioned to the D0<sub>uninitialized</sub> state, then the PES22H16 resumes generation of PM\_PME messages.

### Power Budgeting Capability

The PES22H16 contains the mechanisms necessary to implement the PCI express power budgeting enhanced capability. However, by default, these mechanisms are not enabled. To enable the power budgeting capability, registers in this capability should be initialized and the Next Pointer (NXTPTR) field in the PCI Express VC Enhanced Capability Header (PCIEVCECAP) register should be initialized to point to the power budgeting capability.

The power budgeting capability consists of the four power budgeting capability registers defined in the PCIe 1.1 base specification and eight general purpose read-write registers. See section Power Budgeting Enhanced Capability on page 9-57 for a description of these registers. The Power Budgeting Capabilities (PWRBCAP) register contains the PCI express enhanced capability header for the power budgeting capability. By default, this register has an initial read-only value of zero. To enable the power budgeting capability, this register should be initialized via the serial EEPROM.

The Power Budgeting Data Value [0..7] (PWRBDV[0..7]) registers are used to hold the power budgeting information for that port in a particular operating condition. The PWRBDV registers may be read and written when the Power Budgeting Data Value Unlock (PWRBDVUL) bit is set in the System Control (SYSCTL) register. When the PWRBDVUL bit is cleared, these register are read-only and writes to these registers are ignored. To enable the power budgeting capability, the PWRBDV registers should be initialized with power budgeting information via the serial EEPROM.

Notes



# Hot-Plug and Hot-Swap

## Notes

## Introduction

As illustrated in Figures 8.1 through 8.3, a PCIe switch may be used in one of three hot-plug configurations. Figure 8.1 illustrates the use of the PES22H16 in an application in which two downstream ports are connected to slots into which add-in cards may be hot-plugged. Figure 8.2 illustrates the use of the PES22H16 in an add-in card application. Here the downstream ports are hardwired to devices on the add-in card and the upstream port serves as the add-in card's PCIe interface. In this application the upstream port may be hot-plugged into a slot on the main system. Finally, Figure 8.3 illustrates the use of the PES22H16 in a carrier card application. In this application, the downstream 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 8.1 through 8.2, it will not be discussed further.

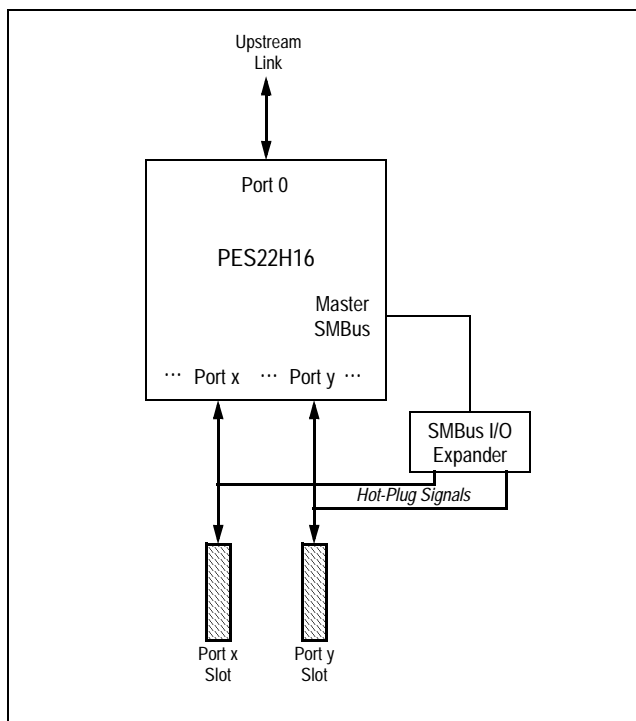


Figure 8.1 Hot-Plug on Switch Downstream Slots Application

# Notes

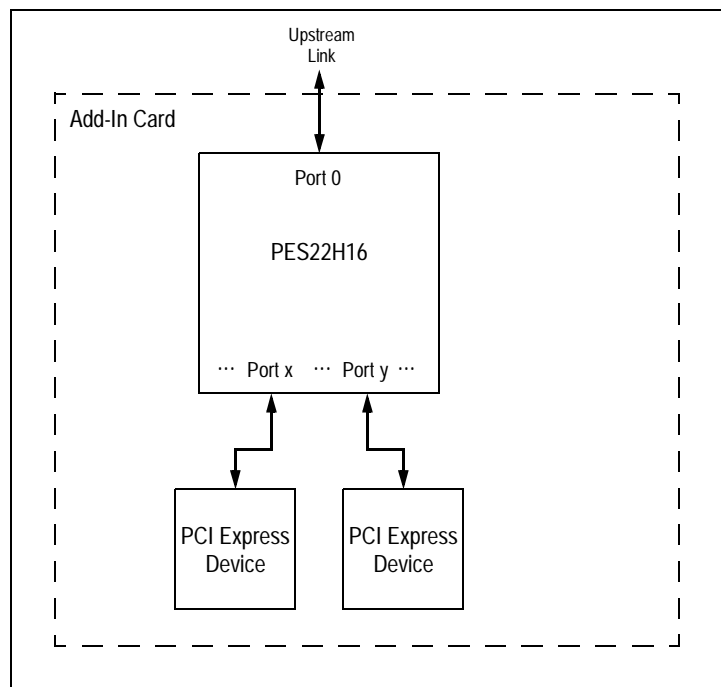


Figure 8.2 Hot-Plug with Switch on Add-In Card Application

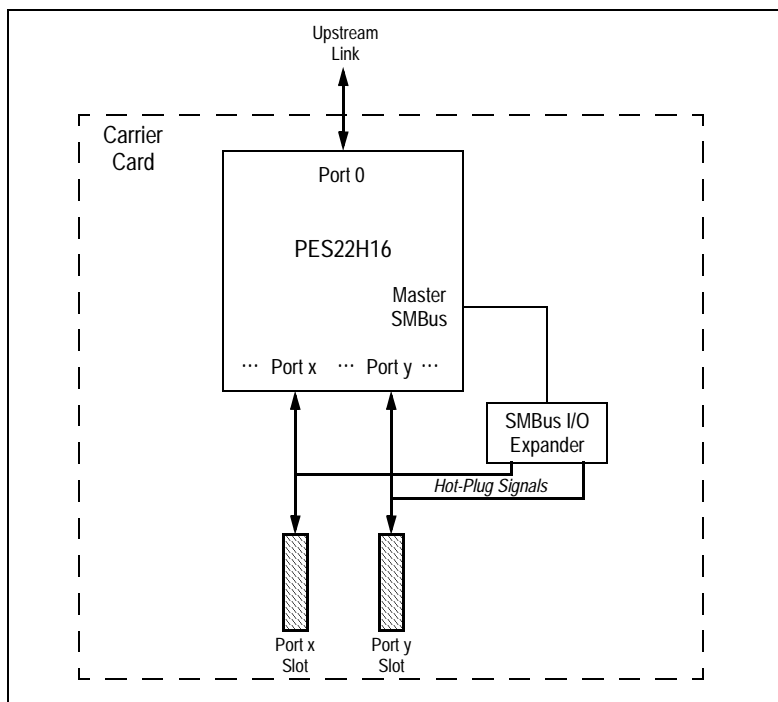


Figure 8.3 Hot-Plug with Carrier Card Application

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 PES22H16.

# Notes

The remainder of this section discusses the use of the PES22H16 in an application in which one or more of the downstream ports are used in an application in which an add-in card may be hot-plugged into a downstream slot. Associated with each downstream port in the PES22H16 is a hot-plug controller. The hot-plug controller may be enabled by setting the HPC bit in the PCI Express Slot Capabilities (PCIESCAP) register associated with that port during configuration (e.g., via serial EEPROM).

The PES22H16 allows 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 PES22H16's downstream port. Table 8.1 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. The PES22H16 only supports presence detect signalling via a pin assertion. It does not support in-band presence detect.

Signal	Type	Name/Description
PxAPN	I	Port x <sup>1</sup> Attention Push button Input.
PxPDN	I	Port x Presence Detect Input.
PxPFN	I	Port x Power Fault Input.
PxMRLN	I	Port x Manually-operated Retention Latch (MRL) Input.
PxAIn	O	Port x Attention Indicator Output.
PxPIN	O	Port x Power Indicator Output.
PxPEP	O	Port x Power Enable Output.
PxILOCKP	O	Port x Electromechanical Interlock.
PxPWRGDN	I	Port x Power Good Input (asserted when slot power is good).
PxRSTN <sup>2</sup>	O	Port x Reset Output.

**Table 8.1 Downstream Port Hot-Plug Signals**

<sup>1</sup> x corresponds to downstream port number (i.e., 1 through 15).

<sup>2</sup> This signal is a GPIO pin alternate function and is not available as an I/O expander output.

Since the polarity of hot-plug signals has been defined differently in various specifications, each hot-plug signal 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 EIC bit in the PCIESCTL register, then the PxILOCKP signal is pulsed for with a pulse 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 (TEMICL) bit in the HPCFGCTL register is set, writing a one to the EIC bit inverts the state of the PxILOCKP signal.

When the Replace MRL Status with EMIL Status (RMRLWEMIL) bit is set in the HPCFGCTL register, then the port's PxMRLN input is used as the electromechanical state input. The state of this input is used as the state of the electromechanical interlock state obtained by reading the Electromechanical Interlock Status (EIS) bit in the PCI Express Slot Status (PCIESSTS) register. In this mode the state of the Manually-operated Retention Latch Sensor State (MRLSS) status is always reported as closed (i.e., zero). When the EMIL bit is cleared, the EIS bit state in the PCIESSTS register always returns a value of zero when read.

When the MRL Automatic Power Off (MRLPWROFF) bit is set in the HPCFGCTL register 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.

## Notes

The state of a port's Power Fault (PxPFN) input is not latched by the PES22H16. 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 Express-Module form factor. Downstream port reset outputs are described in section Downstream Port Reset Outputs on page 3-8.

The default value of hot-plug registers following a hot or 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 PCI Express Slot Status (PCIESSTS) register as a result of serial EEPROM initialization.

### Hot-Plug I/O Expander

the PES22H16 utilizes external SMBus/I2C-bus I/O expanders connected to the master SMBus interface for hot-plug related signals associated with downstream ports. See section I/O Expanders on page 6-6 for details on the operation of the I/O expanders and for the mapping of downstream hot-plug signals to I/O expander inputs and outputs.

### Hot-Plug Interrupts and Wake-up

The hot-plug controller associated with a downstream slot may generate an interrupt or wake-up event. Hot-plug interrupts are only generated 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), and Command Completed (CC).

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 PCI Command (PCICMD) register. When the downstream port or the entire switch is in a D3<sub>hot</sub> state, then the hot-plug controller generates a wake-up event using a PM\_PME message instead of an interrupt if the event interrupt is not masked in the slot control (PCIESCTL) register and hot-plug interrupts are disabled by the HPIE bit. If the event interrupt is not masked and hot-plug interrupts are enabled, then both a PM\_PME and an interrupt are generated. If the event interrupt is masked, then neither a PM\_PME or interrupt are generated. Note that a command completed (CC bit) interrupt will not generate a wake-up event.

### Legacy System Hot-Plug Support

Some systems require support for operating systems that lack PCIe hot-plug support. The PES22H16 supports these systems by providing a General Purpose Event (GPEN) output as an alternate function of GPIO[5] that can be used instead of the INTx, MSI, and PME mechanisms defined by PCI Express hot-plug.

Associated with each downstream port's hot-plug controller is a bit in the General Purpose Event Control (P0\_GPECTL) register. When this bit is set, then the corresponding PCIe base 1.1 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.

GPEN is an alternate function of GPIO[5] and GPIO[5] 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 in the General Purpose Event Status (P0\_GPESTS) register is set. A bit in the P0\_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 signalling mechanism is the only thing that is affected when a port is configured to use general purpose events instead of the PCIe defined hot-plug signalling mechanisms (i.e., INTx, MSI and PME). Thus, the PCIe defined capability, status and mask bits defined in the PCIe 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.

The enhanced hot-plug signalling mechanism supported by the PES22H16 is graphically illustrated in Figure 8.4. This figure provides a conceptual summary of the enhanced hot-plug signalling mechanism in the form of a pseudo logic diagram. Logic gates in this diagram are intended for conveying general concepts, and not for direct implementation.

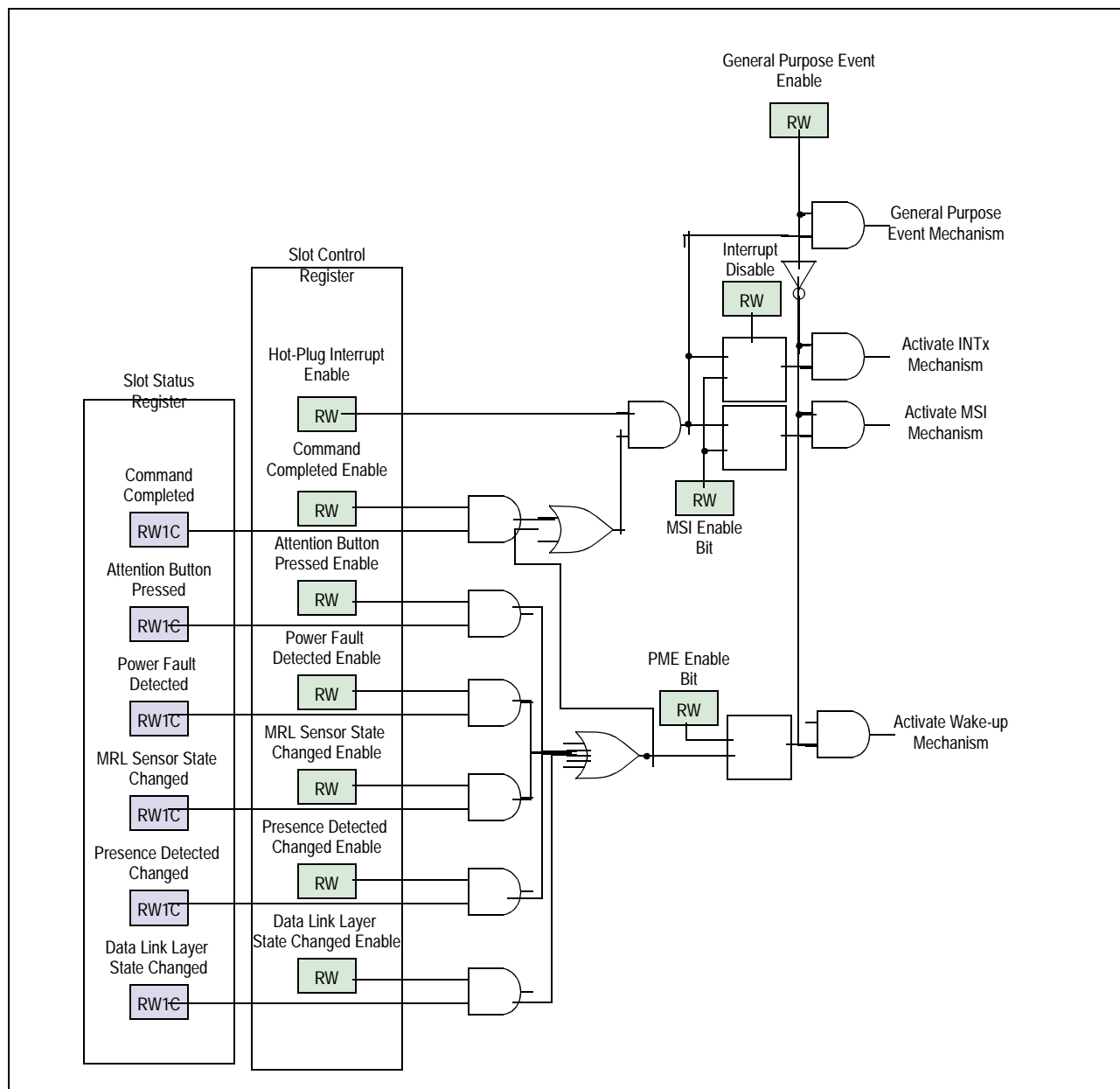


Figure 8.4 PES22H16 Hot-Plug Event Signalling

## Notes

## Hot-Swap

The PES22H16 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, PES22H16 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 PES22H16 may also be used to construct proprietary hot-swap systems. See the PES22H16 Data Sheet for a detailed specification of I/O buffer characteristics.



# Configuration Registers

## Notes

## Configuration Space Organization

Each software visible registers in the PES22H16 is contained in the PCI configuration space of one of the ports. Thus, there are no registers in the PES22H16 that cannot be accessed by the root. Each software visible register in the PES22H16 has a system address. The system address is formed by adding the PCI configuration space offset value of the register to the base address of the port in which it is located. The system address is used for serial EEPROM register initialization and slave SMBus register accesses.

The base address for each PES22H16 port is listed in Table 9.1. The PCI configuration space offset addresses for registers in the upstream port are listed in Table 9.2 while the PCI configuration space offset addresses for registers in downstream ports are listed Table 9.3.

Base Address	PCI Configuration Space
0x0000	Port 0 configuration space (upstream port)
0x1000	Port 1 configuration space (downstream port)
0x2000	Port 2 configuration space (downstream port)
0x3000	Port 3 configuration space (downstream port)
0x4000	Port 4 configuration space (downstream port)
0x5000	Port 5 configuration space (downstream port)
0x6000	Port 6 configuration space (downstream port)
0x7000	Port 7 configuration space (downstream port)
0x8000	Port 8 configuration space (downstream port)
0x9000	Port 9 configuration space (downstream port)
0xA000	Port 10 configuration space (downstream port)
0xB000	Port 11 configuration space (downstream port)
0xC000	Port 12 configuration space (downstream port)
0xD000	Port 13 configuration space (downstream port)
0xE000	Port 14 configuration space (downstream port)
0xF000	Port 15 configuration space (downstream port)

**Table 9.1 Base Addresses for Port Configuration Space Registers**

As shown in Figure 9.1, upstream and downstream ports share a similar PCI configuration space register layout. The upstream port contains global switch control and status registers as well as test mode registers which are not present in the configuration space of downstream ports. Due to the ability to generate MSIs as a result of hot-plug events, the downstream ports contain an MSI capability structure which is not present in the upstream port.

PCIe configuration reads to an upstream port offset not defined in Table 9.2 or a downstream port offset not defined in Table 9.3 return a value of zero. Slave SMBus reads to these offsets return an undefined data value. PCIe configuration writes or Slave SMBus writes to an offset not defined in Table 9.2 or Table 9.3 complete successfully but modify no data and have no other effect.

