IDT™ 89HPES32T8
PCI Express® Switch

User Manual

April 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.

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.

About This Manual

Notes

Introduction

This user manual includes hardware and software information on the 89HPES32T8, 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)** as well as through your local IDT sales representative.

Content Summary

Chapter 1, "PES32T8 Device Overview," provides a complete introduction to the performance capabilities of the 89HPES32T8. Included in this chapter is a summary of features for the device as well as a system block diagram and pin description.

Chapter 2, "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 3, "Theory of Operation," provides basic information on the architecture and operation of the 89HPES32T8 chip.

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

Chapter 4, "Switch Operation," discusses the procedure for forwarding PCIe® TLPs between switch ports.

Chapter 5, "General Purpose I/O," describes how the 16 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 PES32T8.

Chapter 7, "Power Management," describes the power management capability structure located in the configuration space of each PCI-PCI bridge in the PES32T8.

Chapter 8, "Hot-Plug and Hot-Swap," describes the behavior of the hot-plug and hot-swap features in the PES32T8.

Chapter 9, "Configuration Registers," discusses the base addresses, PCI configuration space, and registers associated with the PES32T8.

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.

Notes

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

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

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

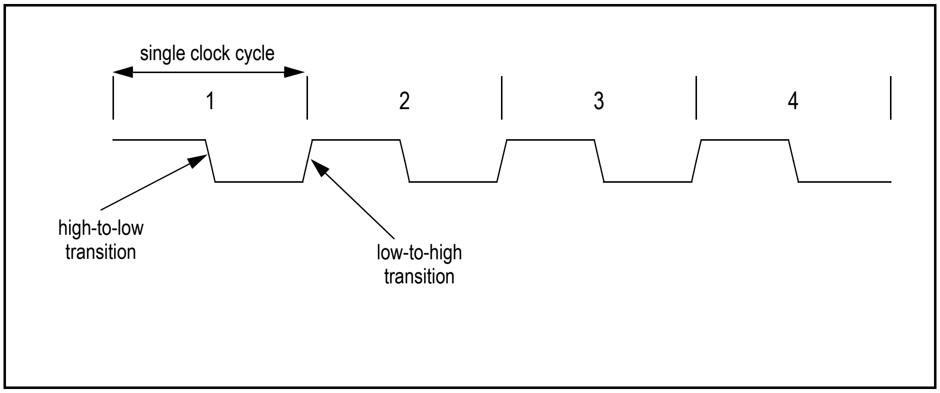


Figure 1 Signal Transitions

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 if x < y or to ABCxD, ABC(x-1)D, ABC(x-2)D,... ABCyD if x > y.

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.

Notes

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.

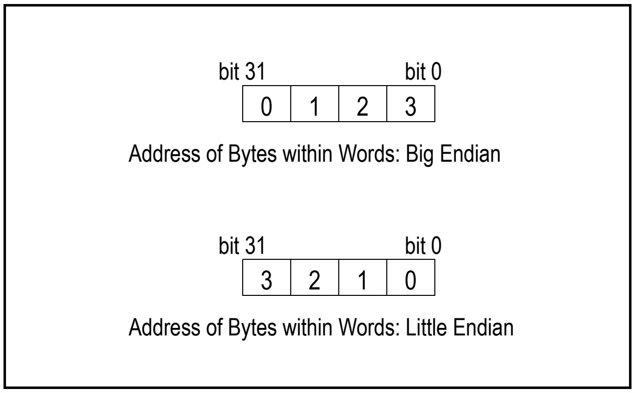

bit 31
bit 0
0
1
2
3
Address of Bytes within Words: Big Endian
bit 31
bit 0
3
2
1
0
Address of Bytes within Words: Little Endian

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 quality 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 2.0, 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

February 8, 2007: Initial Publication.

March 23, 2007: In Chapter 2, changed port description in Figures 2.1 through 2.4.

May 30, 2007: In Table 1.2, added revision information for ZH silicon. Added Notes to Figure 2.5.

July 18, 2007: In Chapter 9, changed bits [10:9] in HPCFGCTL from RO to RW. In Chapter 2, changed references to correctly state SRESET field is in BCTL register, not the SWCTL register.

April 10, 2008: In the About section, Table 2, changed SYSCNTL to SWCTL. In Chapter 9, changed default value for VER field in PCIECAP register from 0x2 to 0x1 and changed 0x0 definition for bit EEPE in SWPERCTL register from “time-out” to “end-to-end parity error”.

Table of Contents

Notes

About This Manual

PES32T8 Device Overview

Clocking, Reset, and Initialization

Theory of Operation

Notes

Link Operation

General Purpose I/O

SMBus Interfaces

Power Management

Notes

Hot-Plug and Hot-Swap

Configuration Registers

JTAG Boundary Scan

Notes

List of Tables

Notes

Notes

List of Figures

Notes

Notes

Notes

Notes

Notes

PES32T8 Device Overview

Notes

Introduction

The 89HPES32T8 is a member of the IDT PRECISE™ family of PCI Express® switching solutions. The PES32T8 is a 32-lane, 8-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 seven downstream ports and supports switching between downstream ports.

Utilizing standard PCI Express interconnect, the PES32T8 provides the most efficient I/O connectivity solution for applications requiring high throughput, low latency, and simple board layout with a minimum number of board layers. It provides connectivity for up to 8 ports across 32 integrated serial lanes. Each lane provides 2.5 Gbps of bandwidth in both directions and is fully compliant with PCI Express Base specification revision 1.1.

The PES32T8 is based on a flexible and efficient layered architecture. The PCI Express layers consists of SerDes, Physical, Data Link and Transaction layers. The PES32T8 can operate either as a store and forward switch or a cut-through switch and is designed to switch memory and I/O transactions. It supports eight Traffic Classes (TCs) and one Virtual Channel (VC) with sophisticated resource management to enable efficient switching and I/O connectivity.

List of Features

- **High Performance PCI Express Switch**
  - *Thirty-two 2.5 Gbps PCI Express lanes*
  - *Eight switch ports*
  - *Upstream port configurable up to x8*
  - *Downstream ports configurable up to x8*
  - *Low-latency cut-through switch architecture*
  - *Support for Max Payload Size up to 2048 bytes*
  - *One virtual channel*
  - *Eight traffic classes*
  - *PCI Express Base Specification Revision 1.1 compliant*
- **Flexible Architecture with Numerous Configuration Options**
  - *Automatic per port link width negotiation to x8, x4, x2 or x1*
  - *Automatic lane reversal on all ports*
  - *Automatic polarity inversion on all lanes*
  - *Ability to load device configuration from serial EEPROM*
- **Legacy Support**
  - *PCI compatible INTx emulation*
  - *Bus locking*
- **Highly Integrated Solution**
  - *Requires no external components*
  - *Incorporates on-chip internal memory for packet buffering and queueing*
  - *Integrates thirty-two 2.5 Gbps embedded SerDes with 8B/10B encoder/decoder (no separate transceivers needed)*

Notes

- **Reliability, Availability, and Serviceability (RAS) Features**
  - *Supports ECRC and Advanced Error Reporting*
  - *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 PCI Express Native Hot-Plug, Hot-Swap capable I/O*
  - *Compatible with Hot-Plug I/O expanders used on PC and server motherboards*
- **Power Management**
  - *Utilizes advanced low-power design techniques to achieve low typical power consumption*
  - *Supports PCI Power Management Interface specification (PCI-PM 1.1)*
    - *Supports device power management states: D0, D3$_{hot}$ and D3$_{cold}$*
  - *Unused SerDes are disabled*
- **Testability and Debug Features**
  - *Ability to read and write any internal register via the SMBus*
- **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 31mm x31mm 500-ball BGA with 1 mm ball spacing**

System Diagrams

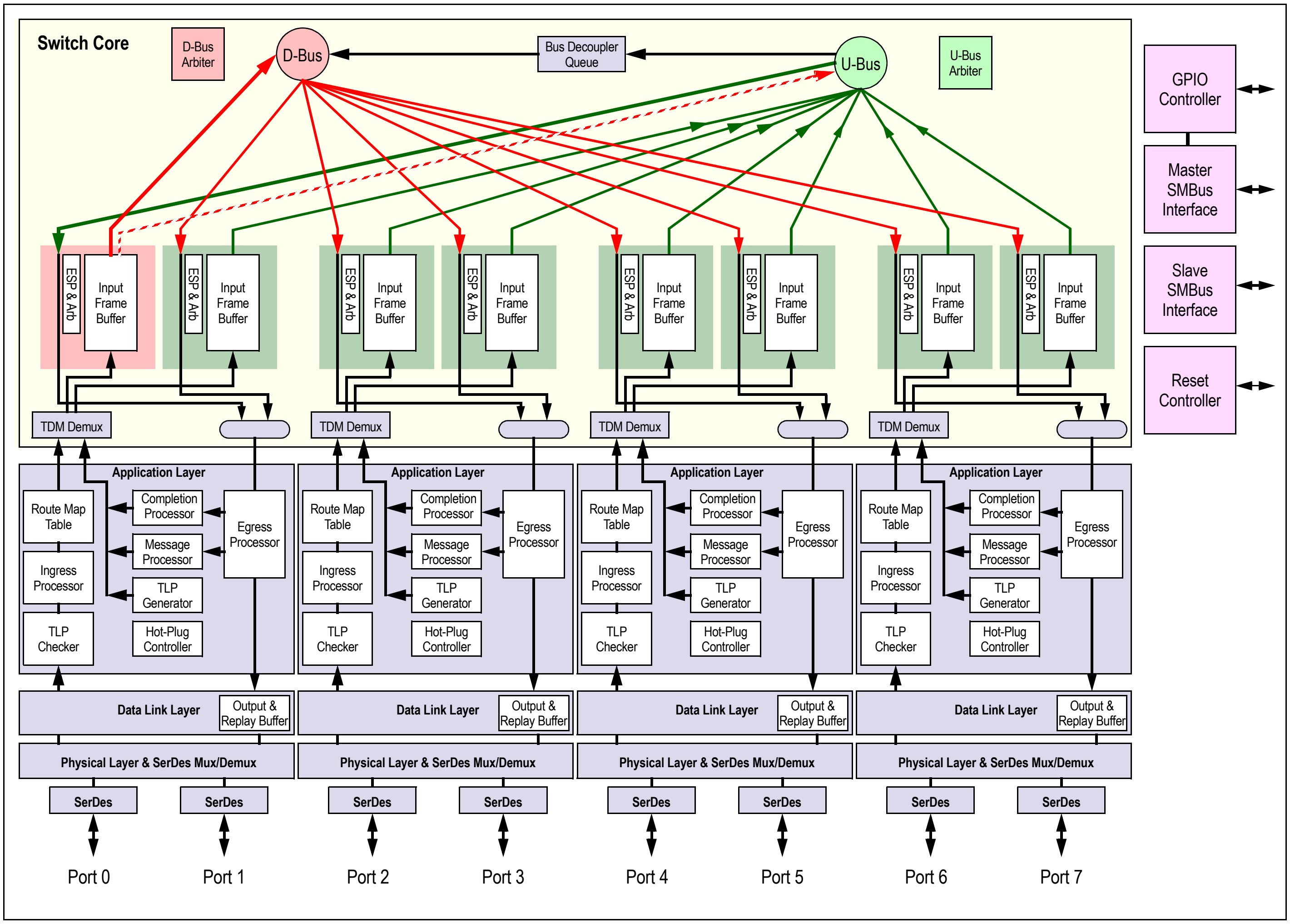


Figure 1.1 PES32T8 Architectural Block Diagram

Logic Diagram

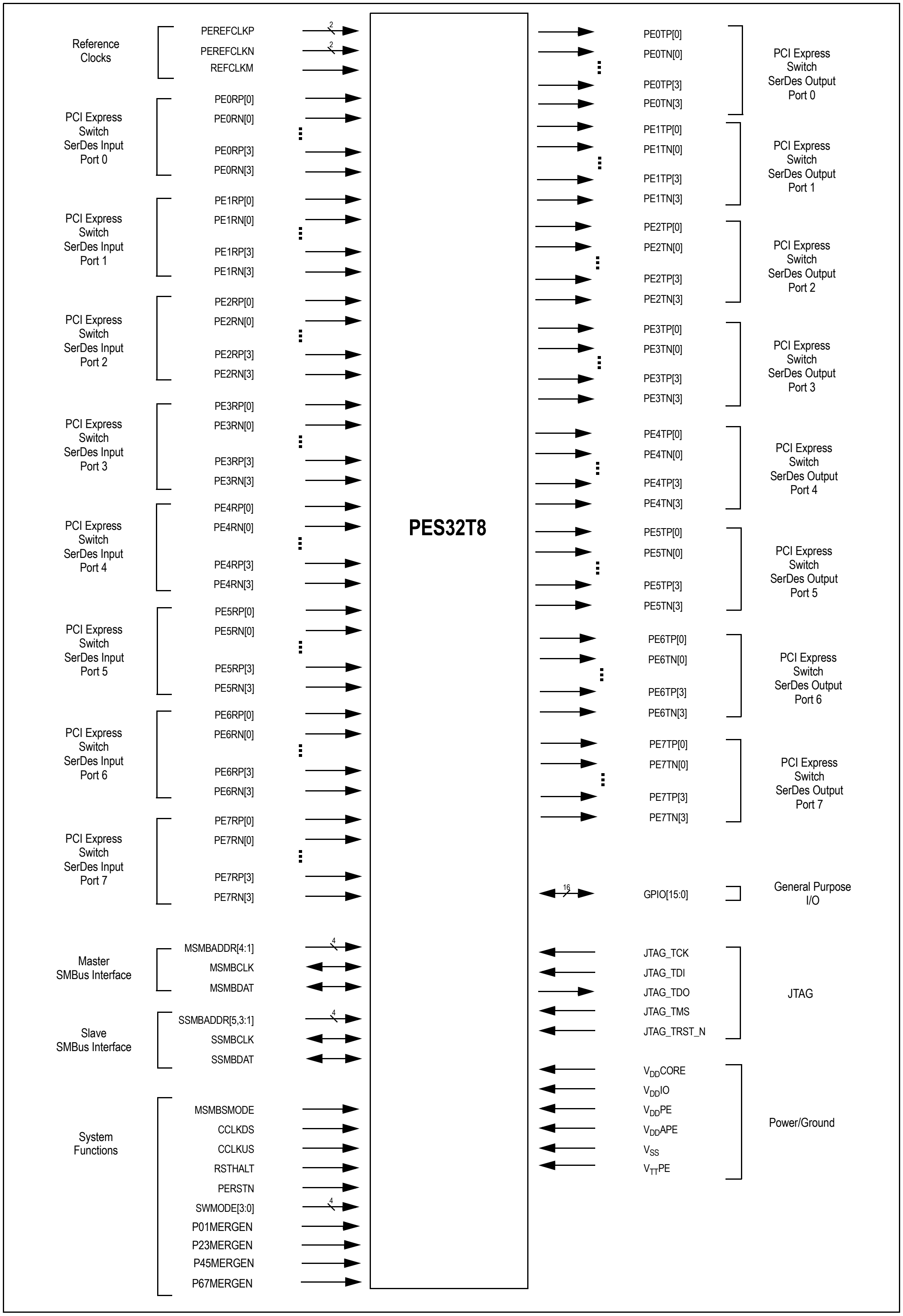


Figure 1.2 PES32T8 Logic Diagram

Notes

System Identification

Vendor ID

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

Device ID

The PES32T8 device ID is shown in Table 1.1.

PCIe Device	Device ID
0x7	0x802F

Table 1.1 PES32T8 Device ID

Revision ID

The PES32T8 revision ID is shown in Table 1.2.

Revision ID	Description
0x0F	Corresponds to ZB silicon
0x0E	Corresponds to ZD silicon
0x0D	Corresponds to ZH silicon

Table 1.2 PES32T8 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.
- Manufacturer ID: 0x33
- LSB: 0x1

SSID/SSVID

The PES32T8 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 PES32T8 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 PES32T8. 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.

Signal	Type	Name/Description
PE0RP[3:0] PE0RN[3:0]	I	**PCI Express Port 0 Serial Data Receive.** Differential PCI Express receive pairs for port 0. Port 0 is the upstream port.
PE0TP[3:0] PE0TN[3:0]	O	**PCI Express Port 0 Serial Data Transmit.** Differential PCI Express transmit pairs for port 0. Port 0 is the upstream port.
PE1RP[3:0] PE1RN[3:0]	I	**PCI Express Port 1 Serial Data Receive.** Differential PCI Express receive pairs for port 1. When port 0 is merged with port 1, these signals become port 0 receive pairs for lanes 4 through 7.
PE1TP[3:0] PE1TN[3:0]	O	**PCI Express Port 1 Serial Data Transmit.** Differential PCI Express transmit pairs for port 1. When port 0 is merged with port 1, these signals become port 0 transmit pairs for lanes 4 through 7.
PE2RP[3:0] PE2RN[3:0]	I	**PCI Express Port 2 Serial Data Receive.** Differential PCI Express receive pairs for port 2.
PE2TP[3:0] PE2TN[3:0]	O	**PCI Express Port 2 Serial Data Transmit.** Differential PCI Express transmit pairs for port 2.
PE3RP[3:0] PE3RN[3:0]	I	**PCI Express Port 3 Serial Data Receive.** Differential PCI Express receive pairs for port 3. When port 2 is merged with port 3, these signals become port 2 receive pairs for lanes 4 through 7.
PE3TP[3:0] PE3TN[3:0]	O	**PCI Express Port 3 Serial Data Transmit.** Differential PCI Express transmit pairs for port 3. When port 2 is merged with port 3, these signals become port 2 transmit pairs for lanes 4 through 7.
PE4RP[3:0] PE4RN[3:0]	I	**PCI Express Port 4 Serial Data Receive.** Differential PCI Express receive pairs for port 4.
PE4TP[3:0] PE4TN[3:0]	O	**PCI Express Port 4 Serial Data Transmit.** Differential PCI Express transmit pairs for port 4.
PE5RP[3:0] PE5RN[3:0]	I	**PCI Express Port 5 Serial Data Receive.** Differential PCI Express receive pairs for port 5. When port 4 is merged with port 5, these signals become port 4 receive pairs for lanes 4 through 7.
PE5TP[3:0] PE5TN[3:0]	O	**PCI Express Port 5 Serial Data Transmit.** Differential PCI Express transmit pairs for port 5. When port 4 is merged with port 5, these signals become port 4 transmit pairs for lanes 4 through 7.
PE6RP[3:0] PE6RN[3:0]	I	**PCI Express Port 6 Serial Data Receive.** Differential PCI Express receive pairs for port 6.
PE6TP[3:0] PE6TN[3:0]	O	**PCI Express Port 6 Serial Data Transmit.** Differential PCI Express transmit pairs for port 6.
PE7RP[3:0] PE7RN[3:0]	I	**PCI Express Port 7 Serial Data Receive.** Differential PCI Express receive pairs for port 7. When port 6 is merged with port 7, these signals become port 6 receive pairs for lanes 4 through 7.

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

Notes

Signal	Type	Name/Description
PE7TP[3:0] PE7TN[3:0]	O	**PCI Express Port 7 Serial Data Transmit.** Differential PCI Express transmit pairs for port 7. When port 6 is merged with port 7, these signals become port 6 transmit pairs for lanes 4 through 7.
PEREFCLKP[2:1] PEREFCLKN[2:1]	I	**PCI Express Reference Clock.** 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.
REFCLKM	I	**PCI Express Reference Clock Mode Select.** This signal selects the frequency of the reference clock input. 0x0 - 100 MHz 0x1 - 125 MHz

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

Signal	Type	Name/Description
MSMBADDR[4:1]	I	**Master SMBus Address.** These pins determine the SMBus address of the serial EEPROM from which configuration information is loaded.
MSMBCLK	I/O	**Master SMBus Clock.** 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	**Master SMBus Data.** This bidirectional signal is used for data on the master SMBus.
SSMBADDR[5,3:1]	I	**Slave SMBus Address.** These pins determine the SMBus address to which the slave SMBus interface responds.
SSMBCLK	I/O	**Slave SMBus Clock.** This bidirectional signal is used to synchronize transfers on the slave SMBus.
SSMBDAT	I/O	**Slave SMBus Data.** 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	**General Purpose I/O.** 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[1]	I/O	**General Purpose I/O.** 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[2]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN0 Alternate function pin type: Input Alternate function: I/O Expander interrupt 0 input

Table 1.5 General Purpose I/O Pins (Part 1 of 3)

Notes

Signal	Type	Name/Description
GPIO[3]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN1 Alternate function pin type: Input Alternate function: I/O Expander interrupt 1 input
GPIO[4]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN2 Alternate function pin type: Input Alternate function: I/O Expander interrupt 2 input
GPIO[5]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: IOEXPINTN3 Alternate function pin type: Input Alternate function: I/O Expander interrupt 3 input
GPIO[6]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: Reserved Alternate function pin type: Input Alternate function: Reserved
GPIO[7]	I/O	**General Purpose I/O.** 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[8]	I/O	**General Purpose I/O.** 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[9]	I/O	**General Purpose I/O.** 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
GPIO[10]	I/O	**General Purpose I/O.** 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	**General Purpose I/O.** 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	**General Purpose I/O.** 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

Table 1.5 General Purpose I/O Pins (Part 2 of 3)

Notes

Signal	Type	Name/Description
GPIO[13]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin.
GPIO[14]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin.
GPIO[15]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin.

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

Signal	Type	Name/Description
CCLKDS	I	**Common Clock Downstream.** 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	**Common Clock Upstream.** 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	**Master SMBus Slow Mode.** 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.
P01MERGEN	I	**Port 0 and 1 Merge.** P01MERGEN is an active low signal. It is pulled low internally via a 251K ohm resistor. When this pin is low, port 0 is merged with port 1 to form a single x8 port. The Serdes lanes associated with port 1 become lanes 4 through 7 of port 0. When this pin is high, port 0 and port 1 are not merged, and each operates as a single x4 port
P23MERGEN	I	**Port 2 and 3 Merge.** P23MERGEN is an active low signal. It is pulled low internally via a 251K ohm resistor. When this pin is low, port 2 is merged with port 3 to form a single x8 port. The Serdes lanes associated with port 3 become lanes 4 through 7 of port 2. When this pin is high, port 2 and port 3 are not merged, and each operates as a single x4 port.
P45MERGEN	I	**Port 4 and 5 Merge.** P45MERGEN is an active low signal. It is pulled low internally via a 251K ohm resistor. When this pin is low, port 4 is merged with port 5 to form a single x8 port. The Serdes lanes associated with port 5 become lanes 4 through 7 of port 4. When this pin is high, port 4 and port 5 are not merged, and each operates as a single x4 port.
P67MERGEN	I	**Port 6 and 7 Merge.** P67MERGEN is an active low signal. It is pulled low internally via a 251K ohm resistor. When this pin is low, port 6 is merged with port 7 to form a single x8 port. The Serdes lanes associated with port 7 become lanes 4 through 7 of port 6. When this pin is high, port 6 and port 7 are not merged, and each operates as a single x4 port.