# Notes

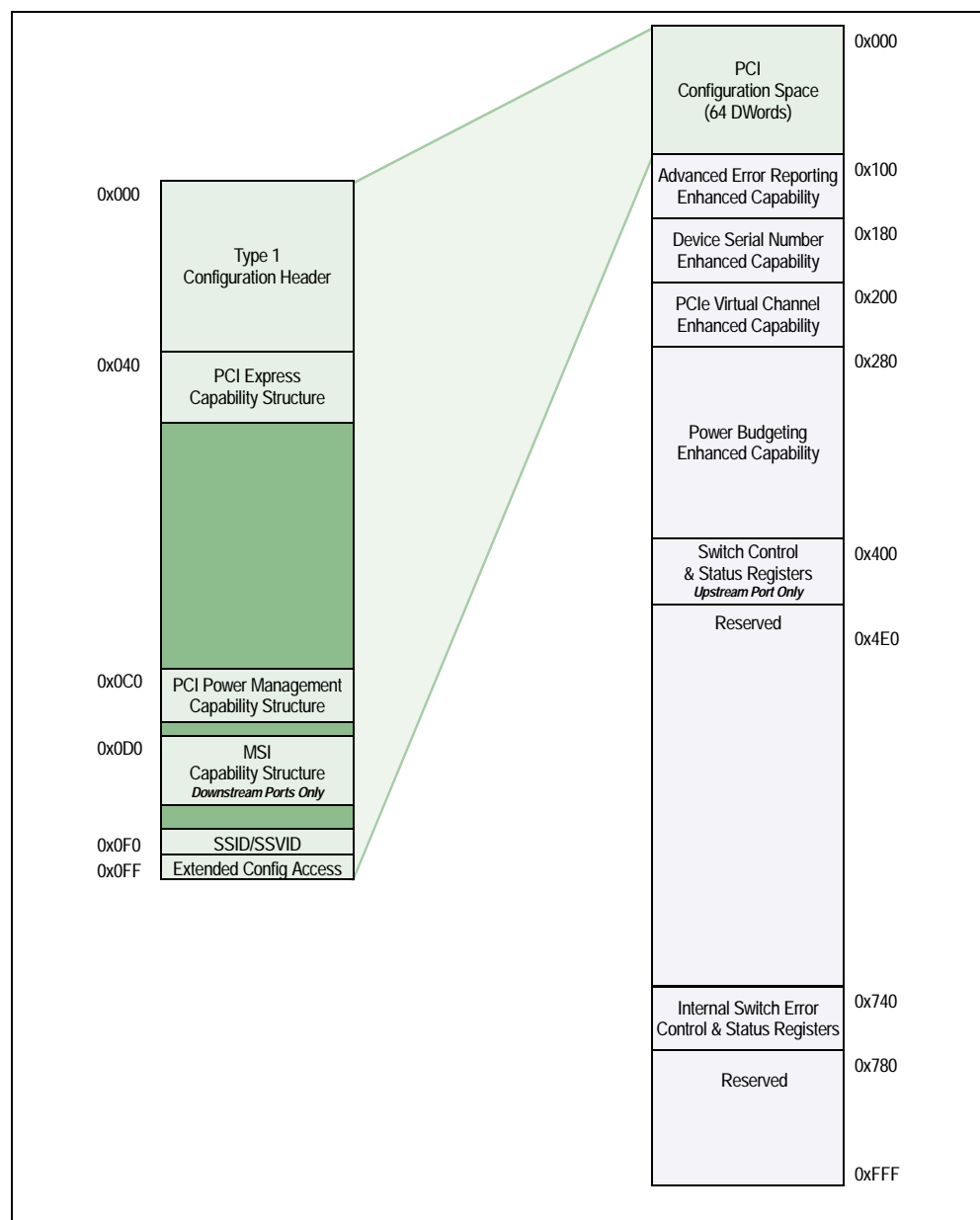


Figure 9.1 Port Configuration Space Organization

# Notes

## Upstream Port (Port 0)

**Note:** In pdf format, clicking on a register name in the Register Definition column creates a jump to the appropriate register. To return to the starting place in this table, click on the same register name (in blue) in the register section.

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x000	Word	P0_VID	<a href="#">VID - Vendor Identification Register (0x000) on page 9-11</a>
0x002	Word	P0_DID	<a href="#">DID - Device Identification Register (0x002) on page 9-12</a>
0x004	Word	P0_PCICMD	<a href="#">PCICMD - PCI Command Register (0x004) on page 9-12</a>
0x006	Word	P0_PCISTS	<a href="#">PCISTS - PCI Status Register (0x006) on page 9-13</a>
0x008	Byte	P0_RID	<a href="#">RID - Revision Identification Register (0x008) on page 9-14</a>
0x009	3 Bytes	P0_CC CODE	<a href="#">CCODE - Class Code Register (0x009) on page 9-14</a>
0x00C	Byte	P0_CLS	<a href="#">CLS - Cache Line Size Register (0x00C) on page 9-14</a>
0x00D	Byte	P0_PLTIMER	<a href="#">PLTIMER - Primary Latency Timer (0x00D) on page 9-15</a>
0x00E	Byte	P0_HDR	<a href="#">HDR - Header Type Register (0x00E) on page 9-15</a>
0x00F	Byte	P0_BIST	<a href="#">BIST - Built-in Self Test Register (0x00F) on page 9-15</a>
0x010	DWord	P0_BAR0	<a href="#">BAR0 - Base Address Register 0 (0x010) on page 9-15</a>
0x014	DWord	P0_BAR1	<a href="#">BAR1 - Base Address Register 1 (0x014) on page 9-15</a>
0x018	Byte	P0_PBUSN	<a href="#">PBUSN - Primary Bus Number Register (0x018) on page 9-15</a>
0x019	Byte	P0_SBUSN	<a href="#">SBUSN - Secondary Bus Number Register (0x019) on page 9-16</a>
0x01A	Byte	P0_SUBUSN	<a href="#">SUBUSN - Subordinate Bus Number Register (0x01A) on page 9-16</a>
0x01B	Byte	P0_SLTIMER	<a href="#">SLTIMER - Secondary Latency Timer Register (0x01B) on page 9-16</a>
0x01C	Byte	P0_IOBASE	<a href="#">IOBASE - I/O Base Register (0x01C) on page 9-16</a>
0x01D	Byte	P0_IOLIMIT	<a href="#">IOLIMIT - I/O Limit Register (0x01D) on page 9-17</a>
0x01E	Word	P0_SECSTS	<a href="#">SECSTS - Secondary Status Register (0x01E) on page 9-17</a>
0x020	Word	P0_MBASE	<a href="#">MBASE - Memory Base Register (0x020) on page 9-18</a>
0x022	Word	P0_MLIMIT	<a href="#">MLIMIT - Memory Limit Register (0x022) on page 9-18</a>
0x024	Word	P0_PMBASE	<a href="#">PMBASE - Prefetchable Memory Base Register (0x024) on page 9-18</a>
0x026	Word	P0_PMLIMIT	<a href="#">PMLIMIT - Prefetchable Memory Limit Register (0x026) on page 9-19</a>
0x028	DWord	P0_PMBASEU	<a href="#">PMBASEU - Prefetchable Memory Base Upper Register (0x028) on page 9-19</a>
0x02C	DWord	P0_PMLIMITU	<a href="#">PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) on page 9-19</a>
0x030	Word	P0_IOBASEU	<a href="#">IOBASEU - I/O Base Upper Register (0x030) on page 9-19</a>
0x032	Word	P0_IOLIMITU	<a href="#">IOLIMITU - I/O Limit Upper Register (0x032) on page 9-20</a>
0x034	Byte	P0_CAPPTR	<a href="#">CAPPTR - Capabilities Pointer Register (0x034) on page 9-20</a>
0x038	DWord	P0_EROMBASE	<a href="#">EROMBASE - Expansion ROM Base Address Register (0x038) on page 9-20</a>

Table 9.2 Upstream Port 0 Configuration Space Registers (Part 1 of 5)

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x03C	Byte	P0_INTRLINE	INTRLINE - Interrupt Line Register (0x03C) on page 9-20
0x03D	Byte	P0_INTRPIN	INTRPIN - Interrupt PIN Register (0x03D) on page 9-21
0x03E	Word	P0_BCTL	BCTL - Bridge Control Register (0x03E) on page 9-21
0x040	DWord	P0_PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 9-22
0x044	DWord	P0_PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 9-23
0x048	Word	P0_PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 9-24
0x04A	Word	P0_PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 9-25
0x04C	DWord	P0_PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 9-26
0x050	Word	P0_PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 9-27
0x052	Word	P0_PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 9-28
0x064	DWord	P0_PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 9-33
0x068	Word	P0_PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068) on page 9-33
0x06A	Word	P0_PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 9-34
0x06C	DWord	P0_PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 9-34
0x070	Word	P0_PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 9-34
0x072	Word	P0_PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 9-34
0x0C0	DWord	P0_PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 9-35
0x0C4	DWord	P0_PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 9-36
0x0F0	Dword	P0_SSIDSSVIDCAP	SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) on page 9-38
0x0F4	Dword	P0_SSIDSSVID	SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) on page 9-38
0x0F8	Word	P0_ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 9-38
0x0FC	Word	P0_ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 9-39
0x100	Dword	P0_AERCAP	AERCAP - AER Capabilities (0x100) on page 9-39
0x104	Dword	P0_AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 9-40
0x108	Dword	P0_AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 9-41
0x10C	Dword	P0_AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 9-42
0x110	Dword	P0_AERCES	AERCES - AER Correctable Error Status (0x110) on page 9-43
0x114	Dword	P0_AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 9-44
0x118	Dword	P0_AERCTL	AERCTL - AER Control (0x118) on page 9-45
0x11C	Dword	P0_AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 9-45
0x120	Dword	P0_AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 9-45

**Table 9.2 Upstream Port 0 Configuration Space Registers (Part 2 of 5)**

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x124	Dword	P0_AERHL3DW	<a href="#">AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 9-45</a>
0x128	Dword	P0_AERHL4DW	<a href="#">AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 9-46</a>
0x180	Dword	P0_SNUMCAP	<a href="#">SNUMCAP - Serial Number Capabilities (0x180) on page 9-46</a>
0x184	Dword	P0_SNUMLDW	<a href="#">SNUMLDW - Serial Number Lower Doubleword (0x184) on page 9-46</a>
0x188	Dword	P0_SNUMUDW	<a href="#">SNUMUDW - Serial Number Upper Doubleword (0x188) on page 9-46</a>
0x200	DWord	P0_PCIEVCECAP	<a href="#">PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200) on page 9-47</a>
0x204	DWord	P0_PVCCAP1	<a href="#">PVCCAP1 - Port VC Capability 1 (0x204) on page 9-47</a>
0x208	DWord	P0_PVCCAP2	<a href="#">PVCCAP2 - Port VC Capability 2 (0x208) on page 9-47</a>
0x20C	Word	P0_PVCCTL	<a href="#">PVCCTL - Port VC Control (0x20C) on page 9-48</a>
0x20E	Word	P0_PVCSTS	<a href="#">PVCSTS - Port VC Status (0x20E) on page 9-48</a>
0x210	DWord	P0_VCR0CAP	<a href="#">VCR0CAP - VC Resource 0 Capability (0x210) on page 9-49</a>
0x214	DWord	P0_VCR0CTL	<a href="#">VCR0CTL - VC Resource 0 Control (0x214) on page 9-49</a>
0x218	DWord	P0_VCR0STS	<a href="#">VCR0STS - VC Resource 0 Status (0x218) on page 9-50</a>
0x21c	DWord	P0_VCR1CAP	<a href="#">VCR1CAP - VC Resource 1 Capability (0x21C) on page 9-50</a>
0x220	DWord	P0_VCR1CTL	<a href="#">VCR1CTL - VC Resource 1 Control (0x220) on page 9-51</a>
0x224	DWord	P0_VCR1STS	<a href="#">VCR1STS - VC Resource 1 Status (0x224) on page 9-52</a>
0x230	DWord	P0_VCR0TBL0	<a href="#">VCR0TBL0 - VC Resource 0 Arbitration Table Entry 0 (0x230) on page 9-52</a>
0x234	DWord	P0_VCR0TBL1	<a href="#">VCR0TBL1 - VC Resource 0 Arbitration Table Entry 1 (0x234) on page 9-53</a>
0x238	DWord	P0_VCR0TBL2	<a href="#">VCR0TBL2 - VC Resource 0 Arbitration Table Entry 2 (0x238) on page 9-53</a>
0x23C	DWord	P0_VCR0TBL3	<a href="#">VCR0TBL3 - VC Resource 0 Arbitration Table Entry 3 (0x23C) on page 9-54</a>
0x240	DWord	P0_VCR1TBL0	<a href="#">VCR1TBL0 - VC Resource 1 Arbitration Table Entry 0 (0x240) on page 9-55</a>
0x244	DWord	P0_VCR1TBL1	<a href="#">VCR1TBL1 - VC Resource 1 Arbitration Table Entry 1 (0x244) on page 9-55</a>
0x248	DWord	P0_VCR1TBL2	<a href="#">VCR1TBL2 - VC Resource 1 Arbitration Table Entry 2 (0x248) on page 9-56</a>
0x24C	DWord	P0_VCR1TBL3	<a href="#">VCR1TBL3 - VC Resource 1 Arbitration Table Entry 3 (0x24C) on page 9-56</a>
0x280	Dword	P0_PWRBCAP	<a href="#">PWRBCAP - Power Budgeting Capabilities (0x280) on page 9-57</a>
0x284	Dword	P0_PWRBDSEL	<a href="#">PWRBDSEL - Power Budgeting Data Select (0x284) on page 9-57</a>
0x288	Dword	P0_PWRBD	<a href="#">PWRBD - Power Budgeting Data (0x288) on page 9-58</a>
0x28C	Dword	P0_PWRBPBC	<a href="#">PWRBPBC - Power Budgeting Power Budget Capability (0x28C) on page 9-58</a>

Table 9.2 Upstream Port 0 Configuration Space Registers (Part 3 of 5)

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x300	Dword	P0_PWRBDV0	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x304	Dword	P0_PWRBDV1	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x308	Dword	P0_PWRBDV2	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x30C	Dword	P0_PWRBDV3	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x310	Dword	P0_PWRBDV4	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x314	Dword	P0_PWRBDV5	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x318	Dword	P0_PWRBDV6	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x31C	Dword	P0_PWRBDV7	PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58
0x400	DWord	SWSTS	SWSTS - Switch Status (0x400) on page 9-59
0x404	DWord	SWCTL	SWCTL - Switch Control (0x404) on page 9-61
0x408	DWord	HPCFGCTL	HPCFGCTL - Hot-Plug Configuration Control (0x408) on page 9-62
0x40C	DWord	GPR	GPR - General Purpose Register (0x40C) on page 9-63
0x418	DWord	GPIOFUNC	GPIOFUNC - General Purpose I/O Control Function (0x418) on page 9-63
0x41C	DWord	GPIOCFG	GPIOCFG - General Purpose I/O Configuration (0x41C) on page 9-63
0x420	DWord	GIOD	GIOD - General Purpose I/O Data (0x420) on page 9-64
0x424	DWord	SMBUSSTS	SMBUSSTS - SMBus Status (0x424) on page 9-64
0x428	DWord	SMBUSCTL	SMBUSCTL - SMBus Control (0x428) on page 9-65
0x42C	DWord	EEPROMINTF	EEPROMINTF - Serial EEPROM Interface (0x42C) on page 9-66
0x430	DWord	IOEXPINTF	IOEXPINTF - I/O Expander Interface (0x430) on page 9-67
0x434	DWord	IOEXPADDR0	IOEXPADDR0 - SMBus I/O Expander Address 0 (0x434) on page 9-68
0x438	DWord	IOEXPADDR1	IOEXPADDR1 - SMBus I/O Expander Address 1 (0x438) on page 9-69
0x43C	DWord	IOEXPADDR2	IOEXPADDR2 - SMBus I/O Expander Address 2 (0x43C) on page 9-69
0x450	DWord	GPECTL	GPECTL - General Purpose Event Control (0x450) on page 9-70
0x454	DWord	GPESTS	GPESTS - General Purpose Event Status (0x454) on page 9-72
0x470	DWord	USPFSTS	USPFSTS - Upstream Port Failover Status (0x470) on page 9-74
0x474	DWord	USPFCTL	USPFCTL - Upstream Port Failover Control (0x474) on page 9-75
0x478	DWord	USPFTIMER	USPFTIMER - Upstream Port Failover Watchdog Timer (0x478) on page 9-75
0x740	Dword	P0_SWPECTL	SWPECTL - Switch Parity Error Control (0x740) on page 9-76

Table 9.2 Upstream Port 0 Configuration Space Registers (Part 4 of 5)

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x744	Dword	P0_SWPESTS	<a href="#">SWPESTS - Switch Parity Error Status (0x744) on page 9-76</a>
0x748	Dword	P0_SWPERCTL	<a href="#">SWPERCTL - Switch Parity Error Reporting Control (0x748) on page 9-77</a>
0x74C	Dword	P0_SWPECNT	<a href="#">SWPECNT - Switch Parity Error Count (0x74C) on page 9-77</a>
0x754	Dword	P0_SWTOSTS	<a href="#">SWTOSTS - Switch Time-Out Status (0x754) on page 9-77</a>
0x758	Dword	P0_SWTORCTL	<a href="#">SWTORCTL - Switch Time-Out Reporting Control (0x758) on page 9-78</a>
0x75C	Dword	P0_SWTOCNT	<a href="#">SWTOCNT - Switch Time-Out Count (0x75C) on page 9-79</a>

**Table 9.2 Upstream Port 0 Configuration Space Registers (Part 5 of 5)**

## Notes

### Downstream Ports (Ports 1 through 15)

**Note:** In pdf format, clicking on a register name in the Register Definition column creates a jump to the appropriate register. To return to the starting place in this table, click on the same register name (in blue) in the register section.