Table 1.6 System Pins (Part 1 of 2)

Notes

Signal	Type	Name/Description
PERSTN	I	**Fundamental Reset.** Assertion of this signal resets all logic inside the PES32T8 and initiates a PCI Express fundamental reset.
RSTHALT	I	**Reset Halt.** When this signal is asserted during a PCI Express fundamental reset, the PES32T8 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 SWCTL register by an SMBus master.
SWMODE[3:0]	I	**Switch Mode.** These configuration pins determine the PES32T8 switch operating mode. These pins should be static and not change after the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 - through 0xF Reserved

Table 1.6 System Pins (Part 2 of 2)

Signal	Type	Name/Description
JTAG_TCK	I	**JTAG Clock**. 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	**JTAG Data Input**. This is the serial data input to the boundary scan logic or JTAG Controller.
JTAG_TDO	O	**JTAG Data Output**. 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	**JTAG Mode**. The value on this signal controls the test mode select of the boundary scan logic or JTAG Controller.
JTAG_TRST_N	I	**JTAG Reset**. 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_{DD}CORE	I	**Core VDD.** Power supply for core logic.
V_{DD}IO	I	**I/O VDD.** LVTTL I/O buffer power supply.
V_{DD}PE	I	**PCI Express Digital Power.** PCI Express digital power used by the digital power of the SerDes.
V_{DD}APE	I	**PCI Express Analog Power.** PCI Express analog power used by the PLL and bias generator.
V_{TT}PE	I	**PCI Express Serial Data Transmit Termination Voltage.**
V_{SS}	I	**Ground.**

Table 1.8 Power and Ground Pins

Notes

Pin Characteristics

Note: Some input pads of the PES32T8 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[1]	Notes
PCI Express Interface	PE0RN[3:0]	I	CML	Serial Link		
	PE0RP[3:0]	I				
	PE0TN[3:0]	O				
	PE0TP[3:0]	O				
	PE1RN[3:0]	I				
	PE1RP[3:0]	I				
	PE1TN[3:0]	O				
	PE1TP[3:0]	O				
	PE2RN[3:0]	I				
	PE2RP[3:0]	I				
	PE2TN[3:0]	O				
	PE2TP[3:0]	O				
	PE3RN[3:0]	I				
	PE3RP[3:0]	I				
	PE3TN[3:0]	O				
	PE3TP[3:0]	O				
	PE4RN[3:0]	I				
	PE4RP[3:0]	I				
	PE4TN[3:0]	O				
	PE4TP[3:0]	O				
	PE5RN[3:0]	I				
	PE5RP[3:0]	I				
	PE5TN[3:0]	O				
	PE5TP[3:0]	O				
	PE6RN[3:0]	I				
	PE6RP[3:0]	I				
	PE6TN[3:0]	O				
	PE6TP[3:0]	O				
	PE7RN[3:0]	I				
	PE7RP[3:0]	I				
	PE7TN[3:0]	O				
	PE7TP[3:0]	O				

Table 1.9 Pin Characteristics (Part 1 of 2)

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor[1]	Notes
PCI Express Interface (cont.)	PEREFCLKN[2:1]	I	LVPECL/ CML	Diff. Clock Input		Refer to Table 9 in the PES32T8 Data Sheet
	PEREFCLKP[2:1]	I				
	REFCLKM	I	LVTTL	Input	pull-down	
SMBus	MSMBADDR[4:1]	I	LVTTL	Input	pull-up	
	MSMBCLK	I/O		STI[2]		pull-up on board
	MSMBDAT	I/O		STI		pull-up on board
	SSMBADDR[5,3:1]	I		Input	pull-up	
	SSMBCLK	I/O		STI		pull-up on board
	SSMBDAT	I/O		STI		pull-up on board
General Purpose I/O	GPIO[15:0]	I/O	LVTTL	High Drive	pull-up	
System Pins	CCLKDS	I	LVTTL	Input	pull-up	
	CCLKUS	I			pull-up	
	MSMBSMODE	I			pull-down	
	P01MERGEN	I			pull-down	
	P23MERGEN	I			pull-down	
	P45MERGEN	I			pull-down	
	P67MERGEN	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 2 of 2)

[1] Internal resistor values under typical operating conditions are 54K Ω for pull-up and 251K Ω for pull-down.

[2] Schmitt Trigger Input (STI).

Port Configuration

The PES32T8 contains eight ports labeled 0 through 7. Each port is 4 lanes wide. Port 0 is always the upstream port and ports 1 through 7 are always downstream ports. An even port *n* and its odd counterpart, port *n+1,* may be merged into a single x8 port. When this occurs, port *n* is said to be a merged port. When an even port *n* and its odd counterpart, port *n+1*, operate independently, then ports *n* and *n+1* are said to be unmerged. The PES32T8 supports port merging in a static manner during a fundamental reset. If the Port *x* and *y* Merge (P*xy*MERGEN) signal is asserted, then the two x4 ports *x* and *y* are merged into a single x8 merged port called port *x*.

When ports *x* and *y* are merged, the switch port (y), the PCI-to-PCI bridge between x and y, and all resources associated with port *y* are disabled and the following modifications are made to the default PES32T8 configuration.

– *All of the output signals associated with port y remain in a negated state (e.g., hot-plug outputs, link status signals, port reset output, etc.)*

– *All input signals associated with port y are ignored by the PES32T8 and have no effect on its operation.*

– *Configuration read or write transactions to device y on the PES32T8's virtual PCI bus are treated by the upstream port (port 0) as an unsupported request (i.e., the device no longer exists).*
  - This renders the registers in port y's configuration space inaccessible to the root.

– *All registers associated with port y become inaccessible via the SMBus. Reading or writing an inaccessible register has an undefined effect.*
  - Reading a port *y* register returns an undefined value and writing a port *y* register has an undefined effect.

– *All of the SerDes lanes associated with port y become part of port x and are managed by port x as native SerDes lanes (i.e., port x operates as though it were a x8 port).*

– *The initial value of the MAXLNKWDTH field in port x's PCIELCAP register defaults to x8 mode.*

Figures 1.3 through 1.5 illustrate three of many possible PES32T8 configurations. In Figure 1.3, all of the ports are unmerged. In this configuration, the PES32T8 operates as an 8-port switch with all ports having a x4 width. In Figure 1.4 port zero and one are merged as are ports four and five. In this configuration, the PES32T8 operates as a 6-port switch with some having a x4 width and others having a x8 width. Finally, Figure 1.5 illustrates a PES32T8 configuration in which all ports are merged. In this configuration, the PES32T8 operates as a 4-port switch with all ports having a x8 width.

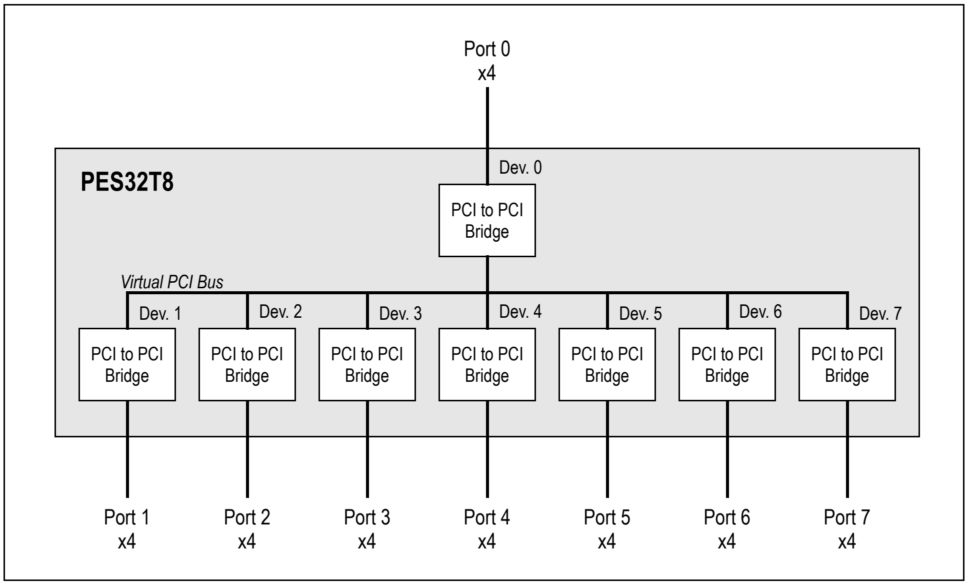


Figure 1.3 All Ports Unmerged Configuration

Notes

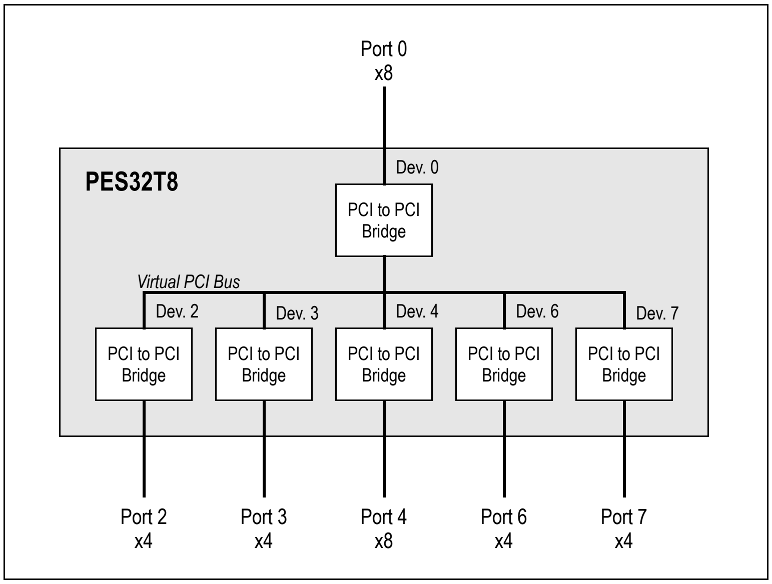


Figure 1.4 Some Ports Merged Configuration

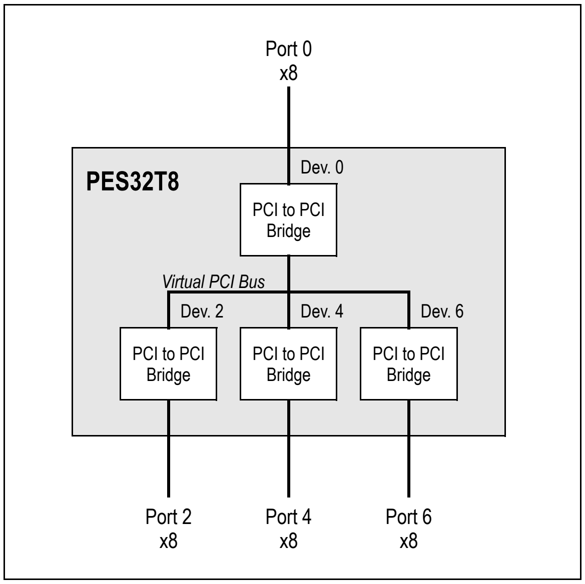


Figure 1.5 All Ports Merged Configuration

Clocking, Reset, and Initialization

Notes

Introduction

The PES32T8 has two differential reference clock inputs that are used internally to generate all of the clocks required by the internal switch logic and the SerDes. While not required, it is recommended that both 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 2.1 Reference Clock Mode Encoding

Each of the reference clock differential inputs feeds several on-chip PLLs. Each PLL generates a 2.5 GHz clock which is used by several SerDes lanes and produces a 250 MHz core clock.