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x000	Word	Px_VID	<a href="#">VID - Vendor Identification Register (0x000) on page 9-11</a>
0x002	Word	Px_DID	<a href="#">DID - Device Identification Register (0x002) on page 9-12</a>
0x004	Word	Px_PCICMD	<a href="#">PCICMD - PCI Command Register (0x004) on page 9-12</a>
0x006	Word	Px_PCISTS	<a href="#">PCISTS - PCI Status Register (0x006) on page 9-13</a>
0x008	Byte	Px_RID	<a href="#">RID - Revision Identification Register (0x008) on page 9-14</a>
0x009	3 Bytes	Px_CC CODE	<a href="#">CC CODE - Class Code Register (0x009) on page 9-14</a>
0x00C	Byte	Px_CLS	<a href="#">CLS - Cache Line Size Register (0x00C) on page 9-14</a>
0x00D	Byte	Px_PLTIMER	<a href="#">PLTIMER - Primary Latency Timer (0x00D) on page 9-15</a>
0x00E	Byte	Px_HDR	<a href="#">HDR - Header Type Register (0x00E) on page 9-15</a>
0x00F	Byte	Px_BIST	<a href="#">BIST - Built-in Self Test Register (0x00F) on page 9-15</a>
0x010	DWord	Px_BAR0	<a href="#">BAR0 - Base Address Register 0 (0x010) on page 9-15</a>
0x014	DWord	Px_BAR1	<a href="#">BAR1 - Base Address Register 1 (0x014) on page 9-15</a>
0x018	Byte	Px_PBUSN	<a href="#">PBUSN - Primary Bus Number Register (0x018) on page 9-15</a>
0x019	Byte	Px_SBUSN	<a href="#">SBUSN - Secondary Bus Number Register (0x019) on page 9-16</a>
0x01A	Byte	Px_SUBUSN	<a href="#">SUBUSN - Subordinate Bus Number Register (0x01A) on page 9-16</a>
0x01B	Byte	Px_SLTIMER	<a href="#">SLTIMER - Secondary Latency Timer Register (0x01B) on page 9-16</a>
0x01C	Byte	Px_IOBASE	<a href="#">IOBASE - I/O Base Register (0x01C) on page 9-16</a>
0x01D	Byte	Px_IOLIMIT	<a href="#">IOLIMIT - I/O Limit Register (0x01D) on page 9-17</a>
0x01E	Word	Px_SECSTS	<a href="#">SECSTS - Secondary Status Register (0x01E) on page 9-17</a>
0x020	Word	Px_MBASE	<a href="#">MBASE - Memory Base Register (0x020) on page 9-18</a>
0x022	Word	Px_MLIMIT	<a href="#">MLIMIT - Memory Limit Register (0x022) on page 9-18</a>
0x024	Word	Px_PMBASE	<a href="#">PMBASE - Prefetchable Memory Base Register (0x024) on page 9-18</a>
0x026	Word	Px_PMLIMIT	<a href="#">PMLIMIT - Prefetchable Memory Limit Register (0x026) on page 9-19</a>
0x028	DWord	Px_PMBASEU	<a href="#">PMBASEU - Prefetchable Memory Base Upper Register (0x028) on page 9-19</a>
0x02C	DWord	Px_PMLIMITU	<a href="#">PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) on page 9-19</a>
0x030	Word	Px_IOBASEU	<a href="#">IOBASEU - I/O Base Upper Register (0x030) on page 9-19</a>
0x032	Word	Px_IOLIMITU	<a href="#">IOLIMITU - I/O Limit Upper Register (0x032) on page 9-20</a>
0x034	Byte	Px_CAPPTR	<a href="#">CAPPTR - Capabilities Pointer Register (0x034) on page 9-20</a>
0x038	DWord	Px_EROMBASE	<a href="#">EROMBASE - Expansion ROM Base Address Register (0x038) on page 9-20</a>

**Table 9.3 Downstream Ports 1 through 15 Configuration Space Registers (Part 1 of 4)**

## Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x03C	Byte	Px_INTRLINE	INTRLINE - Interrupt Line Register (0x03C) on page 9-20
0x03D	Byte	Px_INTRPIN	INTRPIN - Interrupt PIN Register (0x03D) on page 9-21
0x03E	Word	Px_BCTL	BCTL - Bridge Control Register (0x03E) on page 9-21
0x040	DWord	Px_PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 9-22
0x044	DWord	Px_PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 9-23
0x048	Word	Px_PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 9-24
0x04A	Word	Px_PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 9-25
0x04C	DWord	Px_PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 9-26
0x050	Word	Px_PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 9-27
0x052	Word	Px_PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 9-28
0x054	DWord	Px_PCIESCAP	PCIESCAP - PCI Express Slot Capabilities (0x054) on page 9-29
0x058	Word	Px_PCIESCTL	PCIESCTL - PCI Express Slot Control (0x058) on page 9-31
0x05A	Word	Px_PCIESSTS	PCIESSTS - PCI Express Slot Status (0x05A) on page 9-32
0x064	DWord	Px_PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 9-33
0x068	Word	Px_PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068) on page 9-33
0x06A	Word	Px_PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 9-34
0x06C	DWord	Px_PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 9-34
0x070	Word	Px_PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 9-34
0x072	Word	Px_PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 9-34
0x074	DWord	Px_PCIESCAP2	PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074) on page 9-34
0x078	Word	Px_PCIESCTL2	PCIESCTL2 - PCI Express Slot Control 2 (0x078) on page 9-34
0x07A	Word	Px_PCIESSTS2	PCIESSTS2 - PCI Express Slot Status 2 (0x07A) on page 9-35
0x0C0	DWord	Px_PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 9-35
0x0C4	DWord	Px_PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 9-36
0x0D0	DWord	Px_MSICAP	MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) on page 9-36
0x0D4	DWord	Px_MSIADDR	MSIADDR - Message Signaled Interrupt Address (0x0D4) on page 9-37
0x0D8	DWord	Px_MSUIADDR	MSUIADDR - Message Signaled Interrupt Upper Address (0x0D8) on page 9-37
0x0DC	DWord	Px_MSIMDATA	MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) on page 9-38
0x0F0	Dword	Px_SSIDSSVIDCAP	SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) on page 9-38
0x0F4	Dword	Px_SSIDSSVID	SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) on page 9-38

**Table 9.3 Downstream Ports 1 through 15 Configuration Space Registers (Part 2 of 4)**

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0F8	Word	Px_ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 9-38
0x0FC	Word	Px_ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 9-39
0x100	Dword	Px_AERCAP	AERCAP - AER Capabilities (0x100) on page 9-39
0x104	Dword	Px_AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 9-40
0x108	Dword	Px_AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 9-41
0x10C	Dword	Px_AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 9-42
0x110	Dword	Px_AERCES	AERCES - AER Correctable Error Status (0x110) on page 9-43
0x114	Dword	Px_AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 9-44
0x118	Dword	Px_AERCTL	AERCTL - AER Control (0x118) on page 9-45
0x11C	Dword	Px_AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 9-45
0x120	Dword	Px_AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 9-45
0x124	Dword	Px_AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 9-45
0x128	Dword	Px_AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 9-46
0x180	Dword	Px_SNUMCAP	SNUMCAP - Serial Number Capabilities (0x180) on page 9-46
0x184	Dword	Px_SNUMLDW	SNUMLDW - Serial Number Lower Doubleword (0x184) on page 9-46
0x188	Dword	Px_SNUMUDW	SNUMUDW - Serial Number Upper Doubleword (0x188) on page 9-46
0x200	DWord	Px_PCIEVCECAP	PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200) on page 9-47
0x204	DWord	Px_PVCCAP1	PVCCAP1 - Port VC Capability 1 (0x204) on page 9-47
0x208	DWord	Px_PVCCAP2	PVCCAP2 - Port VC Capability 2 (0x208) on page 9-47
0x20C	Word	Px_PVCCTL	PVCCTL - Port VC Control (0x20C) on page 9-48
0x20E	Word	Px_PVCSTS	PVCSTS - Port VC Status (0x20E) on page 9-48
0x210	DWord	Px_VCR0CAP	VCR0CAP - VC Resource 0 Capability (0x210) on page 9-49
0x214	DWord	Px_VCR0CTL	VCR0CTL - VC Resource 0 Control (0x214) on page 9-49
0x218	DWord	Px_VCR0STS	VCR0STS - VC Resource 0 Status (0x218) on page 9-50
0x21C	DWord	Px_VCR1CAP	VCR1CAP - VC Resource 1 Capability (0x21C) on page 9-50
0x220	DWord	Px_VCR1CTL	VCR1CTL - VC Resource 1 Control (0x220) on page 9-51
0x224	DWord	Px_VCR1STS	VCR1STS - VC Resource 1 Status (0x224) on page 9-52
0x280	Dword	Px_PWRBCAP	PWRBCAP - Power Budgeting Capabilities (0x280) on page 9-57
0x284	Dword	Px_PWRBDSEL	PWRBDSEL - Power Budgeting Data Select (0x284) on page 9-57
0x288	Dword	Px_PWRBD	PWRBD - Power Budgeting Data (0x288) on page 9-58

**Table 9.3 Downstream Ports 1 through 15 Configuration Space Registers (Part 3 of 4)**

# Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x28C	Dword	Px_PWRBPBC	<a href="#">PWRBPBC - Power Budgeting Power Budget Capability (0x28C) on page 9-58</a>
0x300	Dword	Px_PWRBDV0	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x304	Dword	Px_PWRBDV1	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x308	Dword	Px_PWRBDV2	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x30C	Dword	Px_PWRBDV3	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x310	Dword	Px_PWRBDV4	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x314	Dword	Px_PWRBDV5	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x318	Dword	Px_PWRBDV6	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x31C	Dword	Px_PWRBDV7	<a href="#">PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300) on page 9-58</a>
0x740	Dword	Px_SWPECTL	<a href="#">SWPECTL - Switch Parity Error Control (0x740) on page 9-76</a>
0x744	Dword	Px_SWPESTS	<a href="#">SWPESTS - Switch Parity Error Status (0x744) on page 9-76</a>
0x748	Dword	Px_SWPERCTL	<a href="#">SWPERCTL - Switch Parity Error Reporting Control (0x748) on page 9-77</a>
0x74C	Dword	Px_SWPECNT	<a href="#">SWPECNT - Switch Parity Error Count (0x74C) on page 9-77</a>
0x754	Dword	Px_SWTOSTS	<a href="#">SWTOSTS - Switch Time-Out Status (0x754) on page 9-77</a>
0x758	Dword	Px_SWTORCTL	<a href="#">SWTORCTL - Switch Time-Out Reporting Control (0x758) on page 9-78</a>
0x75C	Dword	Px_SWTOCNT	<a href="#">SWTOCNT - Switch Time-Out Count (0x75C) on page 9-79</a>

**Table 9.3 Downstream Ports 1 through 15 Configuration Space Registers (Part 4 of 4)**

## Register Definitions

### Type 1 Configuration Header Registers

#### VID - Vendor Identification Register (0x000)

Bit Field	Field Name	Type	Default Value	Description
15:0	VID	RO	0x111D	<b>Vendor Identification.</b> This field contains the 16-bit vendor ID value assigned to IDT. See section Vendor ID on page 1-4.

## Notes

### DID - Device Identification Register (0x002)

Bit Field	Field Name	Type	Default Value	Description
15:0	DID	RO	-	<b>Device Identification.</b> This field contains the 16-bit device ID assigned by IDT to this bridge. See section Device ID on page 1-4.

### PCICMD - PCI Command Register (0x004)

Bit Field	Field Name	Type	Default Value	Description
0	IOAE	RW	0x0	<b>I/O Access Enable.</b> When this bit is cleared, the bridge 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	<b>Memory Access Enable.</b> When this bit is cleared, the bridge 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	<b>Bus Master Enable.</b> When this bit is cleared, the bridge does not issue requests (e.g., memory, I/O and MSIs since they are in-band writes) on behalf of subordinate devices and responds to non-posted transactions 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	<b>Special Cycle Enable.</b> Not applicable.
4	MWI	RO	0x0	<b>Memory Write Invalidate.</b> Not applicable.
5	VGAS	RO	0x0	<b>VGA Palette Snoop.</b> Not applicable.
6	PERRE	RW	0x0	<b>Parity Error Enable.</b> The Master Data Parity Error bit is set in the PCI Status register (PCISTS) if this bit is set and the bridge receives a poisoned completion or generates a poisoned write. If this bit is cleared, then the Master Data Parity Error bit in the PCI Status register is never set. 0x0 -(disable) Disable Master Parity Error bit reporting. 0x1 -(enable) Enable Master Parity Error bit reporting.
7	ADSTEP	RO	0x0	<b>Address Data Stepping.</b> Not applicable.

## Notes

Bit Field	Field Name	Type	Default Value	Description
8	SERRE	RW	0x0	<b>SERR Enable.</b> 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_COR 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.
9	FB2B	RO	0x0	<b>Fast Back-to-Back Enable.</b> Not applicable.
10	INTXD	RW	0x0	<b>INTx Disable.</b> Controls the ability of the PCI-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	<b>INTx Status.</b> This bit is set when an INTx interrupt is pending from the device. INTx emulation interrupts forwarded by switch ports from devices downstream of the bridge are not reflected in this bit. For downstream ports, this bit is set if an interrupt has been "asserted" by the corresponding port's hot-plug controller. In the upstream port this field is always zero.
4	CAPL	RO	0x1	<b>Capabilities List.</b> This bit is hardwired to one to indicate that the bridge implements an extended capability list item.
5	C66MHZ	RO	0x0	<b>66 MHz Capable.</b> Not applicable.
6	Reserved	RO	0x0	Reserved field.
7	FB2B	RO	0x0	<b>Fast Back-to-Back (FB2B).</b> Not applicable.
8	MDPED	RW1C	0x0	<b>Master Data Parity Error Detected.</b> This bit is set when the PERRE bit is set in the PCI Command register and the bridge receives a poisoned completion or generates a poisoned write request on the primary side of the bridge. 0x0 - (noerror) no error. 0x1 - (error) Poisoned write request or completion received on primary side.
10:9	DEVT	RO	0x0	<b>DEVSEL# Timing.</b> Not applicable.
11	STAS	RO	0x0	<b>Signalled Target Abort.</b> Not applicable since a target abort is never signalled.

# Notes

Bit Field	Field Name	Type	Default Value	Description
12	RTAS	RO	0x0	<b>Received Target Abort.</b> Not applicable.
13	RMAS	RO	0x0	<b>Received Master Abort.</b> Not applicable.
14	SSE	RW1C	0x0	<b>Signalled System Error.</b> 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 signalled.
15	DPE	RW1C	0x0	<b>Detected Parity Error.</b> 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 register.

## RID - Revision Identification Register (0x008)

Bit Field	Field Name	Type	Default Value	Description
7:0	RID	RWL	-	<b>Revision ID.</b> This field contains the revision identification number for the device. See section Revision ID on page 1-4.

## CCODE - Class Code Register (0x009)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTF	RO	0x00	<b>Interface.</b> This value indicates that the device is a PCI-PCI bridge that does not support subtractive decode.
15:8	SUB	RO	0x04	<b>Sub Class Code.</b> This value indicates that the device is a PCI-PCI bridge.
23:16	BASE	RO	0x06	<b>Base Class Code.</b> This value indicates that the device is a bridge.

## CLS - Cache Line Size Register (0x00C)

Bit Field	Field Name	Type	Default Value	Description
7:0	CLS	RW	0x00	<b>Cache Line Size.</b> 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.

# Notes

## PLTIMER - Primary Latency Timer (0x00D)

Bit Field	Field Name	Type	Default Value	Description
7:0	PLTIMER	RO	0x00	<b>Primary Latency Timer.</b> Not applicable.

## HDR - Header Type Register (0x00E)

Bit Field	Field Name	Type	Default Value	Description
7:0	HDR	RO	0x01	<b>Header Type.</b> This value indicates a type 1 header with a single function bridge layout.

## BIST - Built-in Self Test Register (0x00F)

Bit Field	Field Name	Type	Default Value	Description
7:0	BIST	RO	0x0	<b>BIST.</b> This value indicates that the bridge does not implement BIST.

## BAR0 - Base Address Register 0 (0x010)

Bit Field	Field Name	Type	Default Value	Description
31:0	BAR	RO	0x0	<b>Base Address Register.</b> Not applicable.

## BAR1 - Base Address Register 1 (0x014)

Bit Field	Field Name	Type	Default Value	Description
31:0	BAR	RO	0x0	<b>Base Address Register.</b> Not applicable.

## PBUSN - Primary Bus Number Register (0x018)

Bit Field	Field Name	Type	Default Value	Description
7:0	PBUSN	RW	0x0	<b>Primary Bus Number.</b> 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 PES22H16 but is implemented as a read/write register for software compatibility

## Notes

### SBUSN - Secondary Bus Number Register (0x019)

Bit Field	Field Name	Type	Default Value	Description
7:0	SBUSN	RW	0x0	<b>Secondary Bus Number.</b> 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	<b>Subordinate Bus Number.</b> 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	<b>Secondary Latency Timer.</b> Not applicable.

### IOBASE - I/O Base Register (0x01C)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RWL	0x1	<b>I/O Capability.</b> 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	<b>I/O Base.</b> 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 4KB boundary that is below the primary interface of the bridge.

## Notes

### IOLIMIT - I/O Limit Register (0x01D)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RO	0x1	<b>I/O Capability.</b> 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	<b>I/O Limit.</b> 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 0xFFFF, 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	<b>Master Data Parity Error.</b> This bit is controlled by the Parity Error Response Enable bit in the Bridge Control register. If the Parity Response Enable bit is cleared, then this bit is never set. Otherwise, this bit is set if the bridge receives a poisoned completion or generates a poisoned write on the secondary side of the bridge.
10:9	DVSEL	RO	0x0	Not applicable.
11	STAS	RO	0x0	<b>Signalled Target Abort Status.</b> Not applicable.
12	RTAS	RO	0x0	<b>Received Target Abort Status.</b> Not applicable.
13	RMAS	RO	0x0	<b>Received Master Abort Status.</b> Not applicable.
14	RSE	RW1C	0x0	<b>Received System Error.</b> This bit is controlled by the SERR enable bit in the Bridge Control (BCTL) register. If the SERRE bit is cleared in BCTL, then this bit is never set. Otherwise, this bit is set if the secondary side of the bridge receives an ERR_FATAL or ERR_NONFATAL message.
15	DPE	RW1C	0x0	<b>Detected Parity Error.</b> 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

# Notes

## 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	<b>Memory Address Base.</b> 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	<b>Memory Address Limit.</b> 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.

## PMBASE - Prefetchable Memory Base Register (0x024)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RWL	0x1	<b>Prefetchable Memory Capability.</b> 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	<b>Prefetchable Memory Address Base.</b> 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.

## Notes

### PMLIMIT - Prefetchable Memory Limit Register (0x026)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RO	0x1	<b>Prefetchable Memory Capability.</b> 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	<b>Prefetchable Memory Address Limit.</b> 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_FF FF	<b>Prefetchable Memory Address Base Upper.</b> 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.

### PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C)

Bit Field	Field Name	Type	Default Value	Description
31:0	PMLIMITU	RW	0x0	<b>Prefetchable Memory Address Limit Upper.</b> 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	<b>I/O Address Base Upper.</b> 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.

# Notes

## IOLIMITU - I/O Limit Upper Register (0x032)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOLIMITU	RW	0x0	<b>Prefetchable IO Limit Upper.</b> 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	<b>Capabilities Pointer.</b> 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	<b>Expansion ROM Base Address.</b> The bridge does not implement an expansion ROM. Thus, this field is hardwired to zero.

## INTRLINE - Interrupt Line Register (0x03C)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRLINE	RW	0x0	<b>Interrupt Line.</b> This register communicates interrupt line routing information. Values in this register are programmed by system software and are system architecture specific. The bridge does not use the value in this register. Legacy interrupts may be implemented by downstream ports.

## Notes

### INTRPIN - Interrupt PIN Register (0x03D)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTRPIN	RWL	0x0	<p><b>Interrupt Pin.</b> Interrupt pin or legacy interrupt messages are not used by the bridge by default. However, they can be used for hot-plug by the downstream ports.</p> <p>This field should only be configured with values of 0x0 through 0x4.</p> <p>0x0 - (none) Bridge does not generate any interrupts.</p> <p>0x1 - (INTA) Bridge generates INTA interrupts.</p> <p>0x2 - (INTB) Bridge generates INTB interrupts.</p> <p>0x3 - (INTC) Bridge generates INTC interrupts.</p> <p>0x4 - (INTD) Bridge generates INTD interrupts.</p>

### BCTL - Bridge Control Register (0x03E)

Bit Field	Field Name	Type	Default Value	Description
0	PERRE	RW	0x0	<p><b>Parity Error Response Enable.</b> This bit controls the bridges response to poisoned TLPs on the secondary interface.</p> <p>0x0 - (ignore) Ignore poisoned TLPs (i.e., parity errors) on the secondary interface.</p> <p>0x1 - (report) Enable poisoned TLP (i.e., parity error) detection and reporting on the secondary interface of the bridge.</p>
1	SERRE	RW	0x0	<p><b>System Error Enable.</b> This bit controls forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL from the secondary interface of the bridge to the primary interface.</p> <p>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.</p> <p>0x0 - (ignore) Do not forward errors from the secondary to the primary interface.</p> <p>0x1 - (report) Enable forwarding of errors from secondary to the primary interface.</p>
2	ISAEN	RW	0x0	<p><b>ISA Enable.</b> This bit controls the routing of ISA I/O transactions.</p> <p>0 - (disable) Forward downstream all I/O addresses in the address range defined by the I/O base and I/O limit registers</p> <p>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)</p>
3	VGAEN	RW	0x0	<p><b>VGA Enable.</b> Controls the routing of processor-initiated transactions targeting VGA.</p> <p>0 - (block) Do not forward VGA compatible addresses from the primary interface to the secondary interface</p> <p>1 - (forward) Forward VGA compatible addresses from the primary to the secondary interface.</p>

# Notes

Bit Field	Field Name	Type	Default Value	Description
4	VGA16EN	RW	0x0	<b>VGA 16-bit Enable.</b> 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.
5	Reserved	RO	0x0	Reserved field.
6	SRESET	RW	0x0	<b>Secondary Bus Reset.</b> Setting this bit triggers a secondary bus reset. In the upstream port, setting this bit initiates a upstream secondary bus reset. In a downstream port, setting this bit initiates a secondary bus reset.
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	<b>Capability ID.</b> The value of 0x10 identifies this capability as a PCI Express capability structure.
15:8	NXTPTR	RWL	0xC0	<b>Next Pointer.</b> This field contains a pointer to the next capability structure.
19:16	VER	RWL	0x1	<b>PCI Express Capability Version.</b> This field indicates the PCI-SIG defined PCI Express capability structure version number.
23:20	TYPE	RO	Upstream: 0x5  Downstream: 0x6	<b>Port Type.</b> This field identifies the type of switch port (upstream or downstream).
24	SLOT	RWL	0x0	<b>Slot Implemented.</b> 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	<b>Interrupt Message Number.</b> The function is allocated none (upstream ports) or only one (downstream ports) MSI. Therefore, this field is set to zero.
30	TCS	RWL	0x1	<b>TCS Routing Supported.</b> The PES22H16 supports TCS routing.
31	Reserved	RO	0x0	Reserved field.