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 2.1 through 2.4 illustrate the operation of the CCLKUS and CCLKDS clocks using a common clock and a non-common clock.

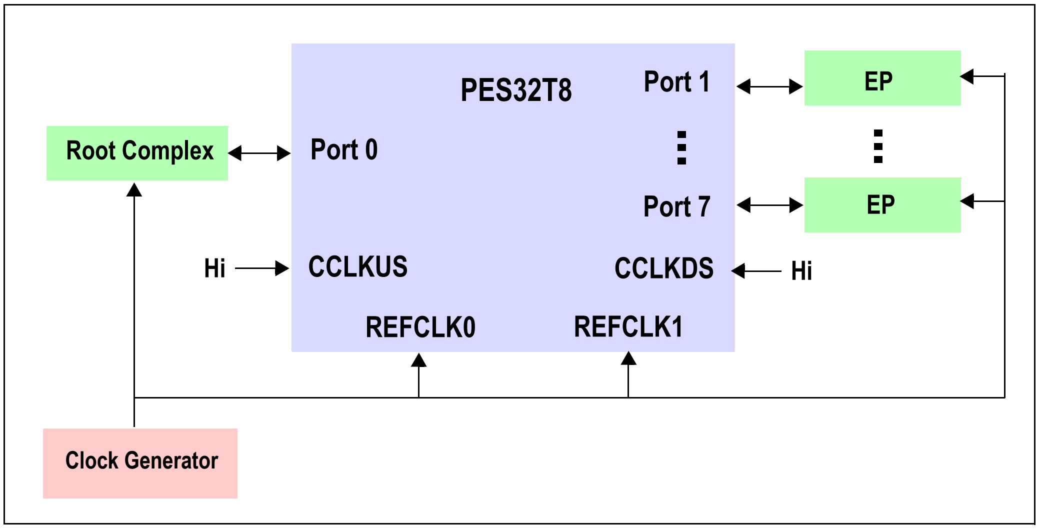


Figure 2.1 Common Clock on Upstream and Downstream (option to enable or disable Spread Spectrum Clock)

Notes

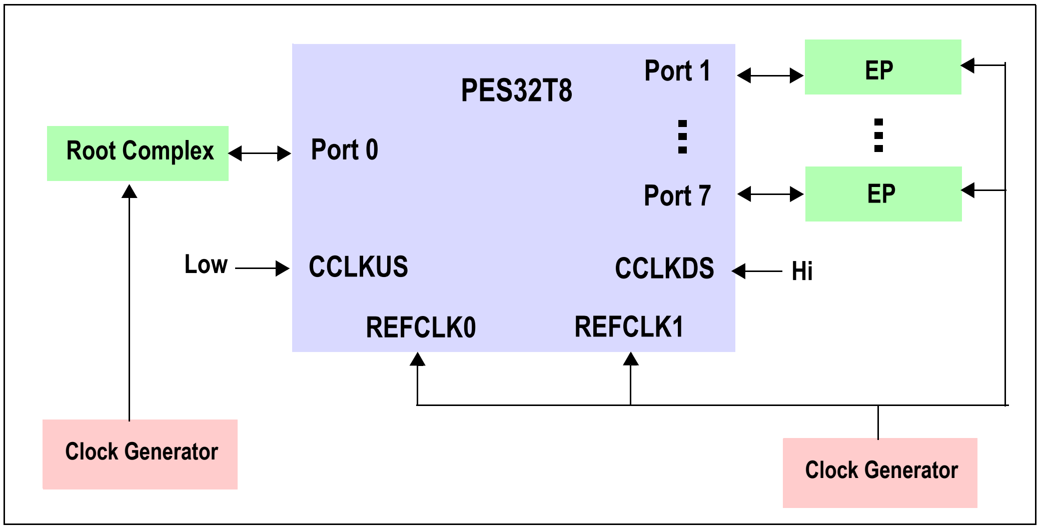


Figure 2.2 Non-Common Clock on Upstream; Common Clock on Downstream (must disable Spread Spectrum Clock)

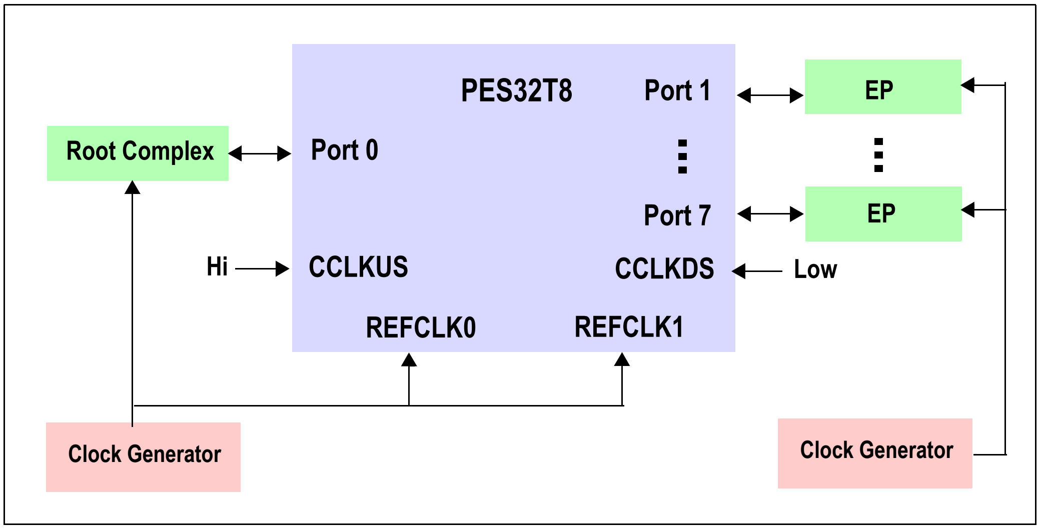


Figure 2.3 Common Clock on Upstream; Non-Common Clock on Downstream (must disable Spread Spectrum Clock)

Notes

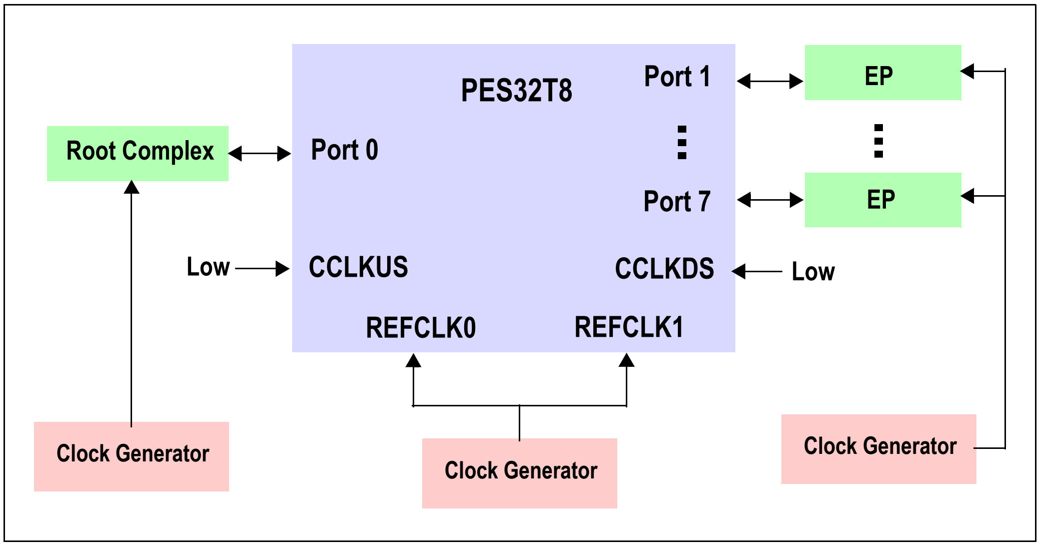


Figure 2.4 Non-Common Clock on Upstream and Downstream (must disable Spread Spectrum Clock)

Initialization

A boot configuration vector consisting of the signals listed in Table 2.2 is sampled by the PES32T8 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 PES32T8.

While basic switch operation may be configured using signals in the boot configuration vector, advanced switch features may 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 2.2. The state of all of the boot configuration signals in Table 2.2 sampled during the most recent cold reset may be determined by reading the SWSTS register.

Signal	Type	Name/Description
CCLKDS	I	**Common Clock Downstream.** 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	**Common Clock Upstream.** 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	**Master SMBus Slow Mode.** 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.
P01MERGEN	I	**Port 0 and 1 Merge.** When this pin is asserted, port 1 is merged with port 0 to form a single x8 port. The SerDes lanes associated with port 1 become lanes 4 through 7 of port 0.

Table 2.2 Boot Configuration Vector Signals (Part 1 of 2)

Notes

Signal	Type	Name/Description
P23MERGEN	I	**Port 2 and 3 Merge.** When this pin is asserted, port 3 is merged with port 2 to form a single x8 port. The SerDes lanes associated with port 3 become lanes 4 through 7 of port 2.
P45MERGEN	I	**Port 4 and 5 Merge.** When this pin is asserted, port 5 is merged with port 4 to form a single x8 port. The SerDes lanes associated with port 5 become lanes 4 through 7 of port 4.
P67MERGEN	I	**Port 6 and 7 Merge.** When this pin is asserted, port 7 is merged with port 6 to form a single x8 port. The SerDes lanes associated with port 7 become lanes 4 through 7 of port 6.
PERSTN	I	**Fundamental Reset.** Assertion of this signal resets all logic inside PES32T8 and initiates a PCI Express fundamental reset.
RSTHALT	I	**Reset Halt.** When this signal is asserted during a PCI Express fundamental reset, PES32T8 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 SWCTL register by an SMBus master.
SWMODE[3:0]	I	**Switch Mode.** These configuration pins determine the PES32T8 switch operating mode. These pins should be static and not change after the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 through 0xF - Reserved

Table 2.2 Boot Configuration Vector Signals (Part 2 of 2)

Reset

The PES32T8 defines four reset categories:

- fundamental reset
- hot reset
- upstream secondary bus reset
- downstream secondary bus reset.

A fundamental reset causes all logic in the PES32T8 to be returned to its initial state. A hot reset causes all logic in the PES32T8 to be returned to its 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).
2. On negation of PERSTN, sample the boot configuration signals listed in Table 2.2. 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. Initialize the PLL and SerDes.
4. Begin link training. While link training is in progress, proceed to step 5.
5. If the Reset Halt (RSTHALT) pin is asserted, set the RSTHALT bit in the SWSTS register.
6. If the switch operating mode is not a test mode, then the reset signal to the PCI Express phy, data link, and transaction layers (stacks) and associated logic is negated, but these stacks 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.