# Notes

## PCIEDCAP - PCI Express Device Capabilities (0x044)

Bit Field	Field Name	Type	Default Value	Description
2:0	MPAYLOAD	RWL	0x4	<b>Maximum Payload Size Supported.</b> This field indicates the maximum payload size that the device can support for TLPs. The default value corresponds to 2048 bytes.
4:3	PFS	RO	0x0	<b>Phantom Functions Supported.</b> This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers. The value is hardwired to 0x0 to indicate that no function number bits are used for phantom functions.
5	ETAG	RWL	0x1	<b>Extended Tag Field Support.</b> This field indicates the maximum supported size of the Tag field as a requester.
8:6	E0AL	RO	0x0	<b>Endpoint L0s Acceptable Latency.</b> 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 does not apply to a switch.
11:9	E1AL	RO	0x0	<b>Endpoint L1 Acceptable Latency.</b> 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 does not apply to a switch.
12	ABP	RO	0x0	<b>Attention Button Present.</b> In PCIe base 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 PCIe base 1.1
13	AIP	RO	0x0	<b>Attention Indicator Present.</b> In PCIe 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 PCIe base 1.1
14	PIP	RO	0x0	<b>Power Indicator Present.</b> In PCIe base 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 PCIe base 1.1
15	RBERR	RO	0x1	<b>Role Based Error Reporting.</b> This bit is set to indicate that the PES22H16 supports error reporting as defined in the PCIe base 1.1 specification.
17:16	Reserved	RO	0x0	Reserved field.

## Notes

Bit Field	Field Name	Type	Default Value	Description
25:18	CSPLV	RO	0x0	<b>Captured Slot Power Limit Value.</b> 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 the upstream port. This field is always zero in downstream ports.
27:26	CSPLS	RO	0x0	<b>Captured Slot Power Limit Scale.</b> 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 the upstream port. This field is always zero in downstream ports. 0 - (v1) 1.0x 1 - (v1p1) 0.1x 2 - (v0p01) 0.01x 3 - (v0p001x) 0.001x
31:28	Reserved	RO	0x0	Reserved field.

### PCIEDCTL - PCI Express Device Control (0x048)

Bit Field	Field Name	Type	Default Value	Description
0	CEREN	RW	0x0	<b>Correctable Error Reporting Enable.</b> This bit controls reporting of correctable errors.
1	NFEREN	RW	0x0	<b>Non-Fatal Error Reporting Enable.</b> This bit controls reporting of non-fatal errors.
2	FEREN	RW	0x0	<b>Fatal Error Reporting Enable.</b> This bit controls reporting of fatal errors.
3	URREN	RW	0x0	<b>Unsupported Request Reporting Enable.</b> This bit controls reporting of unsupported requests.
4	ERO	RO	0x0	<b>Enable Relaxed Ordering.</b> When set, this bit enables relaxed ordering. The switch never sets the relaxed ordering bit in transactions it initiates as a requester.
7:5	MPS	RW	0x0	<b>Max Payload Size.</b> This field sets maximum TLP payload size for the device. 0x0 - (\$128) 128 bytes max payload size 0x1 - (\$256) 256 bytes max payload size 0x2 - (\$512) 512 bytes max payload size 0x3 - (\$1024) 1024 bytes max payload size 0x4 - (\$2048) 2048 bytes max payload size 0x5 - reserved (treated as 128 bytes) 0x6 - reserved (treated as 128 bytes) 0x7 - reserved (treated as 128 bytes)

# Notes

Bit Field	Field Name	Type	Default Value	Description
8	ETFEN	RW	0x0	<b>Extended Tag Field Enable.</b> 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	<b>Phantom Function Enable.</b> The bridge does not support phantom function numbers. Therefore, this field is hardwired to zero.
10	AUXPMEN	RO	0x0	<b>Auxiliary Power PM Enable.</b> The device does not implement this capability.
11	ENS	RO	0x0	<b>Enable No Snoop.</b> The bridge 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	<b>Maximum Read Request Size.</b> The bridge does not generate transactions larger than 128 bytes 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	<b>Correctable Error Detected.</b> This bit indicates the status of correctable errors. Errors are logged in this register regardless of whether error reporting is enabled or not.
1	NFED	RW1C	0x0	<b>Non-Fatal Error Detected.</b> This bit indicates the status of correctable errors. Errors are logged in this register regardless of whether error reporting is enabled or not.
2	FED	RW1C	0x0	<b>Fatal Error Detected.</b> This bit indicates the status of Fatal errors. Errors are logged in this registers regardless of whether error reporting is enabled or not.
3	URD	RW1C	0x0	<b>Unsupported Request Detected.</b> This bit indicates the device received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not.
4	AUXPD	RO	0x0	<b>Aux Power Detected.</b> 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	<b>Transactions Pending.</b> The bridge 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	MAXLNK-SPD	RO	0x1	<b>Maximum Link Speed.</b> This field indicates the supported link speeds of the port. 1 - (gen1) 2.5 Gbps 2 - (gen2) 5 Gbps others - reserved
9:4	MAXLNK-WDTH	RWL	HWINIT	<b>Maximum Link Width.</b> 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. Setting this field to a invalid or reserved value results in x1 being used. The initial value of this field is: (x4) x4 link width.
11:10	ASPMS	RO	0x3	<b>Active State Power Management (ASPM) Support.</b> This field is hardwired to 0x3 to indicate L0s and L1 Support.
14:12	L0SEL	RWL	see text	<b>L0s Exit Latency.</b> This field indicates the L0s exit latency for the given PCI Express link. This field depends of whether a common or separate reference clock is used When separate clocks are used, 1 $\mu$ s to 2 $\mu$ s is reported with a read-only value of 0x5. When a common clock is used, 256 ns to 512 ns is reported with a read-only value of 0x3.
17:15	L1EL	RWL	0x2	<b>L1 Exit Latency.</b> This field indicates the L1 exit latency for the given PCI Express link. Transitioning from L1 to L0 always requires 2.3 $\mu$ S. Therefore, a value 2 $\mu$ s to less than 4 $\mu$ s is reported with a default value of 0x2.
18	CPM	RWL	0x0	<b>Clock Power Management.</b> This bit indicates if the component tolerates removal of the reference clock via the "CLKREQ#" mechanism. The PES22H16 does not support the removal of reference clocks.
19	SDERR	RWL	Upstream: 0x0  Downstream: 0x1	<b>Surprise Down Error Reporting.</b> The PES22H16 downstream ports support surprise down error reporting. This field does not apply to an upstream port and should be set to zero.
20	DLLLA	RWL	Upstream: 0x0  Downstream: 0x1	<b>Data Link Layer Link Active Reporting.</b> The PES22H16 downstream ports support the capability of reporting the DL_Active state of the data link control and management State machine. This field is not applicable for the upstream port and must be zero.

## Notes

Bit Field	Field Name	Type	Default Value	Description
21	LBN	RWL	0x0	<b>Link Bandwidth Notification Capability.</b> When set, this bit indicates support for the link bandwidth notification status and interrupt mechanisms. The PES22H16 downstream ports support the capability. This field is not applicable for the upstream port and must be zero.
23:22	Reserved	RO	0x0	Reserved field.
31:24	PORTNUM	RO	Port 0: 0x0 Port 1: 0x1 Port 2: 0x2 Port 3: 0x3 Port 4: 0x4 Port 5: 0x5 Port 6: 0x6 Port 7: 0x7 Port 8: 0x8 Port 9: 0x9 Port 10: 0xA Port 11: 0xB Port 12: 0xC Port 13: 0xD Port 14: 0xE Port 15: 0xF	<b>Port Number.</b> 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	<b>Active State Power Management (ASPM) Control.</b> This field controls the level of ASPM supported by the link. The initial value corresponds to disabled. The value contained in Serial EEPROM may override this default value 0x0 -(disabled) disabled 0x1 -(I0s) L0s enable entry 0x2 -(I1) L1 enable entry 0x3 -(I0sI1) L0s and L1 enable entry
2	Reserved	RO	0x0	Reserved field.
3	RCB	RO	0x0	<b>Read Completion Boundary.</b> This field is not applicable and is hardwired to zero.
4	LDIS	RW	0x0	<b>Link Disable.</b> When set in a downstream port, this bit disables the link. This field is not applicable in the upstream port.

## Notes

Bit Field	Field Name	Type	Default Value	Description
5	LRET	RW	0x0	<b>Link Retrain.</b> 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 is done, all modifications that affect link retraining are applied in the subsequent retraining. For compliance with the PCIe 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 hard-wired to zero. When the REGUNLOCK bit is set, writing a one to the LRET bit initiates link retraining on the upstream port.
6	CCLK	RW	0x0	<b>Common Clock Configuration.</b> When set, this bit indicates that this component and the component at the opposite end of the link are operating with a distributed common reference clock.
7	ESYNC	RW	0x0	<b>Extended Sync.</b> When set this bit forces transmission of additional ordered sets when exiting the L0s state and when in the recovery state.
8	CLKP-WRMGT	RO	0x0	<b>Enable Clock Power Management.</b> The PES22H16 does not support this feature.
15:9	Reserved	RO	0x0	Reserved Field

### PCIELSTS - PCI Express Link Status (0x052)

Bit Field	Field Name	Type	Default Value	Description
3:0	LS	RO	0x1	<b>Link Speed.</b> This field indicates the current link speeds of the port. 1 -(gen1) 2.5 Gbps 2 -(gen2) 5 Gbps others-reserved
9:4	LW	RO	HWINIT	<b>Link Width.</b> This field indicates the negotiated width of the link.
10	TERR	RO	0x0	<b>Training Error.</b> In PCIe base 1.0a when set, this bit indicates that a link training error has occurred. The value of this field is undefined in PCIe base 1.1
11	LTRAIN	RO	0x0	<b>Link Training.</b> When set, this bit indicates that link training is in progress.

## Notes

Bit Field	Field Name	Type	Default Value	Description
12	SCLK	RWL	HWINIT	<b>Slot Clock Configuration.</b> When set, this bit indicates that the component uses the same physical reference clock that the platform provides. The initial value of this field is the state of the CCLKUS signal for the upstream port and the CCLKDS signal for downstream ports. The serial EEPROM may override these default values.
13	DLLA	RO	0x0	<b>Data Link Layer Link Active.</b> This bit indicates the status for the data link control and management state machine. This bit is always zero if the DLLA bit in the PCIELCAP register is not set. 0x0 -(notactive) Data link layer not active state 0x1 -(active) Data link layer active state
15:14	Reserved	RW1C	0x0	Reserved field.

### PCIESCAP - PCI Express Slot Capabilities (0x054)

Bit Field	Field Name	Type	Default Value	Description
0	ABP	RWL	0x0	<b>Attention Button Present.</b> 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	<b>Power Control Present.</b> 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	<b>MRL Sensor Present.</b> 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	<b>Attention Indicator Present.</b> 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	<b>Power Indicator Present.</b> 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	<b>Hot-Plug Surprise.</b> 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	<b>Hot-Plug Capable.</b> 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.

## Notes

Bit Field	Field Name	Type	Default Value	Description
14:7	SPLV	RW	0x0	<b>Slot Power Limit Value.</b> 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 non DL_Up 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.
16:15	SPLS	RW	0x0	<b>Slot Power Limit Scale.</b> 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 non DL_Up 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	<b>Electromechanical Interlock Present.</b> 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	<b>No Command Completed Support.</b> 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	<b>Physical Slot Number.</b> 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	0x0	<b>Attention Button Pressed Enable.</b> 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.
1	PFDE	RW	0x0	<b>Power Fault Detected Enable.</b> 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.
2	MRLSCE	RW	0x0	<b>MRL Sensor Change Enable.</b> 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.
3	PDCE	RW	0x0	<b>Presence Detected Changed Enable.</b> This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a presence detect change event.
4	CCIE	RW	0x0	<b>Command Complete Interrupt Enable.</b> This bit when set enables the generation of a Hot-Plug interrupt when a command is completed by the Hot-Plug Controller. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
5	HPIE	RW	0x0	<b>Hot-Plug Interrupt Enable.</b> 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.
7:6	AIC	RW	0x3	<b>Attention Indicator Control.</b> When read, this register returns the current state of the Attention Indicator. Writing to this register sets the indicator and causes the port to send the appropriate ATTENTION_INDICATOR_* message. 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 PCI-ESCAP register. 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	0x1	<b>Power Indicator Control.</b> When read, this register returns the current state of the Power Indicator. Writing to this register sets the indicator and causes the port to send the appropriate POWER_INDICATOR_* message. 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 PCI-ESCAP register. 0x0 -(reserved) Reserved 0x1 -(on) On 0x2 -(blink) Blink 0x3 -(off) Off This field has no effect on the upstream port.
10	PCC	RW	0x0	<b>Power Controller Control.</b> 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. 0x0 -(on) Power on 0x1 -(off) Power off
11	EIC	RW	0x0	<b>Electromechanical Interlock Control.</b> 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	DLLASCE	RW	0x0	<b>Data Link Layer Link Active State Change Enable.</b> This bit when set enables generation of a Hot-Plug interrupt or wake-up even on a data link layer active field state change.
15:13	Reserved	RO	0x0	Reserved field.

### PCIESSTS - PCI Express Slot Status (0x05A)

Bit Field	Field Name	Type	Default Value	Description
0	ABP	RW1C	0x0	<b>Attention Button Pressed.</b> Set when the attention button is pressed.
1	PFD	RW1C	0x0	<b>Power Fault Detected.</b> Set when the Power Controller detects a power fault.
2	MRLSC	RW1C	0x0	<b>MRL Sensor Changed.</b> Set when an MRL Sensor state change is detected.
3	PDC	RW1C	0x0	<b>Presence Detected Changed.</b> Set when a Presence Detected change is detected.

# Notes

Bit Field	Field Name	Type	Default Value	Description
4	CC	RW1C	0x0	<b>Command Completed.</b> 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	<b>MRL Sensor State.</b> This field enclosed the current state of the MRL sensor. 0x0 -(closed) MRL closed 0x1 -(open) MRL open
6	PDS	RO	0x1	<b>Presence Detect State.</b> This bit indicates the presence of a card in the slot corresponding to the port and reflects the state of the Presence Detect status. 0x0 -(empty) Slot empty 0x1 -(present) Card present
7	EIS	RO	0x0	<b>Electromechanical Interlock Status.</b> When an electromechanical interlock is implemented, this bit indicates the current status of the interlock. 0x0 -(disengaged) Electromechanical interlock disengaged 0x1 -(engaged) Electromechanical interlock engaged
8	DLLASC	RW1C	0x0	<b>Data Link Layer Link Active State Change.</b> This bit is set when the state of the data link layer active field in the link status register changes state. 0x0 -(nochange) No DLLLA state change 0x1 -(changed) DLLLA state change
15:9	Reserved	RO	0x0	Reserved field.

## PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064)

Bit Field	Field Name	Type	Default Value	Description
31:0	Reserved	RO	0x0	Reserved field.

## PCIEDCTL2 - PCI Express Device Control 2 (0x068)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.

# 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
15:0	Reserved	RO	0x0	Reserved field.

## PCIELSTS2 - PCI Express Link Status 2 (0x072)

Bit Field	Field Name	Type	Default Value	Description
15:0	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.

## Power Management Capability Structure

### PMCAP - PCI Power Management Capabilities (0x0C0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x1	<b>Capability ID.</b> The value of 0x1 identifies this capability as a PCI power management capability structure.
15:8	NXTPTR	RWL	Upstream: 0x0 Downstream: 0xD0	<b>Next Pointer.</b> This field contains a pointer to the next capability structure. For the upstream port, the value of this field is 0x0 indicating that it is the last capability. For downstream ports, this field is 0xD0 and points to the MSI capability structure.
18:16	VER	RO	0x3	<b>Power Management Capability Version.</b> This field indicates compliance with version two of the specification. Complies with version the PCI Bus Power Management Interface Specification, Revision 1.2.
19	PMECLK	RO	0x0	<b>PME Clock.</b> Does not apply to PCI Express.
20	Reserved	RO	0x0	Reserved field.
21	DEVSP	RWL	0x0	<b>Device Specific Initialization.</b> The value of zero indicates that no device specific initialization is required.
24:22	AUXI	RO	0x0	<b>AUX Current.</b> not used
25	D1	RO	0x0	<b>D1 Support.</b> This field indicates that the PES22H16 does not support D1.
26	D2	RO	0x0	<b>D2 Support.</b> This field indicates that the PES22H16 does not support D2.
31:27	PME	RWL	0b11001	<b>PME Support.</b> This field indicates the power states in which the port 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 <sub>cold</sub> . 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 device 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	<b>Power State.</b> This field is used to determine the current power state and to set a new power state. 0x0 - (d0) D0 state 0x1 - (d1) D1 state (not supported by the PES22H16 and reserved) 0x2- (d2) D2 state (not supported by the PES22H16 and reserved) 0x3 - (d3) D3 <sub>hot</sub> state
2	Reserved	RO	0x0	Reserved field.
3	NOSOFTRST	RWL	0x1	<b>No Soft Reset.</b> This bit indicates if the configuration context is preserved by the bridge when the device transitions from a D3 <sub>hot</sub> to D0 power management state. 0x0 - (reset) State reset 0x0 - (preserved) State preserved
7:4	Reserved	RO	0x0	Reserved field.
8	PMEE	RW Sticky	0x0	<b>PME Enable.</b> When this bit is set, PME message generation is enabled for the port. If a hot-plug wakeup event is desired when exiting the D3 <sub>cold</sub> 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	<b>Data Select.</b> The optional data register is not implemented.
14:13	DSCALE	RO	0x0	<b>Data Scale.</b> The optional data register is not implemented.
15	PMES	RW1C Sticky	0x0	<b>PME Status.</b> This bit is set if a PME is generated by the port even if the PMEE bit is cleared. This bit is not set when the bridge is propagating a PME message but the port 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	<b>B2/B3 Support.</b> Does not apply to PCI Express.
23	BPCCE	RO	0x0	<b>Bus Power/Clock Control Enable.</b> Does not apply to PCI Express.
31:24	DATA	RO	0x0	<b>Data.</b> 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	<b>Capability ID.</b> The value of 0x5 identifies this capability as a MSI capability structure.