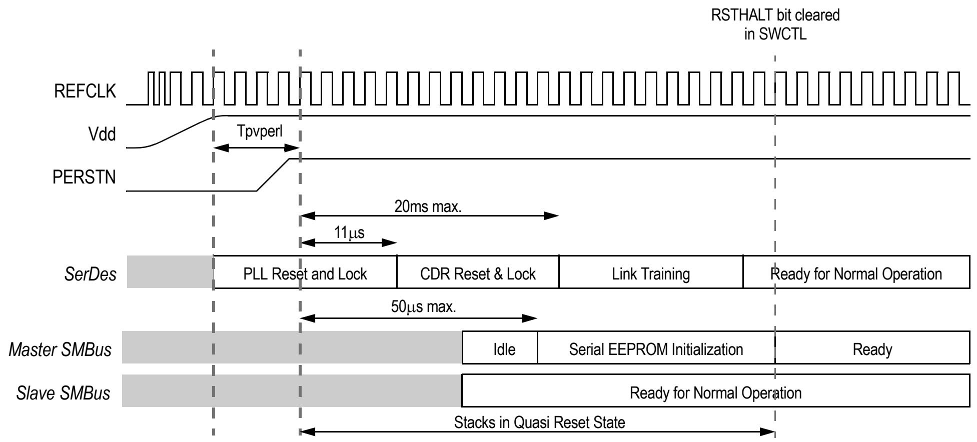
   The state of the MSMBSMODE signal is examined. If it is asserted, then the master SMBus is initialized to operate at 100 KHz rather than 400 KHz.
8. The slave SMBus is taken out of reset and initialized. The slave SMBus address specified by the SSMBADDR[5,3:1] pins is used.
9. The master SMBus is taken 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 PES32T8 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 PES32T8 returning a completion to the requester before the warm reset process begins.

The PES32T8 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 2.5.


RSTHALT bit cleared in SWCTL
REFCLK
Vdd
Tpvperl
PERSTN
20ms max.
11µs
SerDes
PLL Reset and Lock
CDR Reset & Lock
Link Training
Ready for Normal Operation
50µs max.
Master SMBus
Idle
Serial EEPROM Initialization
Ready
Slave SMBus
Ready for Normal Operation
Stacks in Quasi Reset State

Notes:
1) Reference Clock (REFCLK) not shown to scale.
2) The PES32T8 requires a minimum time for Tperst-clk of 1µs. The PES32T8 requires a minimum time for Tpvperl of 1ms.
3) In a system, the values of Tpvperl and Tperst-clk depend on the mechanical form factor in which the PES32T8 is used. For example, the PCIe Card Electromechanical Specification, Revision 2.0, specifies minimum values of Tperst-clk=100µs and Tpvperl=100ms.

Figure 2.5 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 Switch Control (SWCTL) register.

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 PES32T8 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 PES32T8 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 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 PES32T8 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).

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 PES32T8 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 Bridge Control Register (BCTL) 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 PES32T8 are discarded.
3. Wait for the Secondary Bus Reset (SRESET) bit in the Bridge Control Register (BCTL) 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 P7RSTN) 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 PES32T8 ensures through hardware that the minimum PxRSTN assertion pulse width is no less than 200 µS.

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.

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

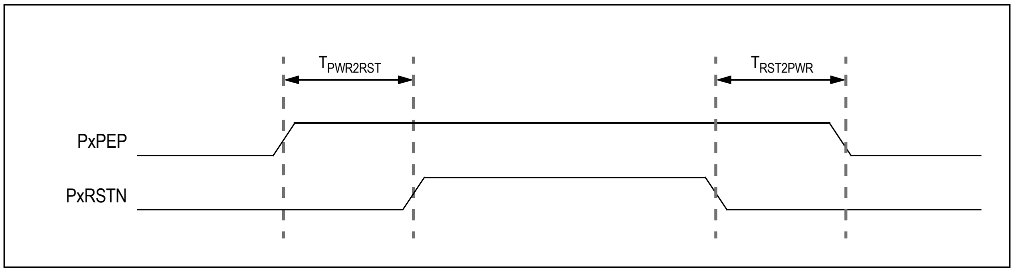


Figure 2.6 Power Enable Controlled Reset Output Mode Operation

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 2.7.

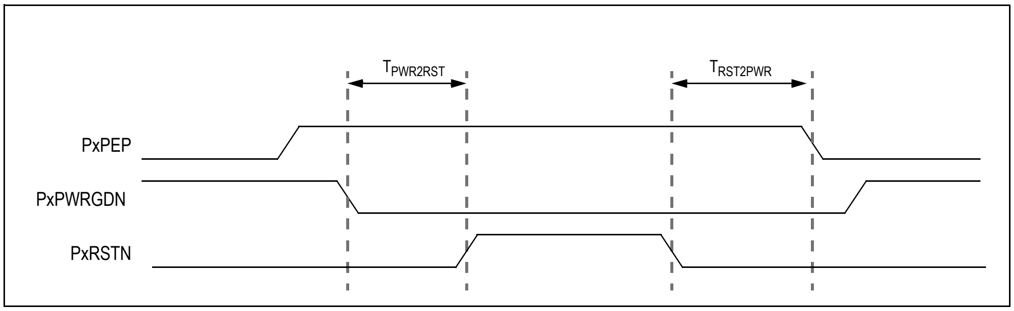


Figure 2.7 Power Good Controlled Reset Output Mode Operation

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.

Notes

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), the corresponding port reset output is immediately asserted. 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 PES32T8.

Theory of Operation

Notes

Introduction

An architectural block diagram of the PES32T8 is shown in Figure 1.1 in Chapter 1. The PES32T8 contains eight x4 ports labeled port 0 through port 7. Port 0 is always the upstream port and ports one through seven are always downstream ports. An even port n and its odd counterpart, port n+1, may be merged into a single x8 port.

At a high level, the PES32T8 consists of four bifurcating PCIe stacks and a switch core. Each bifurcating stack may be configured to operate as a single x8 stack or two x4 stacks. A stack consists of a logic that performs functions associated with the physical, data link, and transactions layers described in the PCIe base 1.1 specification. In addition, a stack performs switch application layer functions such as TLP routing using route map tables, processing configuration read and write requests, etc.

The switch core is responsible for transferring TLPs between stacks. Its main functions are: input buffering, maintaining per-port ingress and egress flow control information, port arbitration, scheduling, and forwarding TLPs between stacks. In typical fan-out applications, all data from downstream ports are destined to memory in the root complex and all TLPs from the root complex are destined to endpoints. Thus, there is no peer-to-peer (i.e., endpoint to endpoint) traffic. Since the PES32T8 is optimized for fan-out applications, its switch core is based on a dual bus architecture.

The downstream bus (D-Bus) is used to transfer TLPs from the upstream port to a downstream port while the upstream bus (U-Bus) is used to transfer TLPs from a downstream port to an upstream port. D-Bus and U-Bus transfers may occur in parallel. While not optimized for peer-to-peer traffic, the PES32T8 supports these transfers. A peer-to-peer transfer occurs by first transferring a TLP from a downstream port into a bus decoupler queue over the U-Bus. Once in the bus decoupler queue, the TLP is transferred to the peer downstream port over the D-Bus. Thus, unlike upstream and downstream traffic which utilize either the U-Bus or D-Bus, peer-to-peer transfers utilize both buses.

The size and constraints of the bus decoupler queue are shown in Table 3.3.

The PES32T8 switch core implements a per-port input buffer called the Input Frame Buffer (IFB). Each input buffer consists of four queues. These queues are the posted transaction queue (posted queue), the non-posted transaction queue (non-posted queue), the completion transaction queue (completion queue), and an insertion buffer to hold TLPs generated by the stack.

The size of each of these queues is shown in Table 3.1. Each queue is implemented as a data queue and a descriptor queue. Thus, there is a limitation on both the amount of data as well as on the number of TLPs that can be stored in a queue. When ports are merged, all the queues are merged[1]. In merged mode, the amount of data as well as the number of TLPs that each queue supports is double that shown in Table 3.1.

IFB Queue	Total Queue Size and Limitations	Advertised Header Credits	Advertised Data Credits
Posted	4 KB or up to 32 TLPs	32	208 (3328 bytes)
Non-posted	768 B or up to 32 TLPs	32	32 (512 bytes)
Completion	4 KB or up to 32 TLPs	32	208 (3328 bytes)

Table 3.1 IFB Buffer Sizes

[1] The insertion buffers are not merged and in all modes the size the insertion buffer is as shown in Table 3.3.

Associated with each port in the data link layer is a shared output and replay buffer. That is, the buffer is partitioned into two sections with a section dedicated to each x4 port in bifurcated mode. This buffer contains TLPs that have been transmitted but have not been acknowledged by the link partner. Space unused to hold replay TLPs is used to provide a per-port output buffer. This output buffer enables switch core transfers to occur at x8 rates even when the corresponding output port has negotiated to a lower link width. As with the IFB, when ports are merged the buffer is merged but the number of descriptors is not increased. The size is shown in Table 3.2.

Buffer	Size and Limitations
Output and Replay Buffer	4 KB of data or up to 32 TLPs
Output and Replay Buffer (merged mode)	8 KB of data or up to 32 TLPs

Table 3.2 PES32T8 Buffer Sizes

The size of the bus decoupler queue and insertion buffer is shown in Table 3.3.

Buffer	Size and Limitations
Bus Decoupler Queue	2 KB of data (i.e., 1 maximum size TLP) or up to 1 TLP (2112 byte total size)
Insertion Buffer	12 Dwords of data (3 or 4 DW of header, 0 or 1 DW of payload data, 0 or 1 DW for ECRC) or up to 2 TLPs (one posted and one completion)

Table 3.3 Bus Decoupler Queue and Insertion Buffer Size

Data Paths

All data paths through the stacks and switch core are 66-bits wide and consists of 64 data bits and two even DWord parity bits. In bifurcated mode, a stack implements two ports and processes ingress port data received from the SerDes in a Time Division Multiplexed (TDM) manner. Even clock cycles are dedicated to the even numbered port and odd clock cycles are dedicated to the odd numbered port. The time division multiplexing of data received from each port's SerDes is performed by the physical layer & SerDes multiplexing/demultiplexing logic.

Since a port may have negotiated to a link width less than x4, not every cycle dedicated to the port may be used. For example, an even numbered port that has negotiated down to a x1 width will transfer a data quantity through its pipeline every eighth clock cycles (i.e., it will utilize one out of every four clock cycles dedicated to that port).

In merged mode, a stack implements one port and allocates all clock cycles to port data received from the SerDes. However, as in bifurcated mode, depending on the negotiated link width not all clock cycles may be used to transfer data. In bifurcated x4 mode, the stack presents port data to the switch core in a TDM manner. The switch core demultiplexes this data and enqueues it into the appropriate queue within the input buffer dedicated to that port. In merged mode the switch core presents port data to the switch core every clock cycle (i.e., the data is not TDMed).

After being queued in an input frame buffer, all data transferred through the switch core is transferred in a continuous TLP manner (i.e., the data path is never multiplexed). Thus, TLPs are transferred at a x8 rate from an input frame buffer, across the appropriate bus or buses, and to an egress stack. Egress data is always presented by the switch core, processed by a stack, and queued in an output and replay buffer in continuous TLP manner (i.e. the data is not TDMed) regardless of the operating mode of the stack. If the stack is in bifurcated mode, data is read from the output and replay buffer by the data link layer in a TDM manner. The physical layer demultiplexes this data and presents it to the appropriate SerDes.

Store-and-Forward vs. Cut-Through Switching and Latency

The PES32T8 utilizes an input buffered cut-through switching architecture to forward PCIe TLPs between switch ports. All TLPs that are received on an ingress link whose width is greater than or equal to the width of the egress link are cut-through the switch. Switch latency is defined as the time from the first TLP symbol being received on the ingress link to the first TLP symbol being transmitted on the egress link.