## Notes

Bit Field	Field Name	Type	Default Value	Description
15:8	NXTPTR	RWL	0x0	<b>Next Pointer.</b> This field contains a pointer to the next capability structure. This field is set to 0x0 indicating that it is the last capability.
16	EN	RW	0x0	<b>Enable.</b> This bit enables MSI. 0x0 -(disable) disabled 0x1 -(enable) enabled
19:17	MMC	RO	0x0	<b>Multiple Message Capable.</b> This field contains the number of requested messages.
22:20	MME	RW	0x0	<b>Multiple Message Enable.</b> Hardwired to one message.
23	A64	RO	0x1	<b>64-bit Address Capable.</b> 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	<b>Message Address.</b> This field specifies the lower portion of the DWORD address of the MSI memory write transaction. The PES22H16 assumes that all downstream port generated MSIs are targeted to the root and routes these transactions to the upstream port. Configuring the address contained in a downstream port's MSIADDR and MSIADDRU registers to an address that does not route to the upstream port and generating an MSI produces undefined results.

### MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8)

Bit Field	Field Name	Type	Default Value	Description
31:0	UADDR	RW	0x0	<b>Upper Message Address.</b> 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 field is used. The PES22H16 assumes that all downstream port generated MSIs are targeted to the root and routes these transactions to the upstream port. Configuring the address contained in a downstream port's MSIADDR and MSIADDRU registers to an address that does not route to the upstream port and generating an MSI produces undefined results.

# Notes

## MSIMDATA - Message Signaled Interrupt Message Data (0x0DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	MDATA	RW	0x0	<b>Message Data.</b> This field contains the lower 16-bits of data that are written when a MSI is signalled.
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	<b>Capability ID.</b> The value of 0xD identifies this capability as a SSID/SSVID capability structure.
15:8	NXTPTR	RWL	0x00	<b>Next Pointer.</b> This field contains a pointer to the next capability structure.
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	<b>SubSystem Vendor ID.</b> 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	<b>Subsystem ID.</b> 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	<b>Register Number.</b> This field selects the configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 1.0a
11:8	EREG	RW	0x0	<b>Extended Register Number.</b> This field selects the extended configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 1.0a
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	<b>Configuration Data.</b> 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. When the ECFGADDR register points to the ECFGDATA register, then reads from ECFGDATA return zero and writes are ignored. When the ECFGADDR register points to itself, writes to the ECFGDATA register modify the contents of the ECFGADDR register. SMBus reads of this field return a value of zero and SMBus writes have no effect.

### Advanced Error Reporting (AER) Enhanced Capability

#### AERCAP - AER Capabilities (0x100)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x1	<b>Capability ID.</b> The value of 0x1 indicates an advanced error reporting capability structure.
19:16	CAPVER	RO	0x1	<b>Capability Version.</b> The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x180	<b>Next Pointer.</b>

# Notes

## AERUES - AER Uncorrectable Error Status (0x104)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW1C	0x0 Sticky	<b>Undefined.</b> This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW1C	0x0 Sticky	<b>Data Link Protocol Error Status.</b> This bit is set when a data link layer protocol error is detected.
5	SDOENERR	RW1C	0x0 Sticky	<b>Surprise Down Error Status.</b> This bit is set when a data link layer protocol error is detected. This bit is always zero if the SDERR bit in the PCIELCAP register is not set.
11:6	Reserved	RO	0x0 Sticky	Reserved field.
12	POISONED	RW1C	0x0 Sticky	<b>Poisoned TLP Status.</b> This bit is set when a poisoned TLP is detected.
13	FCPERR	RW1C	0x0 Sticky	<b>Flow Control Protocol Error Status.</b> This bit is set when a flow control protocol error is detected.
14	COMPTO	RO	0x0	<b>Completion Time-out Status.</b> A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	<b>Completer Abort Status.</b> The PES22H16 never responds to a non-posted request with a completer abort.
16	UECOMP	RW1C	0x0 Sticky	<b>Unexpected Completion Status.</b> This bit is set when an unexpected completion is detected.
17	RCVOVR	RW1C	0x0 Sticky	<b>Receiver Overflow Status.</b> This bit is set when a receiver overflow is detected.
18	MAL-FORMED	RW1C	0x0 Sticky	<b>Malformed TLP Status.</b> This bit is set when a malformed TLP is detected.
19	ECRC	RW1C	0x0 Sticky	<b>ECRC Status.</b> This bit is set when an ECRC error is detected.
20	UR	RW1C	0x0 Sticky	<b>UR Status.</b> This bit is set when an unsupported request is detected.
31:21	Reserved	RO	0x0	Reserved field.

# Notes

## AERUEM - AER Uncorrectable Error Mask (0x108)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	<b>Undefined.</b> This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x0 Sticky	<b>Data Link Protocol Error Mask.</b> 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 and an error is not reported to the root complex.
5	SDOENERR	RW	0x0 Sticky	<b>Surprise Down Error Mask.</b> 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 and an error is not reported to the root complex.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	<b>Poisoned TLP Mask.</b> 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 and an error is not reported to the root complex.
13	FCPERR	RW	0x0 Sticky	<b>Flow Control Protocol Error Mask.</b> 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 and an error is not reported to the root complex.
14	COMPTO	RO	0x0	<b>Completion Time-out Mask.</b> A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	<b>Completer Abort Mask.</b> The PES22H16 never responds to a non-posted request with a completer abort.
16	UECOMP	RW	0x0 Sticky	<b>Unexpected Completion Mask.</b> 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 and an error is not reported to the root complex.
17	RCVOVR	RW	0x0 Sticky	<b>Receiver Overflow Mask.</b> 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 and an error is not reported to the root complex.
18	MAL-FORMED	RW	0x0 Sticky	<b>Malformed TLP Mask.</b> 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 and an error is not reported to the root complex.

# Notes

Bit Field	Field Name	Type	Default Value	Description
19	ECRC	RW	0x0 Sticky	<b>ECRC Mask.</b> 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 and an error is not reported to the root complex.
20	UR	RW	0x0 Sticky	<b>UR Mask.</b> 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 and an error is not reported to the root complex.
31:21	Reserved	RO	0x0	Reserved field.

## AERUESV - AER Uncorrectable Error Severity (0x10C)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	<b>Undefined.</b> This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x1 Sticky	<b>Data Link Protocol Error Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	RW	0x1 Sticky	<b>Surprise Down Error Status.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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.
11:6	Reserved	RO	0x0	Reserved field.
12	POISONED	RW	0x0 Sticky	<b>Poisoned TLP Status Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	RW	0x1 Sticky	<b>Flow Control Protocol Error Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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.
14	COMPTO	RO	0x0	<b>Completion Time-out Status.</b> A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	<b>Completer Abort Status.</b> The PES22H16 never responds to a non-posted request with a completer abort.

## Notes

Bit Field	Field Name	Type	Default Value	Description
16	UECOMP	RW	0x0 Sticky	<b>Unexpected Completion Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	<b>Receiver Overflow Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	MAL-FORMED	RW	0x1 Sticky	<b>Malformed TLP Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	<b>ECRC Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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	<b>UR Severity.</b> If the corresponding event is not masked in the AERUEM register, then when the event occurs, 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.
31:21	Reserved	RO	0x0	Reserved field.

**AERCES - AER Correctable Error Status (0x110)**

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW1C	0x0 Sticky	<b>Receiver Error Status.</b> This bit is set when the physical layer detects a receiver error.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW1C	0x0 Sticky	<b>Bad TLP Status.</b> This bit is set when a bad TLP is detected.
7	BADDLLP	RW1C	0x0 Sticky	<b>Bad DLLP Status.</b> This bit is set when a bad DLLP is detected.
8	RPLYROVR	RW1C	0x0 Sticky	<b>Replay Number Rollover Status.</b> 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.

# Notes

Bit Field	Field Name	Type	Default Value	Description
12	RPLYTO	RW1C	0x0 Sticky	<b>Replay Timer Time-Out Status.</b> This bit is set when the replay timer in the data link layer times out.
13	ADVISO-RYNF	RW1C	0x0 Sticky	<b>Advisory Non-Fatal Error Status.</b> This bit is set when an advisory non-fatal error is detected as described in Section 6.2.3.2.4 of the PCIe base 1.1 specification.
31:14	Reserved	RO	0x0	Reserved field.

## AERCEM - AER Correctable Error Mask (0x114)

Bit Field	Field Name	Type	Default Value	Description
0	RCVERR	RW	0x0 Sticky	<b>Receiver Error Mask.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 Sticky	<b>Bad TLP Mask.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
7	BADDLLP	RW	0x0 Sticky	<b>Bad DLLP Mask.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
8	RPLYROVR	RW	0x0 Sticky	<b>Replay Number Rollover Mask.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 Sticky	<b>Replay Timer Time-Out Mask.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
13	ADVISO-RYNF	RW	0x1 Sticky	<b>Advisory Non-Fatal Error Status.</b> When this bit is set, the corresponding bit in the AERCEM register is masked. When a bit is masked in the AERCEM register, the corresponding event is not reported to the root complex.
31:14	Reserved	RO	0x0	Reserved field.

# Notes

## AERCTL - AER Control (0x118)

Bit Field	Field Name	Type	Default Value	Description
4:0	FEPTR	RO	0x0 Sticky	<b>First Error Pointer.</b> This field contains a pointer to the bit in the AERUES register that resulted in the first reported error.
5	ECRCGC	RWL	0x1	<b>ECRC Generation Capable.</b> This bit indicates if the device is capable of generating ECRC.
6	ECRCGE	RW	0x0 Sticky	<b>ECRC Generation Enable.</b> When this bit is set, ECRC generation is enabled.
7	ECRCCC	RWL	0x1	<b>ECRC Check Capable.</b> This bit indicates if the device is capable of checking ECRC.
8	ECRCCE	RW	0x0 Sticky	<b>ECRC Check Enable.</b> When set, this bit enables ECRC checking.
31:9	Reserved	RO	0x0	Reserved field.

## AERHL1DW - AER Header Log 1st Doubleword (0x11C)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RO	0x0 Sticky	<b>Header Log.</b> 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	RO	0x0 Sticky	<b>Header Log.</b> 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	RO	0x0 Sticky	<b>Header Log.</b> This field contains the 3rd doubleword of the TLP header that resulted in the first reported uncorrectable error.

# Notes

## AERHL4DW - AER Header Log 4th Doubleword (0x128)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RO	0x0 Sticky	<b>Header Log.</b> This field contains the 4th doubleword of the TLP header that resulted in the first reported uncorrectable error.

## Device Serial Number Enhanced Capability

### SNUMCAP - Serial Number Capabilities (0x180)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x3	<b>Capability ID.</b> The value of 0x3 indicates a device serial number capability structure.
19:16	CAPVER	RO	0x1	<b>Capability Version.</b> The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x200	<b>Next Pointer.</b>

### SNUMLDW - Serial Number Lower Doubleword (0x184)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 Sticky	<b>Lower 32-bits of Device Serial Number.</b> This field contains the lower 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device.

### SNUMUDW - Serial Number Upper Doubleword (0x188)

Bit Field	Field Name	Type	Default Value	Description
31:0	SNUM	RWL	0x0 Sticky	<b>Upper 32-bits of Device Serial Number.</b> This field contains the upper 32-bits of the IEEE defined 64-bit extended unique identifier (EUI-64) assigned to the device.

## Notes

### PCI Express Virtual Channel Capability

#### PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x2	<b>Capability ID.</b> The value of 0x2. indicates a virtual channel capability structure.
19:16	CAPVER	RO	0x1	<b>Capability Version.</b> The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x0	<b>Next Pointer.</b> The value of 0x0 indicates that there are no extended capabilities.

#### PVCCAP1 - Port VC Capability 1 (0x204)

Bit Field	Field Name	Type	Default Value	Description
2:0	EVCCNT	RO	0x01	<b>Extended VC Count.</b> The value 0x1 indicates that the default VC (VC0) and VC1 are implemented. The default value of this field is 0x1.
3	Reserved	RO	0x0	Reserved field.
6:4	LPEVCCNT	RO	0x0	<b>Low Priority Extended VC Count.</b> The value of 0x0 indicates that all VCs are in the upper priority group and are arbitrated based on strict priority (VC7 highest, VC0 lowest).
7	Reserved	RO	0x0	Reserved field.
9:8	REFCLK	RO	0x0	<b>Reference Clock.</b> Time-based WRR is not implemented.
11:10	PATBSIZ	RO	0x2	<b>Port Arbitration Table Entry Size.</b> This field indicates the size of the port arbitration table in the device. The field is set to 0x2 to indicate a table with 4-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.

#### PVCCAP2 - Port VC Capability 2 (0x208)

Bit Field	Field Name	Type	Default Value	Description
7:0	VCARBCAP	RO	0x0	<b>VC Arbitration Capability.</b> This field indicates the type of VC arbitration that is supported by the port for the low priority VC group. This field is valid for all ports that report a low priority extended VC count greater than zero. Each bit in this field corresponds to a VC arbitration capability. bit 0 - hardware fixed arbitration (i.e., round robin) bits 1 through 7 - reserved

# Notes

Bit Field	Field Name	Type	Default Value	Description
23:8	Reserved	RO	0x0	Reserved field.
31:24	VCATBL-OFF	RO	0x0	<b>VC Arbitration Table Offset.</b> This field contains the offset of the VC arbitration table from the base address of the Virtual Channel Capability structure in double quad words (16 bytes). The value of zero indicates that the VC arbitration table is not present.

## PVCCTL - Port VC Control (0x20C)

Bit Field	Field Name	Type	Default Value	Description
0	LVCAT	RO	0x0	<b>Load VC Arbitration Table.</b> This bit, when set, updates the VC arbitration logic from the VC Arbitration Table for the VC resource. Since the device does not implement a VC arbitration table, this field has no functional effect. This bit always returns 0 when read.
3:1	VCARBSSEL	RW	0x0	<b>VC Arbitration Select.</b> This field configures the VC arbitration by selecting on of the supported arbitration schemes indicated by the VC arbitration capability field (i.e., the VCARBCAP field in the PVCCAP2 register).
15:4	Reserved	RO	0x0	Reserved field.

## PVCSTS - Port VC Status (0x20E)

Bit Field	Field Name	Type	Default Value	Description
0	VCATS	RO	0x0	<b>VC Arbitration Table Status.</b> This bit indicates the coherency status of the VC arbitration table.
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	RWL	0x1	<b>Port Arbitration Capability.</b> This field indicates the type of port arbitration supported by the VC. Each bit corresponds to a Port Arbitration capability. When more than one arbitration scheme is supported, multiple bits may be set. All ports support hardware fixed round robin. bit 0 - hardware fixed round robin bits 1 through 5 - reserved
13:8	Reserved	RO	0x0	Reserved field.
14	APS	RO	0x0	<b>Advanced Packet Switching.</b> Not supported.
15	RJST	RO	0x0	<b>Reject Snoop Transactions.</b> No supported for switch ports.
22:16	MAXTS	RO	0x0	<b>Maximum Time Slots.</b> Since this VC does not support time-based WRR, this field is not valid.
23	Reserved	RO	0x0	Reserved field.
31:24	PATBLOFF	RWL	0x0	<b>Port Arbitration Table Offset.</b> This field contains the offset of the port arbitration table from the base address of the Virtual Channel Capability structure in double quad words (16 bytes).

### 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	<b>TC/VC Map.</b> 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	RW	0x0	<b>Load Port Arbitration Table.</b> This bit, when set, updates the Port Arbitration logic from the Port Arbitration Table for the VC resource. In addition, this field is only valid when the Port Arbitration Table is used by the selected Port Arbitration scheme (that is indicated by a set bit in the Port Arbitration Capability field selected by Port Arbitration Select). Software sets this bit to signal hardware to update Port Arbitration logic with new values stored in Port Arbitration Table; clearing this bit has no effect. Software uses the Port Arbitration Table Status bit to confirm whether the new values of Port Arbitration Table are completely latched by the arbitration logic. This bit always returns 0 when read.

## Notes

Bit Field	Field Name	Type	Default Value	Description
19:17	PARBSEL	RW	0x0	<b>Port Arbitration Select.</b> This field configures the VC resource to provide a particular Port Arbitration service. The permissible values of this field is a number that corresponds to one of the asserted bits in the Port Arbitration Capability field of the VC resource.
23:20	Reserved	RO	0x0	Reserved field.
26:24	VCID	RO	0x0	<b>VC ID.</b> This field assigns a VC ID to the VC resource. For VC0 this field is hardwired to zero.
30:27	Reserved	RO	0x0	Reserved field.
31	VCEN	RO	0x1	<b>VC Enable.</b> This field, when set, enables a virtual channel. For VC0, this field is hardwired to one (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	<b>Port Arbitration Table Status.</b> This bit indicates the coherency status of the port arbitration table associated with the VC resource and is valid only when the port arbitration table is used by the selected arbitration algorithm. This bit is set when any entry of the port arbitration table is written by software and remains set until hardware finishes loading the value after software sets the LPAT field in the VCR0CTL register.
17	VCNEG	RO	0x0	<b>VC Negotiation Pending.</b> This field indicates whether the virtual channel negotiation is in a pending state. Since VC0 is always enabled and this bit can only be read via PCIe using VC0, this bit is always zero for VC0. This bit cannot be used by an SMBus device to determine the VC negotiation status of VC0.
30:18	Reserved	RO	0x0	Reserved field.

### VCR1CAP - VC Resource 1 Capability (0x21C)

Bit Field	Field Name	Type	Default Value	Description
7:0	PARBC	RWL	0x1	<b>Port Arbitration Capability.</b> This field indicates the type of port arbitration supported by the VC. Each bit corresponds to a Port Arbitration capability. When more than one arbitration scheme is supported, multiple bits may be set. All ports support hardware fixed round robin. bit 0 - hardware fixed round robin bits 1 through 5 - reserved
13:8	Reserved	RO	0x0	Reserved field.
14	APS	RO	0x0	<b>Advanced Packet Switching.</b> Not supported.

# Notes

Bit Field	Field Name	Type	Default Value	Description
15	RJST	RO	0x0	<b>Reject Snoop Transactions.</b> No supported for switch ports.
22:16	MAXTS	RO	0x0	<b>Maximum Time Slots.</b> Since this VC does not support time-based WRR, this field is not valid.
23	Reserved	RO	0x0	Reserved field.
31:24	PATBLOFF	RWL	0x0	<b>Port Arbitration Table Offset.</b> This field contains the offset of the port arbitration table from the base address of the Virtual Channel Capability structure in double quad words (16 bytes).