The best case latency for transactions that can be cut-through is shown in Table 3.4.

Ingress to Egress	Latency (ns)
x8 to x8	184
x8 to x4	184
x8 to x1	184
x4 to x4	204
x4 to x1	204
x1 to x1	272

Table 3.4 Latency

If the ingress link width is less than the egress link width, then an entire TLP must be received before it can be transmitted on the switch egress port. This is necessary since once a TLP transmission begins, it must complete uninterrupted at the rate of the egress port. Thus, when the negotiated ingress link width is less than the egress link width, the PES8T5 operates in a store and forward manner. Ingress TLPs are queued in the IFB until the entire TLP has been received.

Due to bifurcation of the port 0 stack and the switch core architecture, it is only possible to transfer data between port zero and one in one direction at a time. Furthermore, whenever a transfer is occurring between these ports, no other transfer may occur between a downstream and an upstream port. Thus the performance of port one differs from that of the other downstream ports. To reduce the performance impact of transfers between ports zero and one, TLP transfers between these ports are by default not cut-through (i.e., they use a store-and-forward mechanism). Cut through between these ports may be enabled by resetting the Port 0 and 1 Cut-Through Disable (P01CTDIS) bit in the Switch Cut Through Control (SWCTCTL) register. For additional information, see section Port 1 Performance Characteristics on page 3-19.

Switch Core

A simplified view of the switch core is shown in Figure 3.1. The switch core consists of two buses. The primary purpose of the Upstream Bus (U-Bus) is to route TLPs received on a downstream port to the upstream port while the primary purpose of the Downstream Bus (D-Bus) is route TLPs received on the upstream port to a downstream port.

To facilitate peer-to-peer transactions, a bus decoupler is provided to link the U-Bus to the D-Bus. In addition to providing a data path between the U-Bus and the D-Bus, the bus decoupler provides adequate buffering to accommodate one maximum-sized TLP to allow the U-Bus and D-Bus to operate independently. Thus, the transfer from a downstream port to bus decoupler is independent of the transfer from the bus decoupler to a downstream port. While it may appear that the bus decoupler introduces a store-and-forward architecture for peer-to-peer transfers, this is not the case. Transactions flowing through the bus decoupler may be cut-through and typically add no more than five clock cycles of latency.

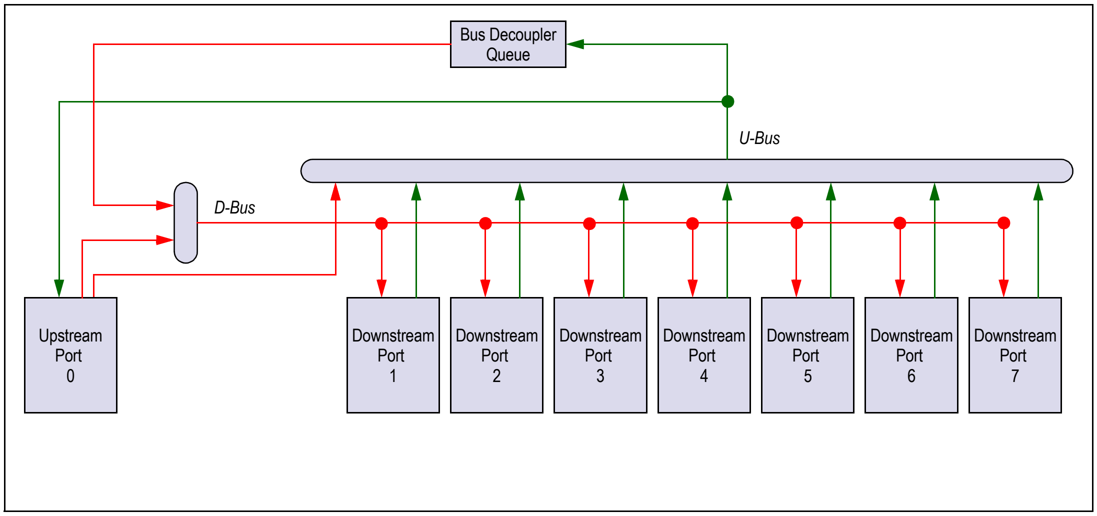


Figure 3.1 Simplified Switch Core U-Bus and D-Bus Datapath

In addition to transactions between the upstream port and downstream port and peer-to-peer transactions, the switch core is responsible for routing of transactions which are destined to the same stack on which the TLP was received. These transactions are referred to as route-to-self.

Transaction Routing

The PES32T8 supports routing of all transaction types defined in the PCIe base 1.1 specification. This includes routing of specification-defined transactions as well as those that may be used in vendor defined messages and in future revisions of the PCIe specification.

Note: The PES32T8 supports routing of trusted configuration transactions.

Specifically, the PES32T8 supports the following type of routing:

- Address routing with 32-bit or 64-bit format
- ID based routing using bus, device and function numbers.
- Implicit routing utilizing
  - Route to root
  - Broadcast from root
  - Local - terminate at receiver
  - Gathered and routed to root

Notes

A summary of TLP types that use the above routing methods is provided in Table 3.5.

Routing Method	TLP Type Using Routing Method
Route by Address	MRd, MrdLk, MWr, IORd, IOWr, Msg, MsgD
ID Based Routing	CfgRd0, CfgWr0, CfgRd1, CfgWr1, TCfgRd, TCfgWr, Cpl, CpdD, CplLk, CplDLk, Msg, MsgD
Imlicit Routing - Route to Root	Msg, MsgD
Implicit Routing - Broadcast from Root	Msg, MsgD
Implicit Routing - Local	Msg, MsgD
Implicit Routing - Gathered and Routed to Root[1]	Only supported for PME_TO_Ack messages in response to a root initiated PME_Turn_Off message.

Table 3.5 Switch Routing Methods

[1] The only Gathered and Routed to Root message supported is a PME_TO_Ack message received on a downstream port.

Transaction Reordering

Each IFB has a free-running timer which is clocked at the 250 MHz core clock frequency. When a TLP is enters the IFB, a time-stamp of when the TLP arrived is stored in a descriptor associated with the TLP. This time-stamp is used to implement a switch time-out and to provide a relative order of TLPs in the IFB.

The IFB contains four input queues per port. These queues are the posted transaction queue (posted queue), the non-posted transaction queue (non-posted queue), the completion transaction queue (completion queue) and an insertion buffer to hold TLPs generated by the stack.

While there are four physical queues in the IFB, TLPs in the insertion buffer are either posted or completion TLPs. Using the time-stamp with each TLP, the IFB logically merges the head of the completion queue with the appropriate TLP type queue. Thus, the IFB has three logical queue heads corresponding to posted, non-posted and completion TLPs.

The IFB examines information associated with the heads of the three logical queues, and presents a signal called “valid” to the switch core for each TLP that may be dispatched. Along with the “valid” signal, the IFB indicates the destination port, payload size, and relative age of the TLPs. This information is used by the switch core scheduling and port arbitration logic to select a TLP to transfer through the switch core.

The generation of “valid” signals is based on PCIe ordering rules and is summarized Table 3.6. The notation x > y indicates that the TLP of type x is older (i.e., has an older time-stamp) than the TLP of type y. It is impossible for two TLPs to have the same timestamp. The notation x.ro indicates that the relaxed ordering attribute is set in the header of the TLP at the head of logical queue type x. Table 3.6 only lists the relaxed ordering attributes in cases where it affects the state of a valid signal.

Logical Queue Head Ordering	Posted Valid	Non-Posted Valid	Completion Valid
P > NP > CP.ro	1	0	1
P > NP > CP	1	0	0
P > CP > NP	1	0	0
P > CP.ro > NP	1	1	0
NP > P > CP	1	1	0

Table 3.6 IFB Transaction Ordering (Part 1 of 2)

Notes

Logical Queue Head Ordering	Posted Valid	Non-Posted Valid	Completion Valid
NP > P > CP.ro	1	1	1
NP > CP > P	1	1	1
CP > P > NP	1	0	0
CP > NP > P	1	1	1

Table 3.6 IFB Transaction Ordering (Part 2 of 2)

When two logical TLP types are destined to the same egress port from the same ingress port, the relative TLP type age is used to order the TLPs (i.e., the older one is allowed to progress first). Other than the ordering rules shown in Table 3.6, relative age plays no role in ordering of TLPs destined to different egress ports from the same ingress ports. Thus, TLPs destined to different egress ports may be aggressively reordered even when there is no congestion in the system.

Since ordering is performed by examining the heads of the three logical IFB queues, TLPs of a particular type are never reordered (i.e., a posted will never bypass another posted with an earlier timestamp).

When the Disable Relaxed Ordering (DRO) bit is set in the Switch Control (SWCTL) register, all of the IFBs in the PES32T8 strongly order transactions regardless of the state of the relaxed ordering attribute.

Scheduling and Port Arbitration

Associated with each port is an Egress Selection Picker (ESP) and associated with each bus (i.e., U-Bus or D-Bus) is a bus arbiter. The function of the ESP is to provide a candidate vector with one bit per port indicating which ports have a TLP in their input frame buffer or insertion buffer that can be transferred to that output port.

In producing the candidate vector, each port's ESP takes the following factors into consideration.

- The availability and ordering, as reported by each port's Input Frame Buffer (IFB), of TLPs of each type (i.e., posted, non-posted, completion, and insertion) that can be transferred from the head of the port's IFB queues.
- The size of each TLP type that can be transferred from each port's IFB queues.
- The amount of space available in the corresponding (i.e., the port with which the ESP is associated) port's output and replay buffer.
- The ability of the application layer in the corresponding port to accept a TLP not destined to that port's egress (e.g., one that is processed by the completion processor).
- The number of PCIe header and data credits available of each TLP type indicated by the corresponding port's link partner.
- The occupancy of the bus decoupler queue.

The candidate vector produced by each port's ESP is presented to the U-Bus and D-Bus arbiters.

- For downstream ports:
  - The upstream portion of the candidate vector is provided to the D-Bus arbiter.
  - The downstream portion of the candidate vector is provided to the U-Bus arbiter. An assertion in this portion of the candidate vector indicates a peer-to-peer or downstream route-to-self transfer.
- For the upstream port:
  - The entire candidate vector is provided to the U-Bus arbiter.

The U-Bus and D-Bus arbiters select the transaction that will occur on the corresponding bus and initiate the transfer. U-Bus and D-Bus transactions are independent of each other and proceed in parallel.

U-Bus Arbiter

The function of the U-Bus arbiter is to select the transaction that will take place on the U-Bus. Arbitration of the U-Bus proceeds in parallel with transfers on the bus. U-Bus arbitration takes at most 2 clock cycles and in most cases is completely overlapped with U-Bus transfers. In the worst case of back-to-back 3

Dword TLPs, the maximum overhead introduced by the U-Bus arbiter is one clock cycle per TLP. U-Bus arbitration logically occurs in two stages. In the first stage, a transfer class is selected, and in the second stage, the actual transfer within that transfer class is selected.

There are four U-Bus transfer classes. The transfer classes are: downstream-to-upstream, peer-to-peer, upstream route-to-self, and downstream route-to-self.

– Downstream-to-upstream transfers occur from a downstream port, through the U-Bus multiplexor to the upstream port. The requests for this transfer class come from the candidate vector produced by the upstream port's ESP.

– Peer-to-peer transfers occur from a downstream port, through the U-Bus multiplexor, and to the bus decoupler queue. The requests for this transfer class come from the downstream portion of the candidate vector produced by each downstream ports' ESP.