## VCR1CTL - VC Resource 1 Control (0x220)

Bit Field	Field Name	Type	Default Value	Description
7:0	TCVCMAP	bit 0 RO bits 1 through 7 RW	0x0	<b>TC/VC Map.</b> 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	RW	0x0	<b>Load Port Arbitration Table.</b> This bit, when set, updates the Port Arbitration logic from the Port Arbitration Table for the VC resource. In addition, this field is only valid when the Port Arbitration Table is used by the selected Port Arbitration scheme (that is indicated by a set bit in the Port Arbitration Capability field selected by Port Arbitration Select). Software sets this bit to signal hardware to update Port Arbitration logic with new values stored in Port Arbitration Table; clearing this bit has no effect. Software uses the Port Arbitration Table Status bit to confirm whether the new values of Port Arbitration Table are completely latched by the arbitration logic. This bit always returns 0 when read.
19:17	PARBSEL	RW	0x0	<b>Port Arbitration Select.</b> This field configures the VC resource to provide a particular Port Arbitration service. The permissible values of this field is a number that corresponds to one of the asserted bits in the Port Arbitration Capability field of the VC resource.
23:20	Reserved	RO	0x0	Reserved field.
26:24	VCID	RW	0x1	<b>VC ID.</b> This field assigns a VC ID to the VC resource.
30:27	Reserved	RO	0x0	Reserved field.
31	VCEN	RW	0x0	<b>VC Enable.</b> This field, when set, enables the corresponding virtual channel.

## Notes

### VCR1STS - VC Resource 1 Status (0x224)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.
16	PATS	RO	0x0	<b>Port Arbitration Table Status.</b> This bit indicates the coherency status of the port arbitration table associated with the VC resource and is valid only when the port arbitration table is used by the selected arbitration algorithm. This bit is set when any entry of the port arbitration table is written by software and remains set until hardware finishes loading the value after software sets the LPAT field in the VCR1CTL register.
17	VCNEG	RO	0x0	<b>VC Negotiation Pending.</b> This field indicates whether the virtual channel negotiation is in a pending state.
30:18	Reserved	RO	0x0	Reserved field.

### VCR0TBL0 - VC Resource 0 Arbitration Table Entry 0 (0x230)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE0	RW	0x0	<b>Phase 0.</b> This field contains the port ID for the corresponding port arbitration period. Selecting an invalid port ID results in the entry being skipped without delay. The port arbitration behavior when this field contains an illegal value (i.e., reserved or the egress port ID) is undefined. 0x0 -(port_0) Port 0 (upstream port) 0x1 -(port_1) Port 1 0x2 -(port_2) Port 2 0x3 -(port_3) Port 3 0x4 -(port_4) Port 4 0x5 -(port_5) Port 5 0x6 -(port_6) Port 6 0x7 -(port_7) Port 7 0x8 -(port_8) Port 8 0x9 -(port_9) Port 9 0x10 -(port_10) Port 10 0x11 -(port_11) Port 11 0x12 -(port_12) Port 12 0x13 -(port_13) Port 13 0x14 -(port_14) Port 14 0x15 -(port_15) Port 15
7:4	PHASE1	RW	0x0	<b>Phase 1.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE2	RW	0x1	<b>Phase 2.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE3	RW	0x1	<b>Phase 3.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE4	RW	0x2	<b>Phase 4.</b> This field contains the port ID for the corresponding port arbitration period.

# Notes

Bit Field	Field Name	Type	Default Value	Description
23:20	PHASE5	RW	0x2	<b>Phase 5.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE6	RW	0x3	<b>Phase 6.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE7	RW	0x3	<b>Phase 7.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR0TBL1 - VC Resource 0 Arbitration Table Entry 1 (0x234)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE8	RW	0x4	<b>Phase 8.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE9	RW	0x4	<b>Phase 9.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE10	RW	0x5	<b>Phase 10.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE11	RW	0x5	<b>Phase 11.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE12	RW	0x6	<b>Phase 12.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE13	RW	0x6	<b>Phase 13.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE14	RW	0x7	<b>Phase 14.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE15	RW	0x7	<b>Phase 15.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR0TBL2 - VC Resource 0 Arbitration Table Entry 2 (0x238)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE16	RW	0x8	<b>Phase 16.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE17	RW	0x8	<b>Phase 17.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE18	RW	0x9	<b>Phase 18.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE19	RW	0x9	<b>Phase 19.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE20	RW	0x10	<b>Phase 20.</b> This field contains the port ID for the corresponding port arbitration period.

# Notes

Bit Field	Field Name	Type	Default Value	Description
23:20	PHASE21	RW	0x10	<b>Phase 21.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE22	RW	0x11	<b>Phase 22.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE23	RW	0x11	<b>Phase 23.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR0TBL3 - VC Resource 0 Arbitration Table Entry 3 (0x23C)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE24	RW	0xC	<b>Phase 24.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE25	RW	0xC	<b>Phase 25.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE26	RW	0xD	<b>Phase 26.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE27	RW	0xD	<b>Phase 27.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE28	RW	0xE	<b>Phase 28.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE29	RW	0xE	<b>Phase 29.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE30	RW	0xF	<b>Phase 30.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE31	RW	0xF	<b>Phase 31.</b> This field contains the port ID for the corresponding port arbitration period.

# Notes

## VCR1TBL0 - VC Resource 1 Arbitration Table Entry 0 (0x240)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE0	RW	0x0	<b>Phase 0.</b> This field contains the port ID for the corresponding port arbitration period. Selecting an invalid port ID results in the entry being skipped without delay. The port arbitration behavior when this field contains an illegal value (i.e., reserved or the egress port ID) is undefined. 0x0 -(port_0) Port 0 (upstream port) 0x1 -(port_1) Port 1 0x2 -(port_2) Port 2 0x3 -(port_3) Port 3 0x4 -(port_4) Port 4 0x5 -(port_5) Port 5 0x6 -(port_6) Port 6 0x7 -(port_7) Port 7 0x8 -(port_8) Port 8 0x9 -(port_9) Port 9 0x10 -(port_10) Port 10 0x11 -(port_11) Port 11 0x12 -(port_12) Port 12 0x13 -(port_13) Port 13 0x14 -(port_14) Port 14 0x15 -(port_15) Port 15
7:4	PHASE1	RW	0x0	<b>Phase 1.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE2	RW	0x1	<b>Phase 2.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE3	RW	0x2	<b>Phase 3.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE4	RW	0x3	<b>Phase 4.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE5	RW	0x3	<b>Phase 5.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE6	RW	0x4	<b>Phase 6.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE7	RW	0x4	<b>Phase 7.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR1TBL1 - VC Resource 1 Arbitration Table Entry 1 (0x244)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE8	RW	0x4	<b>Phase 8.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE9	RW	0x4	<b>Phase 9.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE10	RW	0x5	<b>Phase 10.</b> This field contains the port ID for the corresponding port arbitration period.

# Notes

Bit Field	Field Name	Type	Default Value	Description
15:12	PHASE11	RW	0x5	<b>Phase 11.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE12	RW	0x6	<b>Phase 12.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE13	RW	0x6	<b>Phase 13.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE14	RW	0x7	<b>Phase 14.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE15	RW	0x7	<b>Phase 15.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR1TBL2 - VC Resource 1 Arbitration Table Entry 2 (0x248)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE16	RW	0x8	<b>Phase 16.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE17	RW	0x8	<b>Phase 17.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE18	RW	0x9	<b>Phase 18.</b> This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE19	RW	0x9	<b>Phase 19.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE20	RW	0x10	<b>Phase 20.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE21	RW	0x10	<b>Phase 21.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE22	RW	0x11	<b>Phase 22.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE23	RW	0x11	<b>Phase 23.</b> This field contains the port ID for the corresponding port arbitration period.

## VCR1TBL3 - VC Resource 1 Arbitration Table Entry 3 (0x24C)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE24	RW	0xC	<b>Phase 24.</b> This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE25	RW	0xC	<b>Phase 25.</b> This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE26	RW	0xD	<b>Phase 26.</b> This field contains the port ID for the corresponding port arbitration period.

## Notes

Bit Field	Field Name	Type	Default Value	Description
15:12	PHASE27	RW	0xD	<b>Phase 27.</b> This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE28	RW	0xE	<b>Phase 28.</b> This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE29	RW	0xE	<b>Phase 29.</b> This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE30	RW	0xF	<b>Phase 30.</b> This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE31	RW	0xF	<b>Phase 31.</b> This field contains the port ID for the corresponding port arbitration period.

### Power Budgeting Enhanced Capability

#### PWRBCAP - Power Budgeting Capabilities (0x280)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RWL	0x0	<b>Capability ID.</b> The value of 0x4 indicates a power budgeting capability structure. If the power budgeting capability is used, then this field should be initialized with data from a serial EEPROM.
19:16	CAPVER	RWL	0x0	<b>Capability Version.</b> The value of 0x1. indicates compatibility with version 1 of the specification. If the power budgeting capability is used, then this field should be initialized with data from a serial EEPROM.
31:20	NXTPTR	RWL	0x0	<b>Next Pointer.</b>

#### PWRBDSEL - Power Budgeting Data Select (0x284)

Bit Field	Field Name	Type	Default Value	Description
7:0	DVSEL	RW	0x0	<b>Data Value Select.</b> This field selects the Power Budgeting Data Value (PWRBDVx) register whose contents are reported in the Data (DATA) field of the Power Budgeting Data (PWRBD) register. Setting this field to a value greater than 7, causes zero to be returned in the DATA field of the PWRBD register.
31:8	Reserved	RO	0x0	Reserved field.

## Notes

### PWRBD - Power Budgeting Data (0x288)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RO	0x0	<b>Data.</b> If the Data Value Select (DVSEL) field in the Power Budgeting Data Select register contains a value of zero through 31, then this field returns the contents of the corresponding Power Budgeting Data Value (PWRBDVx) register. Otherwise, this field contains a value of zero.

### PWRBPBC - Power Budgeting Power Budget Capability (0x28C)

Bit Field	Field Name	Type	Default Value	Description
0	SA	RWL	0x0	<b>System Allocated.</b> When this bit is set, it indicates that the power budget for the device is included within the system power budget and that reported power data for this device should be ignored. If the power budgeting capability is used, then this field should be initialized with data from a serial EEPROM.
31:1	Reserved	RO	0x0	Reserved field.

### PWRBDV[7:0] - Power Budgeting Data Value [7:0] (0x300)

Bit Field	Field Name	Type	Default Value	Description
31:0	DV	RW	Undefined Sticky	<b>Data Value.</b> This 32-bit field is used to hold power budget data in the format described in Section 7.13.3 in the PCIe 1.0 Base Specification. This field may be read and written when the Power Budgeting Data Value Unlock (PWRBDVUL) bit is set in the System Control (SYSCTL) register. When the PWRBDVUL bit is cleared, this register is read-only and writes are ignored. If the power budgeting capability is used, then this field should be initialized with data from a serial EEPROM.

# Notes

## Switch Control and Status Registers

### SWSTS - Switch Status (0x400)

Bit Field	Field Name	Type	Default Value	Description
3:0	SWMODE	RO	HWINIT	<b>Switch Mode.</b> These configuration pins determine the PES22H16 switch operating mode. These pins should be static and not change following the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 through 0x7 - Reserved 0x8 - Normal switch mode with upstream port failover (port 0 selected as the upstream port) 0x9 - Normal switch mode with upstream port failover (port 2 selected as the upstream port) 0xA - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 0 selected as the upstream port) 0xB - Normal switch mode with Serial EEPROM initialization and upstream port failover (port 2 selected as the upstream port) 0xC through 0xF - Reserved
4	Reserved	RO	0x0	Reserved field.
5	CCLKDS	RO	HWINIT	<b>Common Clock Downstream.</b> This bit reflects the value of the CCLKDS signal sampled during the fundamental reset.
6	CCLKUS	RO	HWINIT	<b>Common Clock Upstream.</b> This bit reflects the value of the CCLKUS signal sampled during the fundamental reset.
7	MSMBSMODE	RO	HWINIT	<b>Master SMBus Slow Mode.</b> This bit reflects the value of the MSMBSMODE signal sampled during the fundamental reset.
8	REFCLKM	RO	HWINIT	<b>PCI Express Reference Clock Mode Select.</b> This bit reflects the value of the REFCLKM signal sampled during the fundamental reset.
9	RSTHALT	RO	HWINIT	<b>Reset Halt.</b> This bit reflects the value of the RSTHALT signal sampled during the fundamental reset.
11:10	PEMODE	RWL	HWINIT	<b>PCI Express Base Specification Mode.</b> This field selects the PCIe base specification operating mode for the PES22H16. 0x0 - Reserved 0x1 - (pebase1p1) PCIe 1.1 base specification compliant mode. This is the initial value of this field when the PEMODE pad is bonded to Vcc. 0x2 - (pebase1p0a) PCIe 1.0a base specification compatibility mode. 0x3 - Reserved
12	P01MERGEN	RO	0x0	<b>Port 0 and 1 Merge.</b> This bit reflects the value of the P01MERGEN signal sampled during the fundamental reset.
13	P23MERGEN	RO	0x0	<b>Port 2 and 3 Merge.</b> This bit reflects the value of the P23MERGEN signal sampled during the fundamental reset.
14	P45MERGEN	RO	0x0	<b>Port 4 and 5 Merge.</b> This bit reflects the value of the P45MERGEN signal sampled during the fundamental reset.

# Notes

Bit Field	Field Name	Type	Default Value	Description
15	P67MERGEN	RO	0x0	<b>Port 6 and 7 Merge.</b> This bit reflects the value of the P67MERGEN signal sampled during the fundamental reset.
16	P89MERGEN	RO	0x0	<b>Port 8 and 9 Merge.</b> This bit reflects the value of the P89MERGEN signal sampled during the fundamental reset.
17	P1011MERGEN	RO	0x0	<b>Port 10 and 11 Merge.</b> This bit reflects the value of the P1011MERGEN signal sampled during the fundamental reset.
18	P1213MERGEN	RO	0x0	<b>Port 12 and 13 Merge.</b> This bit reflects the value of the P1213MERGEN signal sampled during the fundamental reset.
19	P1415MERGEN	RO	0x0	<b>Port 14 and 15 Merge.</b> This bit reflects the value of the P1415MERGEN signal sampled during the fundamental reset.
22:20	LOCKMODE	RO	0x0	<b>Lock Mode.</b> This field reflects the current locked status of the switch. 0x0 -(unlocked) Upstream port is unlocked 0x1 -(port1locked) Upstream port is locked with port 1. 0x2 -(port2locked) Upstream port is locked with port 2. 0x3 -(port3locked) Upstream port is locked with port 3. 0x4 -(port4locked) Upstream port is locked with port 4. 0x5 -(port5locked) Upstream port is locked with port 5. 0x6 -(port6locked) Upstream port is locked with port 6. 0x7 -(port7locked) Upstream port is locked with port 7. 0x8 -(port8locked) Upstream port is locked with port 8. 0x9 -(port9locked) Upstream port is locked with port 9. 0xA -(port10locked) Upstream port is locked with port 10. 0xB -(port11locked) Upstream port is locked with port 11. 0xC -(port12locked) Upstream port is locked with port 12. 0xD -(port13locked) Upstream port is locked with port 13. 0xE -(port14locked) Upstream port is locked with port 14. 0xF -(port15locked) Upstream port is locked with port 15.
27:23	Reserved	RO	0x0	Reserved field.
31:28	MARKER	RW	0x0 Sticky	<b>Marker.</b> This field is preserved across a hot reset and is available for general software use. A hot reset does not result in modification of this field.

# Notes

## SWCTL - Switch Control (0x404)

Bit Field	Field Name	Type	Default Value	Description
0	FRST	RW	0x0	<b>Fundamental Reset.</b> Writing a one to this bit initiates a fundamental reset. Writing a zero has no effect. This field always returns a value of zero when read. See section Fundamental Reset on page 3-4 for the behavior of this bit.
1	HRST	RW	0x0	<b>Hot Reset.</b> Writing a one to this bit initiates a hot reset. Writing a zero has no effect. This field always returns a value of zero when read. See section Hot Reset on page 3-6 for the behavior of this bit.
2	RSTHALT	RW	HWINIT	<b>Reset Halt.</b> When this bit is set, all of the switch logic except the SMBus interface remains in a 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. Setting or clearing this bit has no effect following a reset operation. This bit will be set if during serial EEPROM initialization an error is detected or it may be intentionally set by the user through the EEPROM code.
3	REGUNLOCK	RW	0x0 Sticky	<b>Register Unlock.</b> When this bit is set, the contents of registers and fields of type Read and Write when Unlocked (RWL) are modified when written to. When this bit is cleared, all registers and fields denoted as RWL become read-only. While the initial value of this field is cleared, it is set during a reset operation, thus allowing serial EEPROM initialization to modify the contents of RWL fields.
4	PWRBDVUL	RWL	0x0 Sticky	<b>Power Budgeting Data Value Unlock.</b> When this bit is set, the Power Budgeting Data Value [7:0] (PWRBDV[7:0]) registers in all ports may be read and written. When this bit is cleared, then the PWRBDV registers in all ports are read-only.
5	DLDRST	RW	0x0 Sticky	<b>Disable Link Down Hot Reset.</b> When this bit is set, hot resets due to the data link layer of the upstream port transitioning to the DL_Down state are disabled. All other hot reset conditions are unaffected by this bit.
6	DHRSTSEI	RW	0x0 Sticky	<b>Disable Hot Reset Serial EEPROM Initialization.</b> When this bit is set, step 6 "serial EEPROM initialization" is skipped in the hot reset sequence described in section Hot Reset on page 3-6 regardless of the selected switch operating mode.
7	DRO	RW	0x0 Sticky	<b>Disable Relaxed Ordering.</b> The switch implements relaxed ordering for TLPs with the relaxed ordering bit set. When the DRO bit is set, the switch strongly orders all transactions regardless of the state of the relaxed ordering bit in TLPs.
31:8	Reserved	RO	0x0	Reserved field.

# Notes

## HPCFGCTL - Hot-Plug Configuration Control (0x408)

Bit Field	Field Name	Type	Default Value	Description
0	IPXAPN	RW	0x0 Sticky	<b>Invert Polarity of PxAPN.</b> When this bit is set, the polarity of the PxAPN input is inverted in all ports.
1	IPXPDN	RW	0x0 Sticky	<b>Invert Polarity of PxPDN.</b> When this bit is set, the polarity of the PxPDN input is inverted in all ports.
2	IPXPFN	RW	0x0 Sticky	<b>Invert Polarity of PxPFN.</b> When this bit is set, the polarity of the PxPFN input is inverted in all ports.
3	IPXMRLN	RW	0x0 Sticky	<b>Invert Polarity of PxMRLN.</b> When this bit is set, the polarity of the PxMRLN input is inverted in all ports.
4	IPXAIN	RW	0x0 Sticky	<b>Invert Polarity of PxAIN.</b> When this bit is set, the polarity of the PxAIN output is inverted in all ports.
5	IPXPIN	RW	0x0 Sticky	<b>Invert Polarity of PxPIN.</b> When this bit is set, the polarity of the PxPIN output is inverted in all ports.
6	IPXPEP	RW	0x0 Sticky	<b>Invert Polarity of PxPEP.</b> When this bit is set, the polarity of the PxPEP output is inverted in all ports.
7	IPXLOCKP	RW	0x0 Sticky	<b>Invert Polarity of PxLOCKP.</b> When this bit is set, the polarity of the PxLOCKP output is inverted in all ports.
8	IPXP-WRGDN	RW	0x0 Sticky	<b>Invert Polarity of PxPWRGDN.</b> When this bit is set, the polarity of the PxPWRGDN input is inverted in all ports.
10:9	Reserved	RW	0x0	Reserved field.
11	MRLP-WROFF	RW	0x1 Sticky	<b>When the MRL Automatic Power Off.</b> 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	RMRL-WEMIL	RW	0x0 Sticky	<b>Replace MRL Status with EMIL Status.</b> When this bit is set, the PxMRLN signal inputs are used as electromechanical lock state inputs.
13	TEMICTL	RW	0x0 Sticky	<b>Toggle Electromechanical Interlock Control.</b> When this bit is cleared, the Electromechanical Interlock (PxLOCKP) output is pulsed for 125 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 PxLOCKP output.

## Notes

Bit Field	Field Name	Type	Default Value	Description
15:14	RSTMODE	RW	0x0 Sticky	<b>Reset Mode.</b> This field controls the manner in which downstream port reset outputs are generated. 0x0 - (pec) Power enable controlled reset output 0x1 - (pgc) Power good controlled reset output 0x2 - (hot) Hot reset controlled reset output 0x3 - Reserved
23:16	PWR2RST	RW	0x14 Sticky	<b>Slot Power to Reset Negation.</b> This field contains the delay from stable downstream port power to negation of the downstream 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 Sticky	<b>Reset Negation.</b> This field contains the delay from negation of a downstream port's reset to disabling of a downstream port's power in units of 10 mS. A value of zero corresponds to no delay. The default value corresponds to 200 mS.

### GPR - General Purpose Register (0x40C)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	<b>Data.</b> General purpose register available for software use.

### GPIOFUNC - General Purpose I/O Control Function (0x418)

Bit Field	Field Name	Type	Default Value	Description
31:0	GPIOFUNC	RW	0x0 Sticky	<b>GPIO Function.</b> Each bit in this field controls the corresponding GPIO pin. When set to a one, the corresponding GPIO pin operates as the alternate function as defined in Table 5.1. When a bit is cleared to a zero, the corresponding GPIO pin operates as a general purpose I/O pin.

### GPIOCFG - General Purpose I/O Configuration (0x41C)

Bit Field	Field Name	Type	Default Value	Description
31:0	GPIOCFG	RW	0x0 Sticky	<b>GPIO Configuration.</b> 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 zero, 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.

# Notes

## GPIOD - General Purpose I/O Data (0x420)

Bit Field	Field Name	Type	Default Value	Description
31:0	GPIOD	RW	HWINIT Sticky	<b>GPIO Data.</b> 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.

## SMBUSSTS - SMBus Status (0x424)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	SSMBADDR	RO	HWINIT	<b>Slave SMBus Address.</b> This field contains the SMBus address assigned to the slave SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	MSMBADDR	RO	HWINIT	<b>Master SMBus Address.</b> This field contains the SMBus address assigned to the master SMBus interface.
23:16	Reserved	RO	0x0	Reserved field.
24	EEPROM-DONE	RO	0x0	<b>Serial EEPROM Initialization Done.</b> When the switch is configured to operate in a mode in which serial EEPROM initialization occurs during a fundamental reset, this bit is set when serial EEPROM initialization completes or when an error is detected.
25	NAERR	RW1C	0x0	<b>No Acknowledge Error.</b> 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	<b>Lost Arbitration Error.</b> 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	<b>Other Error.</b> This bit is set if a misplaced START or STOP condition is detected by the master SMBus interface.

## Notes

Bit Field	Field Name	Type	Default Value	Description
28	ICSERR	RW1C	0x0	<b>Initialization Checksum Error.</b> 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.
29	URIA	RW1C	0x0	<b>Unmapped Register Initialization Attempt.</b> This bit is set if an attempt is made to initialize via serial EEPROM a register that is not defined in the corresponding PCI configuration space.
31:30	Reserved	RO	0x0	Reserved field.

**SMBUSCTL - SMBus Control (0x428)**

Bit Field	Field Name	Type	Default Value	Description
15:0	MSMBCP	RW	HWINIT Sticky	<b>Master SMBus Clock Prescaler.</b> This field contains a clock prescaler value used during master SMBus transactions. The prescaler 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 0x0139 when the master SMBus is configured to operate in slow mode (i.e., 100 KHz) in the boot configuration and to 0x0053 <sup>1</sup> when it is configured to operate in fast mode (i.e., 400 KHz).
16	MSMBIOM	RW	0x0 Sticky	<b>Master SMBus Ignore Other Masters.</b> When this bit is set, the master SMBus proceeds with transactions regardless of whether it won or lost arbitration.
17	ICHECK-SUM	RW	0x0 Sticky	<b>Ignore Checksum Errors.</b> When this bit is set, serial EEPROM initialization checksum errors are ignored (i.e., the checksum always passes).
19:18	SSMB-MODE	RW	0x0 Sticky	<b>Slave SMBus Mode.</b> The slave SMBus contains internal glitch counters on the SSMBCLK and SSMBDAT signals that wait approximately 1 $\mu$ S 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 1 $\mu$ S delay. 0x1 - (fast) Slave SMBus interface fast mode. Glitch counters operate with 100 nS 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	MSMB-MODE	RW	0x0 Sticky	<b>Master SMBus Mode.</b> The master SMBus contains internal glitch counters on the MSMBCLK and MSMBDAT signals that wait approximately 1 $\mu$ S 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 1 $\mu$ S 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	<b>SMBus Disable Time-out.</b> When this bit is set, SMBus time-outs are disabled on the master and slave SMBuses.
31:23	Reserved	RO	0x0	Reserved field.

<sup>1</sup> 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.

### EEPROMINTF - Serial EEPROM Interface (0x42C)

Bit Field	Field Name	Type	Default Value	Description
15:0	ADDR	RW	0x0	<b>EEPROM Address.</b> This field contains the byte address in the Serial EEPROM to be read or written.
23:16	DATA	RW	0x0	<b>EEPROM Data.</b> A write to this field will initiate 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	<b>EEPROM Busy.</b> 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	<b>EEPROM Operation Completed.</b> 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	<b>EEPROM Operation Select.</b> 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.

# Notes

## IOEXPINTF - I/O Expander Interface (0x430)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOEDATA	RW	0x0	<b>I/O Expander Data.</b> Each bit in this field corresponds to an I/O expander input/output signal. Reading this field returns the current value of the corresponding I/O pin state of the I/O expander number selected in the Select (SEL) field in this register (i.e., the input values last read from the I/O expander and output values supplied to the I/O expander). Writes to this field are ignored unless the I/O Expander Test Mode (IOEXTM) bit is set. When the IOEXTM bit is set, the value for outputs supplied to the I/O expander selected by the SEL field correspond to the value written to this field instead of the value supplied by internal logic. Bits in this field which correspond to inputs are always read-only, even when the IOEXTM bit is set.
23:16	Reserved	RO	0x0	Reserved field.
24	RELOAD-IOEX	RW	0x0	<b>Reload I/O Expander Signals.</b> Writing a one to this field results in an I/O expander SMBus transaction that refreshes all I/O expander input and output signal values in the IOE-DATA field. This bit always returns a zero when read.
25	IOEXTM	RW	0x0	<b>IO Expander Test Mode.</b> Setting this bit puts the I/O expander interface into a test mode. In this test mode, I/O expander output signals generated by the PES22H16 core are ignored and values supplied to the I/O expander correspond to value written to the IOEDATA field.

## Notes

Bit Field	Field Name	Type	Default Value	Description
29:26	SELECT	RW	0x0	<b>I/O Expander Select.</b> This field selects the I/O expander on which fields in this register operate. 0x0 -(ioe0) I/O expander 0 0x1 -(ioe1) I/O expander 1 0x2 -(ioe2) I/O expander 2 0x3 -(ioe3) I/O expander 3 0x4 -(ioe4) I/O expander 4 0x5 -(ioe0) I/O expander 5 0x6 -(ioe1) I/O expander 6 0x7 -(ioe2) I/O expander 7 0x8 -(ioe3) I/O expander 8 0x9 -(ioe4) I/O expander 9 0x10 -(ioe4) I/O expander 10 others - reserved
30	Reserved	RO	0x0	Reserved field.
31	DONE	RW1C	0x0	<b>I/O Expander Operation Done.</b> This bit is set when any of the following conditions occur. RELOADIOEX bit in this register is written, the corresponding I/O expander is selected by the SELECT field in this register, and the corresponding IO expander SMBus transaction completes. The I/O expander is in test mode (i.e., IOEXTM bit set), the IOEDATA field is written, the corresponding I/O expander is selected by the SELECT field in this register, and the corresponding IO expander SMBus transaction updating the I/O expander outputs completes. An I/O Expander Address (IOExADDR) field is written in an SMBus I/O Expander Address (IOEXPADRY) register, the corresponding I/O expander is selected by the SELECT field in this register, and the I/O expander initialization sequence completes.

**IOEXPADDR0 - SMBus I/O Expander Address 0 (0x434)**

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE0ADDR	RWL	0x0 Sticky	<b>I/O Expander 0 Address.</b> 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 Sticky	<b>I/O Expander 1 Address.</b> 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 Sticky	<b>I/O Expander 2 Address.</b> 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 Sticky	<b>I/O Expander 3 Address.</b> This field contains the SMBus address assigned to I/O expander 3 on the master SMBus interface.

## IOEXPADDR1 - SMBus I/O Expander Address 1 (0x438)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE4ADDR	RWL	0x0 Sticky	<b>I/O Expander 4 Address.</b> This field contains the SMBus address assigned to I/O expander 4 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE5ADDR	RWL	0x0 Sticky	<b>I/O Expander 5 Address.</b> This field contains the SMBus address assigned to I/O expander 5 on the master SMBus interface.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE6ADDR	RWL	0x0 Sticky	<b>I/O Expander 6 Address.</b> This field contains the SMBus address assigned to I/O expander 6 on the master SMBus interface.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE7ADDR	RWL	0x0 Sticky	<b>I/O Expander 7 Address.</b> This field contains the SMBus address assigned to I/O expander 7 on the master SMBus interface.

## IOEXPADDR2 - SMBus I/O Expander Address 2 (0x43C)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	IOE8ADDR	RWL	0x0 Sticky	<b>I/O Expander 8 Address.</b> This field contains the SMBus address assigned to I/O expander 8 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE9ADDR	RWL	0x0 Sticky	<b>I/O Expander 9 Address.</b> This field contains the SMBus address assigned to I/O expander 9 on the master SMBus interface.

## Notes

Bit Field	Field Name	Type	Default Value	Description
16	Reserved	RO	0x0	Reserved field.
23:17	IOE10ADDR	RWL	0x0 Sticky	<b>I/O Expander 10 Address.</b> This field contains the SMBus address assigned to I/O expander 10 on the master SMBus interface.
31:24	Reserved	RO	0x0	Reserved field.

### GPECTL - General Purpose Event Control (0x450)

Bit Field	Field Name	Type	Default Value	Description
0	IGPE	RW	0x0 Sticky	<b>Invert General Purpose Event Enable Signal Polarity.</b> 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
1	P1GPEE	RW	0x0 Sticky	<b>Port 1 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 1 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
2	P2GPEE	RW	0x0 Sticky	<b>Port 2 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 2 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
3	P3GPEE	RW	0x0 Sticky	<b>Port 3 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 3 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
4	P4GPEE	RW	0x0 Sticky	<b>Port 4 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 4 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
5	P5GPEE	RW	0x0 Sticky	<b>Port 5 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 5 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].

# Notes

Bit Field	Field Name	Type	Default Value	Description
6	P6GPEE	RW	0x0 Sticky	<b>Port 6 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 6 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
7	P7GPEE	RW	0x0 Sticky	<b>Port 7 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 7 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[7].
8	P8GPEE	RW	0x0 Sticky	<b>Port 8 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 8 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[8].
9	P9GPEE	RW	0x0 Sticky	<b>Port 9 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 9 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[9].
10	P10GPEE	RW	0x0 Sticky	<b>Port 10 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 10 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[10].
11	P11GPEE	RW	0x0 Sticky	<b>Port 11 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 11 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[11].
12	P12GPEE	RW	0x0 Sticky	<b>Port 12 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 12 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[12].
13	P13GPEE	RW	0x0 Sticky	<b>Port 13 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 13 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[13].

## Notes

Bit Field	Field Name	Type	Default Value	Description
14	P14GPEE	RW	0x0 Sticky	<b>Port 14 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 14 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[14].
15	P15GPEE	RW	0x0 Sticky	<b>Port 15 General Purpose Event Enable.</b> When this bit is set, the hot-plug INTx, MSI and PME event notification mechanisms defined by the PCIe base 1.1 specification are disabled for port 15 and are instead signalled through General Purpose Event (GPEN) signal assertions. GPEN is an alternate function of GPIO[15].
31:16	Reserved	RO	0x0	Reserved field.

### GPESTS - General Purpose Event Status (0x454)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
1	P1GPES	RO	0x0	<b>Port 1 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
2	P2GPES	RO	0x0	<b>Port 2 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
3	P3GPES	RO	0x0	<b>Port 3 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
4	P4GPES	RO	0x0	<b>Port 4 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.

# Notes

Bit Field	Field Name	Type	Default Value	Description
5	P5GPES	RO	0x0	<b>Port 5 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
6	P6GPES	RO	0x0	<b>Port 6 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
7	P7GPES	RO	0x0	<b>Port 7 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
8	P8GPES	RO	0x0	<b>Port 8 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
9	P9GPES	RO	0x0	<b>Port 9 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
10	P10GPES	RO	0x0	<b>Port 10 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
11	P11GPES	RO	0x0	<b>Port 11 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.

## Notes

Bit Field	Field Name	Type	Default Value	Description
12	P12GPES	RO	0x0	<b>Port 12 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
13	P13GPES	RO	0x0	<b>Port 13 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
14	P14GPES	RO	0x0	<b>Port 14 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
15	P15GPES	RO	0x0	<b>Port 15 General Purpose Event Status.</b> When this bit is set, the corresponding port is signalling 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 GPECTL register. GPEN is an alternate function of GPIO[5] and GPIO[5] is asserted only if enabled to operate as an alternate function.
31:16	Reserved	RO	0x0	Reserved field.

### USPFSTS - Upstream Port Failover Status (0x470)

Bit Field	Field Name	Type	Default Value	Description
0	USPFEN	RO	HWINIT	<b>Upstream Port Failover Enabled.</b> This bit is set if the switch has been configured to operate in a switch mode that supports upstream port failover.
1	CUSP	RO	HWINIT	<b>Current Upstream Port.</b> This field indicates the current external Sierra switch port that is associated with the upstream port. 0x0 - (portA) port A is associated with the upstream port. 0x1 - (portB) port B is associated with the upstream port.
2	USPC	RW1C	0x0 Sticky	<b>Upstream Port Change.</b> This bit is set when a change completes in the external Sierra switch port that is associated with the upstream port.
3	USPCI	RW1C	0x0 Sticky	<b>Upstream Port Change Initiated.</b> This bit is set when a change is initiated in the external Sierra switch port that is associated with the upstream port.
31:4	Reserved	RO	0x0	Reserved field.

# Notes

## USPFCTL - Upstream Port Failover Control (0x474)

Bit Field	Field Name	Type	Default Value	Description
0	USPSEL	RW	HWINIT Sticky	<b>Upstream Port Software Select.</b> Modifying the value in this field initiates an upstream port failover. When read, this field indicates the current upstream port or the previous value written if an upstream port failover is in progress. Modifying this field while an upstream port failover is in progress produces undefined results. 0x0 - (port0) port 0 is associated with the upstream port. 0x1 - (port2) port 2 is associated with the upstream port. This field has no effect on system operation when the an upstream port failover switch mode has not been selected during a fundamental reset.
1	SIGFEN	RW	HWINIT Sticky	<b>Upstream Port Signal Failover Enable.</b> When this bit is set, an upstream port failover is initiated when the state of the Upstream Port Select (USPSEL) signal does not match the state of the Current Upstream Port (CUSP) field in the Upstream Port Failover Status (USPFSTS). The USPSEL signal is a GPIO alternate function. This field has no effect on system operation when the an upstream port failover switch mode has not been selected during a fundamental reset.
2	TIMFEN	RW	HWINIT Sticky	<b>Upstream Port Timer Failover Enable.</b> When this bit is set, an upstream port failover is initiated whenever the Count (COUNT) field in the Upstream Port Failover Watchdog Timer (USPFTIMER) register reaches zero (i.e., the watchdog timer expires). This field has no effect on system operation when the an upstream port failover switch mode has not been selected during a fundamental reset.
31:3	Reserved	RO	0x0	Reserved field.

## USPFTIMER - Upstream Port Failover Watchdog Timer (0x478)

Bit Field	Field Name	Type	Default Value	Description
31:0	COUNT	RW	0x0 Sticky	<b>Watchdog Timer Count.</b> This field contains the current watchdog timer count value. The value in this field is decremented by one every microsecond (1 $\mu$ S). When this field reaches zero, it stops decremented and the timer is said to have expired. When the value of this field transitions from one to zero, the Upstream Port Timer Failover Enable (TIMFEN) bit is set in the Upstream Port Failover Control (USPFCTL) register, and the switch is in a mode that support upstream port failover, then an upstream failover is initiated. This field has no effect if the switch is not in a mode that supports upstream port failover.

# Notes

## Internal Switch Error Control and Status Registers

### SWPECTL - Switch Parity Error Control (0x740)

Bit Field	Field Name	Type	Default Value	Description
0	DEEPC	RW	0x0 Sticky	<b>Disable End-to-End Parity Checking.</b> When this bit is set, end-to-end parity is not checked by the port and errors are never generated. End-to-end parity is always computed for data sent by the port to the switch core and cannot be disabled.
1	GBEEP	RW	0x0 Sticky	<b>Generate Bad End-to-End Parity.</b> When this bit is set, bad parity is generated for all double words in TLPs emitted to the switch core from this port (i.e., those received on the ingress port or generated by the port) whose TLP header length field (i.e., bits seven through zero of byte zero of the TLP header) match the value in the Error Match Length (Length) field in this register
7:2	Reserved	RO	0x0	Reserved field.
15:8	LENGTH	RW	0x0 Sticky	<b>Error Match Length.</b> When the GBEEP bit is set, bad parity is generated for all double words in TLPs emitted to the switch core from this port (i.e., those received on the ingress port or generated by the stack) whose TLP header length field (i.e., bits seven through zero of byte zero of the TLP header) matches the value in this field.
31:16	Reserved	RO	0x0	Reserved field.

### SWPESTS - Switch Parity Error Status (0x744)

Bit Field	Field Name	Type	Default Value	Description
0	EEPE	RW1C	0x0	<b>End-to-End Parity Error.</b> This bit is set when an end-to-end parity error is detected at the port.
31:1	Reserved	RO	0x0	Reserved field.

## Notes

### SWPERCTL - Switch Parity Error Reporting Control (0x748)

Bit Field	Field Name	Type	Default Value	Description
1:0	EEPE	RW	0x2	<b>End-to-End Parity Error Reporting.</b> This field controls the manner in which end-to-end parity errors detected at this port are reported. An end-to-end parity error is reported as specified in this field whenever the EEPE bit in the Switch Parity Error Status (SWPESTS) register transitions from a zero to a one. 0x0 - (none) The end-to-end parity error is not reported 0x1 - (correctable) The port generates an ERR_COR message to the root. 0x2 - (nonfatal) The port generates an ERR_NONFATAL message to the root. 0x3 - (fatal) The port generates an ERR_FATAL message to the root.
31:2	Reserved	RO	0x0	Reserved field.

### SWPECNT - Switch Parity Error Count (0x74C)

Bit Field	Field Name	Type	Default Value	Description
7:0	EEPEC	RCW	0x0 Sticky	<b>End-to-End Parity Error Count.</b> This field is incremented each time an end-to-end parity error is detected at the port until it saturates at its maximum count value (i.e., it does not roll over from 0xFF to 0x00). This counter saturates at its maximum value. Reading this field causes it to be cleared.
31:8	Reserved	RO	0x0	Reserved field.