– Upstream route-to-self transfers occur from the upstream port insertion buffer output, through the U-Bus multiplexor, and to the upstream port. The request for this transfer class comes from the candidate vector produced by the upstream port's ESP.

– Downstream route-to-self transfers occur from a downstream port, through the U-Bus multiplexor, and to the bus decoupler queue. The requests for this transfer class come from the candidate vector produced by the downstream port's ESP.

A weighted round robin arbitration scheme is used to select a U-Bus transfer class. The percentage of transfers on the U-Bus allocated to a transfer class is controlled by fields in the U-Bus Arbiter Transaction Count (UARBTC) register and U-Bus Arbiter Current Transaction Count (UARBCTC) register.

There are two 8-bit fields in these registers associated with each transfer class. The first field is a transfer count field that indicates how many transfers of that class can occur in an arbitration period. The second field is a current transfer count field that indicates how many transfers from that class are remaining in the current arbitration period.

At the start of each arbitration period, all of the current transfer count fields are initialized to their corresponding transfer count field values. During each arbitration period, the U-Bus arbiter selects, in a fair manner, a transfer class which (a) is requesting service and (b) has the corresponding current transfer count field equal to non-zero. If no such transfer class exists, then the arbitration period is said to have ended, and each current transfer count field is re-initialized with the corresponding transfer count field value.

An arbitration period may end due to all current transaction count registers being zero or because there are no transfer classes with a non-zero current transaction count requesting service. In either case, there is no overhead introduced by the end of an arbitration period (i.e., no clock cycles are added to the arbitration). Once a transfer class has been selected, the U-Bus arbiter selects a transaction within that transfer to initiate on the U-Bus. The arbitration algorithm used to select this transaction is dependent on the selected transaction class.

For downstream-to-upstream transfers, the upstream port's port arbiter selects the transaction that is initiated. The upstream port arbiter implements both a hardwired fixed round robin algorithm as well as a weighted round robin with 32 phases algorithm as defined by the PCIe base 1.1 specification. The arbitration algorithm, as well as weighted round robin arbitration parameters, are software selectable.

– Weighted round robin arbitration with 32-phases is implemented by converting the PCIe port arbitration table into weighted round robin weights. Therefore, over short intervals grants may not match the phase table configuration.

For peer-to-peer transfers, the U-Bus employs a hardwired fair arbitration algorithm that selects the transaction that is initiated. This algorithm ensures fair arbitration among each downstream candidate vector bit that selects a peer-to-peer transfer.

No arbitration is necessary for upstream-to-self transfers since there is only one choice. For downstream-to-self transfers, the U-Bus employs a hardwired fair arbitration algorithm that selects among the downstream ports requesting service.

Figure 3.2 summarizes the two stage U-Bus arbitration process.

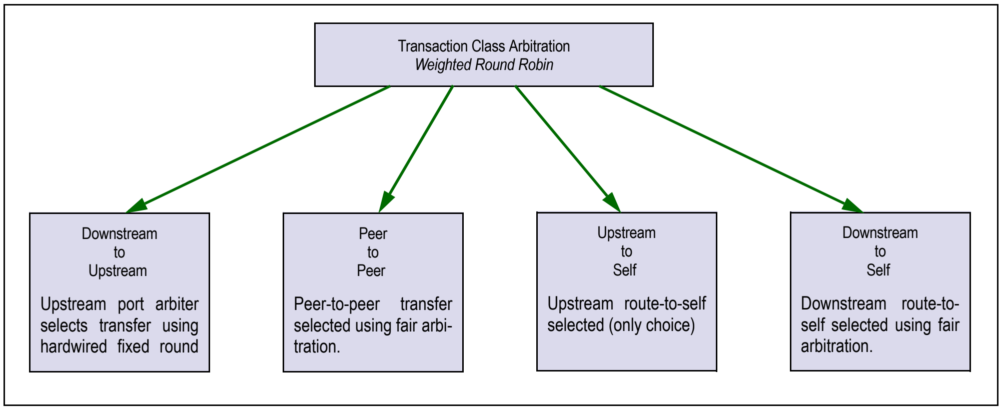

Transaction Class Arbitration
Weighted Round Robin
Downstream to Upstream
Upstream port arbiter selects transfer using hardwired fixed round
Peer to Peer
Peer-to-peer transfer selected using fair arbitration.
Upstream to Self
Upstream route-to-self selected (only choice)
Downstream to Self
Downstream route-to-self selected using fair arbitration.

Figure 3.2 U-Bus Arbitration

D-Bus Arbiter

D-Bus arbitration is similar to U-Bus arbitration but logically occurs in only a single stage. Arbitration of the D-Bus proceeds in parallel with transfers on the bus. D-Bus arbitration takes at most 2 clock cycles and in most cases is completely overlapped with D-Bus transfers. In the worst case of back-to-back 3 Dword TLPs, the maximum overhead introduced by the D-Bus arbiter is one clock cycle per TLP.

There are two D-Bus transfer classes. They are upstream-to-downstream and bus decoupler queue transfers. Upstream-to-downstream transfers proceed from the upstream port, through the D-Bus multiplexor, and to a downstream port. The requests for this transfer comes from the upstream portion of the candidate vector produced by the downstream ports.

Bus decoupler queue transfers proceed from the bus decoupler queue, through the D-Bus multiplexor, and to a downstream port. The queue contains a peer-to-peer or downstream route-to-self transaction performed on the U-Bus. The request for this transfer comes from the bus decoupler queue (i.e., when the bus decoupler queue indicates that there is a transaction in the queue).

A weighted round robin arbitration scheme, identical to the one used by the U-Bus arbiter, is used to select the D-Bus transfer class.

The percentage of transfers on the D-Bus allocated to a transfer class are controlled by fields in the D-Bus Arbiter Transaction Count (DARBTC) and D-Bus Arbiter Current Transaction Count (DARBCTC) register. Once a transaction class is selected, the D-Bus arbiter selects a transaction within that transfer class to initiate on the D-Bus. Unlike U-Bus arbitration, once the transaction class has been selected, there is typically no choice regarding which transfer to perform (i.e., the upstream to downstream or bus decoupler queue transfer is initiated). However, since the upstream input frame buffer has a queue per transaction type, it is possible for multiple upstream to downstream transactions to simultaneously request service. In such a situation, the oldest transaction (i.e., the one with the oldest time-stamp) is selected.

Peer-to-Peer Transactions

The broadest definition of a peer-to-peer transaction is a transaction that originates at one endpoint and targets another endpoint (i.e., the endpoints are peers). In the context of the PES32T8, transactions between downstream ports are referred to as peer-to-peer transactions. However, depending on system topology and configuration, transactions between the upstream port and a downstream port may also be peer-to-peer (i.e., between endpoints). While it is understood that such transactions may be peer-to-peer, in the context of the PES32T8 these are not considered peer-to-peer transactions.

Thus, a peer-to-peer transaction in this specification refers to a transaction which originates at a PES32T8 downstream port and targets another PES32T8 downstream port. While the architecture of the PES32T8 supports peer-to-peer transactions, the performance of these transactions has not been optimized since in fan-out applications peer-to-peer transactions are rare.

Notes

Some systems view peer-to-peer transactions as a potential security vulnerability since this capability allows one endpoint to modify the state of another without protection checks. When the Disable Peer-to-Peer (DP2P) bit is set in the Switch Control (SWCTL) register, all PES32T8 peer-to-peer transactions are disabled. In this mode, any transaction received at a PES32T8 downstream port that targets another downstream port is treated as an unsupported request (UR). Enabling this mode may cause TLPs with legal PCI system architecture routing to be terminated with an error. Thus, it is the responsibility of the system designer and system software to ensure proper system operation in this mode.

The PES32T8 also supports selective disabling of peer-to-peer transactions in a matrix fashion. Associated with each port is TLP Routing Control (Px_TROUTECTL) register. Each bit in the Port Routing Disable (PRDIS) field in this register corresponds to a PES32T8 port (e.g., bit one corresponds to port one). When a bit in this field is set, forwarding of TLPs is disabled to the corresponding port from port in which the TROUTECTL register is located. TLPs with a disabled route are treated as an unsupported request (UR).

- Route-to-self can never be disabled.
- Routing to the upstream port from a downstream port cannot be disabled.
- Routing to a downstream port from the upstream port cannot be disabled.

Bus Locking

The PES32T8 supports locked transactions, allowing legacy software to run without modification on PCIe. Only one locked transaction sequence may be in progress at a time.

- A locked transaction sequence is requested by the root by issuing a Memory Read Request - Locked (MRdLk) transaction. A lock is established when a lock request is successfully completed with a Completion with Data - Locked (CplDLk). A lock is released with an Unlock message (Msg).

When the PES32T8 receives a MRdLk transaction on its upstream port destined for a downstream port, it forwards the MRdLK transaction to the downstream port and locks the downstream port so that all subsequent TLPs destined to that downstream port from ports other than the root are blocked until the lock is released.

- The MRdLK transaction obeys PCI ordering rules, meaning that all queued posted requests for the downstream port are completed prior to the MRdLK being transmitted. The MRdLK is allowed by bypass queued non-posted requests and completions.
- When only the downstream port is locked, no transactions destined to any other port are blocked (e.g., transactions from the other downstream ports to the upstream port and peer-to-peer transactions are not blocked)

When a CplDLk is returned by the locked downstream port, the upstream port becomes locked causing all transactions destined to the upstream port from sources other than the locked downstream port (i.e., other downstream ports) to be blocked. If the lock is unsuccessful, then a CPlLk is returned by the downstream port. Regardless of the success of a lock, the root complex is required to terminate all lock sequences with an Unlock message. The upstream port lock associated with an unsuccessful completion is released when this Unlock message is received.

The CplDLk transaction obeys PCI ordering rules, meaning that all queued posted requests at the locked downstream port destined to the upstream port are completed prior to the CplDLk being transmitted. The CplDLk is allowed to bypass queued non-posted requests and completions. When a CplDLk is returned by the locked downstream port and the upstream port becomes locked, the entire switch becomes locked. This means that all transactions from the other downstream ports destined to the upstream port or the locked downstream port are blocked until the switch is unlocked.

While the switch is locked:

- any register in the switch may be read or written via the SMBus
- it is illegal to read or write any of the PCIe configuration space headers in the switch since the switch can not generate a completion until the switch is unlocked.

The behavior of the switch is undefined when:

- a PCIe configuration space register is read while the switch is locked
- any transaction other than a MWr, MRdLk, and Unlock message is received on the upstream port when the switch is locked
- any transaction other than a CplLk and a CplDLk is received on the locked downstream port when the switch is locked.

While the switch is locked:

- it is possible for the root to perform subsequent reads from the locked device by issuing a MRdLk requests to the locked device and receiving a CplDLk or CplLk response from the locked device. These transactions do not change the state of the switch when the switch is locked. Therefore, a CplLk completion once the switch is locked in no way “unlocks” the switch.
- it is possible for the root to perform subsequent writes to the locked device by issuing MWr requests to the locked device. These transactions in no way change the state of the switch when the switch is locked.
- the upstream port and the locked downstream port may generate messages (i.e., “insert messages”). These messages include interrupt emulation messages and error messages. A locked switch port may also generate MSIs.

When an Unlock message is received on the upstream port, the switch is unlocked. This causes the Unlock message to be forwarded to the locked downstream port and the unblocking of transactions destined to the upstream and previously locked downstream port. The unlock message obeys PCI ordering rules, meaning that all queued posted requests from the upstream port are completed prior to the switch becoming unlocked.