### SWTOSTS - Switch Time-Out Status (0x754)

Bit Field	Field Name	Type	Default Value	Description
0	PTLPTO	RW1C	0x0	<b>Posted TLP Time-Out.</b> This bit is set when a TLP is discarded from the port's IFB posted queue because it is more than 50 ms old.
1	NPTLPTO	RW1C	0x0	<b>Non-Posted TLP Time-Out.</b> This bit is set when a TLP is discarded from the port's IFB non-posted queue because it is more than 50 ms old.
2	CPTLPTO	RW1C	0x0	<b>Completion TLP Time-Out.</b> This bit is set when a TLP is discarded from the port's IFB completion queue because it is more than 50 ms old.
3	ITLPTO	RW1C	0x0	<b>Inserted TLP Time-Out.</b> This bit is set when a TLP is discarded from the port's IFB insertion queue because it is more than 50 ms old.
31:4	Reserved	RO	0x0	Reserved field.

# Notes

## SWTORCTL - Switch Time-Out Reporting Control (0x758)

Bit Field	Field Name	Type	Default Value	Description
1:0	PTLPTO	RW	0x2 Sticky	<b>Posted TLP Time-Out Reporting.</b> This field controls the manner in which posted TLP time-outs are reported. A time-out is reported as specified in this field whenever the corresponding bit in the Switch Time-Out Status (SWTOSTS) register transitions from a zero to a one. 0x0 - (none) The time-out is not reported 0x1 - (correctable) The port generates an ERR_COR message to the root. 0x2 - (nonfatal) The port generates an ERR_NONFATAL message to the root. 0x3 - (fatal) The port generates an ERR_FATAL message to the root.
3:2	NPTLPTO	RW	0x0 Sticky	<b>Non-Posted TLP Time-Out Reporting.</b> This field controls the manner in which non-posted TLP time-outs are reported. A time-out is reported as specified in this field whenever the corresponding bit in the Switch Time-Out Status (SWTOSTS) register transitions from a zero to a one. 0x0 - (none) The time-out is not reported 0x1 - (correctable) The port generates an ERR_COR message to the root. 0x2 - (nonfatal) The port generates an ERR_NONFATAL message to the root. 0x3 - (fatal) The port generates an ERR_FATAL message to the root.
5:4	CPTLPTO	RW	0x0 Sticky	<b>Completion TLP Time-Out Reporting.</b> This field controls the manner in which completion TLP time-outs are reported. A time-out is reported as specified in this field whenever the corresponding bit in the Switch Time-Out Status (SWTOSTS) register transitions from a zero to a one. 0x0 - (none) The time-out is not reported 0x1 - (correctable) The port generates an ERR_COR message to the root. 0x2 - (nonfatal) The port generates an ERR_NONFATAL message to the root. 0x3 - (fatal) The port generates an ERR_FATAL message to the root.
7:6	ITLPTO	RW	0x2 Sticky	<b>Inserted TLP Time-Out Reporting.</b> This field controls the manner in which inserted TLP time-outs are reported. A time-out is reported as specified in this field whenever the corresponding bit in the Switch Time-Out Status (SWTOSTS) register transitions from a zero to a one. 0x0 - (none) The time-out is not reported 0x1 - (correctable) The port generates an ERR_COR message to the root. 0x2 - (nonfatal) The port generates an ERR_NONFATAL message to the root. 0x3 - (fatal) The port generates an ERR_FATAL message to the root.
31:8	Reserved	RO	0x0	Reserved field.

## Notes

## SWTOCNT - Switch Time-Out Count (0x75C)

Bit Field	Field Name	Type	Default Value	Description
7:0	PTLPTOC	RCW	0x0 Sticky	<b>Posted TLP Time-Out Count.</b> This field is incremented each time a TLP is discarded from the port's IFB posted queue because it is more than 50 ms old. This counter saturates at its maximum value. Reading this field causes it to be cleared.
15:8	NPTLPTOC	RCW	0x0 Sticky	<b>Non-Posted TLP Time-Out Count.</b> This field is incremented each time a TLP is discarded from the port's IFB non-posted queue because it is more than 50 ms old. This counter saturates at its maximum value. Reading this field causes it to be cleared.
23:16	CTLPTOC	RCW	0x0 Sticky	<b>Completion TLP Time-Out Count.</b> This field is incremented each time a TLP is discarded from the port's IFB completion queue because it is more than 50 ms old. This counter saturates at its maximum value. Reading this field causes it to be cleared.
31:24	ITLPTOC	RCW	0x0 Sticky	<b>Inserted TLP Time-Out Count.</b> This field is incremented each time a TLP is discarded from the port's IFB insertion queue because it is more than 50 ms old. This counter saturates at its maximum value. Reading this field causes it to be cleared.

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 PES22H16: AC-coupled and DC-coupled (also called AC and DC pins). The Boundary Scan interface in the PES22H16 is IEEE 1149.1 compliant to allow testing of the DC pins. The DC pins are those “normal” pins that do not require AC-coupling.

The presence of AC-coupling capacitors on some of the PES22H16 pins prevents DC values from being driven between a driver and receiver. An AC Boundary Scan methodology, as described in IEEE 1149.6, is available to provide a time-varying signal to pass through the AC-coupling when in AC test mode; however, IEEE 1149.6 is not supported in the PES22H16.

## 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 PES22H16's many external signal pins. The JTAG TAP Controller can also be used for identifying the device part number. The JTAG logic of the PES22H16 is depicted in Figure 10.1.

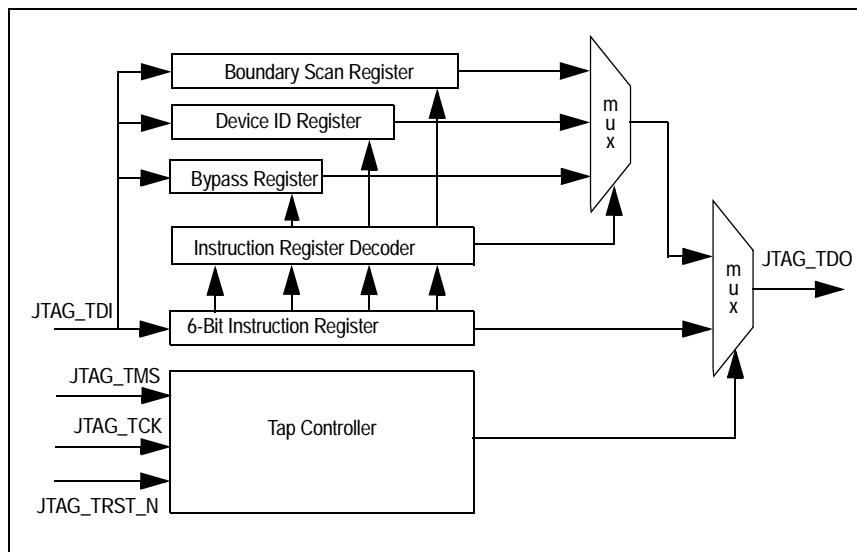


Figure 10.1 Diagram of the JTAG Logic

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 10.1. 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 10.1 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 10.2. 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.

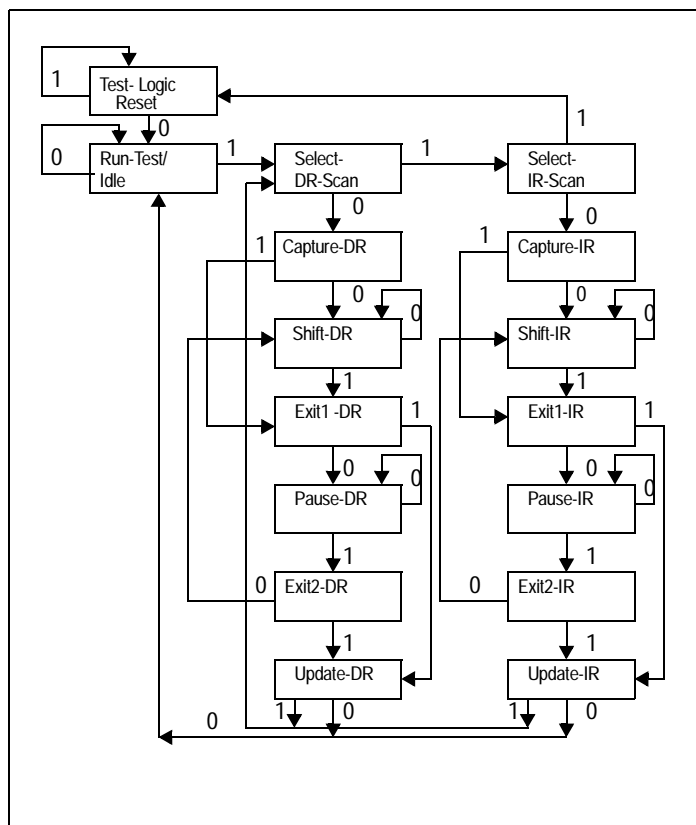


Figure 10.2 State Diagram of PES22H16's TAP Controller

# Notes

## Boundary Scan Chain

Function	Pin Name	Type <sup>1</sup>	Boundary Cell <sup>2</sup>
PCI Express Interface	PE0RN[3:0]	I	O
	PE0RP[3:0]	I	O
	PE0TN[3:0]	O	C
	PE0TP[3:0]	O	
	PE1RN[0]	I	O
	PE1RP[0]	I	O
	PE1TN[0]	O	C
	PE1TP[0]	O	
	PE2RN[3:0]	I	O
	PE2RP[3:0]	I	O
	PE2TN[3:0]	O	C
	PE2TP[3:0]	O	
	PE3RN[0]	I	O
	PE3RP[0]	I	O
	PE3TN[0]	O	C
	PE3TP[0]	O	
	PE4RN[0]	I	O
	PE4RP[0]	I	O
	PE4TN[0]	O	C
	PE4TP[0]	O	
	PE5RN[0]	I	O
	PE5RP[0]	I	O
	PE5TN[0]	O	C
	PE5TP[0]	O	
	PE6RN[0]	I	O
	PE6RP[0]	I	O
	PE6TN[0]	O	C
	PE6TP[0]	O	
	PE7RN[0]	I	O
	PE7RP[0]	I	O
	PE7TN[0]	O	C
	PE7TP[0]	O	
	PE8RN[0]	I	O
	PE8RP[0]	I	O
	PE8TN[0]	O	C
	PE8TP[0]	O	
	PE9RN[0]	I	O

Table 10.2 Boundary Scan Chain (Part 1 of 3)

# Notes

Function	Pin Name	Type <sup>1</sup>	Boundary Cell <sup>2</sup>
PCI Express Interface (cont.)	PE9RP[0]	I	O
	PE9TN[0]	O O	C
	PE9TP[0]		
	PE10RN[0]	I	O
	PE10RP[0]	I	O
	PE10TN[0]	O O	C
	PE10TP[0]		
	PE11RN[0]	I	O
	PE11RP[0]	I	O
	PE11TN[0]	O O	C
	PE11TP[0]		
	PE12RN[0]	I	O
	PE12RP[0]	I	O
	PE12TN[0]	O O	C
	PE12TP[0]		
	PE13RN[0]	I	O
	PE13RP[0]	I	O
	PE13TN[0]	O O	C
	PE13TP[0]		
	PE14RN[0]	I	O
	PE14RP[0]	I	O
	PE14TN[0]	O O	C
	PE14TP[0]		
	PE15RN[0]	I	O
	PE15RP[0]	I	O
	PE15TN[0]	O O	C
	PE15TP[0]		
	PEREFCLKN[3:0]	I	—
	PEREFCLKP[3:0]	I	—
	REFCLKM	I	O
SMBus	MSMBADDR[4:1]	I	O
	MSMBCLK	I/O	O/C
	MSMBDAT	I/O	O/C
	SSMBADDR[5,3:1]	I	O
	SSMBCLK	I/O	O/C
	SSMBDAT	I/O	O/C
General Purpose I/O	GPIO[31:0]	I/O	O/C

Table 10.2 Boundary Scan Chain (Part 2 of 3)

## Notes

Function	Pin Name	Type <sup>1</sup>	Boundary Cell <sup>2</sup>
System Pins	CCLKDS	I	0
	CCLKUS	I	0
	MSMBSMODE	I	0
	PERSTN	I	0
	RSTHALT	I	0
	SWMODE[3:0]	I	—
EJTAG / JTAG	JTAG_TCK	I	—
	JTAG_TDI	I	—
	JTAG_TDO	O	—
	JTAG_TMS	I	—
	JTAG_TRST_N	I	—

**Table 10.2 Boundary Scan Chain (Part 3 of 3)**

<sup>1</sup> I = Input, O = Output

<sup>2</sup> O = Observe, C = Control

## 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).

### 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 10.3.

# Notes

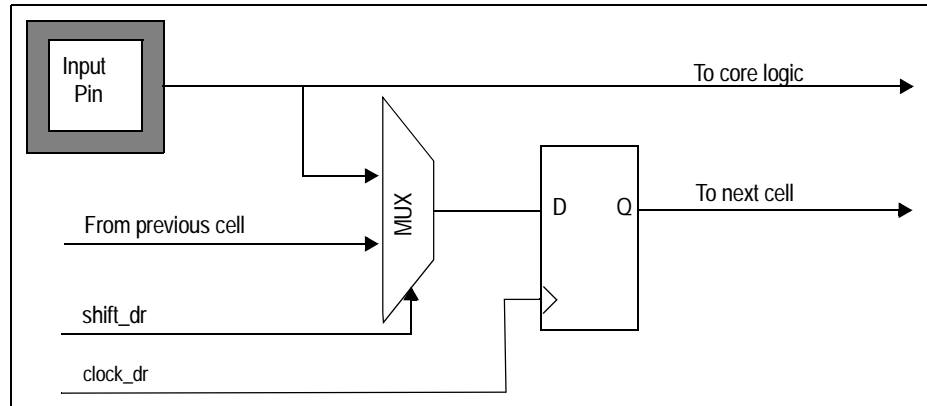


Figure 10.3 Diagram of Observe-only Input Cell

The simplified logic configuration of the output cells is shown in Figure 10.4.

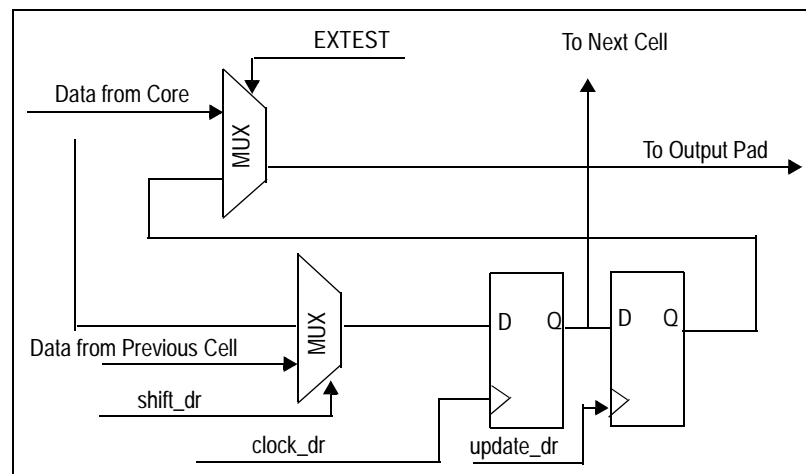


Figure 10.4 Diagram of Output Cell

The output enable cells are also output cells. The simplified logic is shown in Figure 10.5.

# Notes

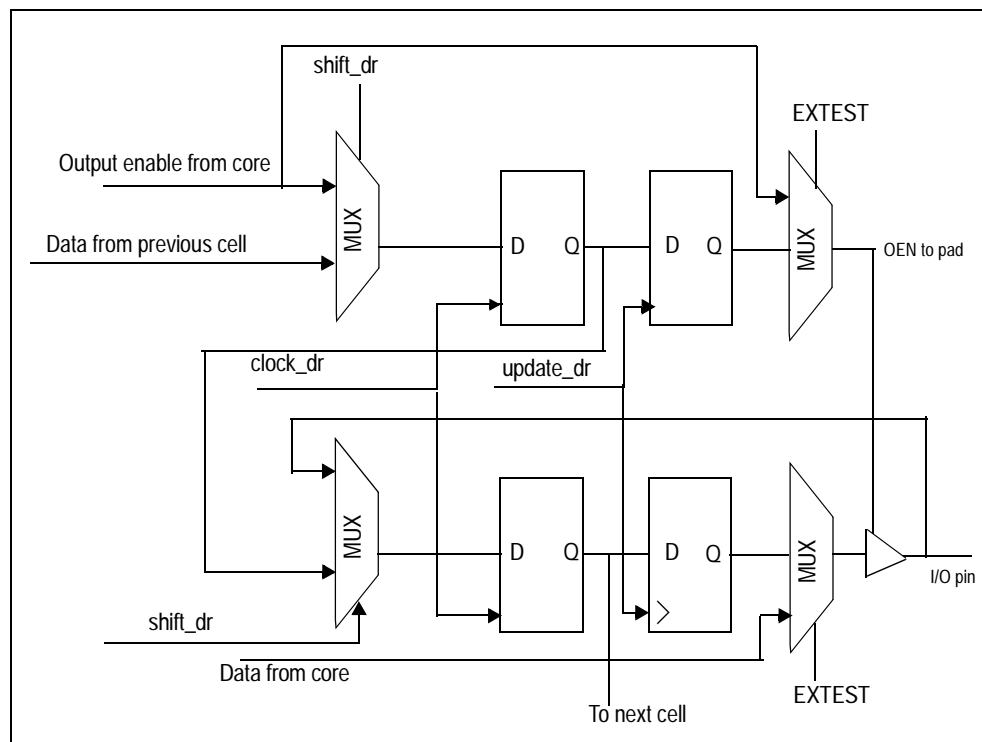


Figure 10.5 Diagram of Output Enable Cell

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 10.5.

## 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 10.3.

## 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
RESERVED		000100 — 101100
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
RESERVED		101110 — 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

**Table 10.3 Instructions Supported by PES22H16's 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 PES22H16. 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 PES22H16, 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 10.6.

Bit(s)	Mnemonic	Description	R/W	Reset
0	Reserved	Reserved	R	0x1
11:1	Manuf_ID	<b>Manufacturer Identity</b> (11 bits) This field identifies the manufacturer as IDT.	R	0x33
27:12	Part_number	<b>Part Number</b> (16 bits) This field identifies the silicon as PES22H16.	R	0x8034
31:28	Version	<b>Version</b> (4 bits) This field identifies the silicon revision of the PES22H16.	R	silicon-dependent

Table 10.4 System Controller Device Identification Register

Version	Part Number	Mnfg. ID	LSB
xxxx	1000 0000 0011 0100	0000 0011 0011	1

Figure 10.6 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.

## RESERVED

Reserved instructions implement various test modes used in the device manufacturing process. The user should not enable these instructions.

## Notes

### Usage Considerations

As previously stated, there are internal pull-ups on JTAG\_TRST\_N, JTAG\_TMS, and JTAG\_TDI. However, JTAG\_TCK also needs to be driven to a known value. It is best to either drive a zero on the JTAG\_TCK pin when it is not being used or to use an external pull-down resistor. In order to guarantee that the 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 and/or JTAG\_TMS high when the chip is in normal operation. If JTAG will not be used, externally pull-down JTAG\_TRST\_N low to disable it.

## IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit [www.renesas.com/contact-us/](http://www.renesas.com/contact-us/).

### Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.