When a TLP from a downstream port is blocked from being forwarded due to a locked switch, then the TLP is delayed until the switch is unlocked. If the switch is locked for an extended period, this may cause TLPs to be discarded due to switch time-outs (see section Switch Time-Outs on page 3-16).

The behavior of the switch is undefined when:

- a MRdLk TLP is received on the upstream port destined to an unlocked downstream port while the switch is locked
- the upstream port is locked with a downstream port and a TLP is received by the upstream port destined to an unlocked downstream port.

The locked status of the switch may be determined by examining the Lock Mode (LOCKMODE) bit in the SWSTS register. This register is meant to be read via the slave SMBus interface.

Port Interrupts

The upstream port, port 0, does not generate legacy interrupts or MSIs. Downstream ports support generation of legacy interrupts and MSIs. The following are sources of downstream port interrupts and MSIs.

- Downstream port’s hot-plug controller
- Link bandwidth notification capability (i.e., assertion of the LBWSTS or LABWSTS bits in the PCIELSTS register when interrupt notification is enabled for these bits)

When a downstream port is configured to generate INTx messages, only INTA is used. When an unmasked interrupt condition occurs, then an MSI or interrupt message is generated by the corresponding port as described in Table 3.7. The removal of the interrupt condition occurs when unmasked status bit(s) causing the interrupt are masked or cleared.

The PES32T8 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.

Unmasked Interrupt	EN bit in MSICAP Register	INTXD bit in PCICMD Register	Actions
Asserted	1	X	MSI message generated
	0	0	Assert_INTA message request generated to switch core
	0	1	None
Negated	1	X	None
	0	0	Deassert_INTA message request generated to switch core
	0	1	None

Table 3.7 Downstream Port Interrupts

Legacy Interrupt Emulation

The PES32T8 supports legacy PCI INTx emulation. Rather than use sideband INTx signals, PCIe defines two messages that indicate the assertion and negation of an interrupt signal. An Assert_INTx message is used to signal the assertion of an interrupt signal and an Deassert_INTx message is used to signal its negation.

The PES32T8 maintains an aggregated INTx state for each of the four interrupt signals (i.e., A through D) at each port.

- The value of the INTA, INTB, INTC and INTD aggregated state for the entire switch may be determined by examining the corresponding field in the upstream port's Interrupt Status (P0_INTSTS) register.
- The aggregated INTx state for a downstream port may be determined by reading the corresponding field in the port's Interrupt Status (Px_INTSTS) register. This register contains the aggregated state of interrupts generated by that port (i.e., hot-plug) plus interrupt messages received from the downstream link partner. The interrupt state reflects the state of interrupts as seen by that port (i.e., before downstream port interrupts are mapped to upstream port interrupts).

An Assert_INTx message is sent to the root by the upstream port (i.e., port 0), when the aggregated state of the corresponding interrupt in the upstream port transitions from a negated to an asserted state. A Deassert_INTx message is sent to the root by the upstream port when the aggregated state of the corresponding interrupt in the upstream port transitions from an asserted to a negated state.

PCI to PCI bridges must map interrupts on the secondary side of the bridge according to the device number of the device on the secondary side of the bridge. No mapping is performed for the PCI to PCI bridges corresponding to downstream ports as these ports only connect to device zero. A mapping is performed for the upstream port (i.e., port 0). This mapping for the PES32T8 is summarized in Table 3.8.

	Upstream Port Interrupt (Port 0)			
	INTA	**INTB**	**INTC**	**INTD**
Downstream Port[1] Interrupt	Port 1 INTD	Port 1 INTA	Port 1 INTB	Port 1 INTC
	Port 2 INTC	Port 2 INTD	Port 2 INTA	Port 2 INTB
	Port 3 INTB	Port 3 INTC	Port 3 INTD	Port 3 INTA
	Port 4 INTA	Port 4 INTB	Port 4 INTC	Port 4 INTD
	Port 5 INTD	Port 5 INTA	Port 5 INTB	Port 5 INTC
	Port 6 INTC	Port 6 INTD	Port 6 INTA	Port 6 INTB
	Port 7 INTB	Port 7 INTC	Port 7 INTD	Port 7 INTA

Table 3.8 PES32T8 Downstream to Upstream Port Interrupt Routing

[1] Port X INTy corresponds to external downstream generated INTy interrupts and INTy interrupts generated by the port

If a Downstream Port goes to DL_Down status, the INTx virtual wires associated with that port are negated, and the upstream port's aggregate sate is updated accordingly. This may result in the upstream port generating a Deassert_Intx message.

Standard PCIe Error Detection and Handling

This section describes standard PCIe error detection and handling as prescribed by the PCIe base 1.1 specification.

Physical Layer Errors

Table 3.9 lists error checks performed by the physical layer and action taken when an error is detected.

Error Condition	PCIe Base 1.1 Specification Section	Action Taken
Invalid symbol or running disparity error detected.	4.2.1.3	Correctable error processing
Any TLP or DLLP framing rule violation.	4.2.2.1	Correctable error processing
8b/10b decode error	4.2.4.4	Correctable error processing
Any violation of the link initialization or training protocol	4.2.4	Uncorrectable error processing

Table 3.9 Physical Layer Errors

Data Link Layer Errors

Table 3.10 lists error checks performed by the data link layer and action taken when an error is detected.

Notes

Error Condition	PCIe Base 1.1 Specification Section	Action Taken
TLP ending in ENDB with LCRC that does not match inverted calculated LCRC	3.5.3.1	TLP discarded
TLP received with incorrect LCRC	3.5.3.1	Correctable error processing
TLP received with sequence number not equal to NEXT_RCV_SEQ and this is not a duplicate TLP	3.5.3.1	Correctable error processing
Bad DLLP[1]	3.5.2.1	Correctable error processing
Replay time-out	3.5.2.1	Correctable error processing
REPLAY NUM rollover	3.5.2.1	Correctable error processing
Violation of flow control initialization protocol	3.3.1	Uncorrectable error processing
Sequence number specified by AckNak_Seq does not correspond to an unacknowledged TLP or to the value in ACKD_SEQ	3.5.2.1	Uncorrectable error processing
Non-blocked surprise down as defined in 3.2.1	3.5.2.1 & 3.2.1	If checking is enabled, fatal error processing

Table 3.10 Data Link Layer Errors

[1] A bad DLLP is a DLP with a bad LCRC.

Transaction Layer Errors

The PES32T8 transaction layer functions are performed by the application layer. Table 3.11 lists error checks performed by the transaction link layer and action taken when an error is detected.

Error Condition	PCIe Base 1.1 Specification Section	Action Taken
Poisoned TLP received	2.7.2.2	For the non-advisory cases: non-fatal error processing. Advisory cases: correctable error processing. TLP header logged in AER.
ECRC check failure	2.7.1	For the non-advisory cases: non-fatal error processing. Advisory cases when ECRC checking is enabled: non-fatal error processing. TLP header logged in AER.
Unsupported request	Numerous	For the non-advisory cases: non-fatal error processing. Advisory cases: correctable error processing. TLP header logged in AER.

Table 3.11 Transaction Layer Errors (Part 1 of 2)

Notes

Error Condition	PCIe Base 1.1 Specification Section	Action Taken
Completer abort Completion time-out	2.3.1 2.8	Not applicable. The PES32T8 never generates non-posted transactions as a requester.
Unexpected completion	2.3.2	For the non-advisory cases: non-fatal error processing. Advisory cases: correctable error processing. TLP header logged in AER.
Receiver overflow	2.6.1.2	RO bit set in TLSTSE debug register. Fatal error processing. TLP header is not logged in AER.
Flow control protocol error	2.6.1	Not applicable. The PES32T8 does not check for any flow control errors.
Malformed TLP	See Table 3.12 and Table 3.13	Fatal error processing. TLP header logged in AER.

Table 3.11 Transaction Layer Errors (Part 2 of 2)

The PES32T8 supports the following advisory non-fatal error cases as defined by the PCIe base 1.1 specification.

– Non-posted request that is treated as an Unsupported Request (UR).
– Poisoned TLP in which the PES32T8 is not the ultimate destination.
– TLP with an ECRC error in which the PES32T8 is not the ultimate destination.
– Unexpected completion.

When ECRC checking is enabled, the PES32T8 checks the ECRC of all TLPs in which the TLP Digest (TD) bit is set in the TLP header. When an ECRC error is detected for a TLP destined to the PES32T8 as the ultimate destination, then non-fatal error processing is initiated. When an ECRC error is detected for a TLP in which the PES32T8 is not the ultimate destination, then advisory correctable error processing is initiated.

Table 3.12 lists the error checks performed by the ingress transaction layer for malformed TLPs. These TLP error checks are performed when a TLP is received by the switch (i.e., by the stack associated with the port on which the switch receives the TLP).

TLP Type	Error Check
All	TLP must have a valid FMT/TYPE combination. Data payload length ≤ Max_Payload_Size (i.e., MPS field in PCIEDCTL register).
All TLPs with data (i.e., FMT[1]=1)	LENGTH field must match actual payload data.
All TLPs with ECRC (i.e., TD=1)	Actual TLP length must match calculated length (HEADER + PAYLOAD + ECRC).

Table 3.12 Ingress Malformed TLP Error Checks (Part 1 of 2)

TLP Type	Error Check

I/O read or write request	LENGTH = 1 (doubleword)
	TC = 0
	ATTR = 0
	Last DWord BE[3:0] = 0b0000

Configuration read or write request	LENGTH = 1 (doubleword)
	TC = 0
	ATTR = 0
	Last DWord BE[3:0] = 0b0000

Message Requests interrupt message Power management message Error signalling message Unlock message Set power limit message	TC = 0
TLPs with Route to Root Complex routing.	May only be received on downstream ports.
TLPs with Broadcast from Root Complex routing.	May only be received on upstream ports.
TLPs with Gathered and Routed to Root Complex routing	May only be received by the downstream ports. Must be a PME_TO_ACK message (all other TLP types with this routing are illegal).
Interrupt messages (INTx)	May only be received by the downstream ports
Trusted Cfg Request	May only be received on upstream port (refer to PCIe specification 2.0, section 6.12.1).
All	TLP traffic class (TC) must be mapped to VC0. TC to VC mapping is controlled by the TC/VC Map (TCVCMAP) field in the ingress port's VC Resource 0 Control (VCR0CTL) register.

Table 3.12 Ingress Malformed TLP Error Checks (Part 2 of 2)

Table 3.13 lists the error checks performed by the egress transaction layer for malformed TLPs. These error checks are performed when a TLP leaves the switch (i.e., by the stack associated with the port on which the TLP leaves the switch).

TLP Type	Error Check
All	TLP traffic class (TC) must be mapped to VC0. TC to VC mapping is controlled by the TC/VC Map (TCVCMAP) field in the egress port's VC Resource 0 Control (VCR0CTL) register

Table 3.13 Egress Malformed TLP Error Checks

Routing Errors

This section lists TLP routing errors that are detected by the PES32T8. All of these errors are treated as unsupported requests.

Address Routed TLPs

- TLPs whose address decoding indicates they are to route back to the port on which they were received.
- TLPs received on the upstream port that match the upstream port's address range but which do not match a downstream port's address range (i.e., TLPs that do not route through the PES32T8).
- TLPs that target a downstream port that is not enabled for such transactions.
  - For prefetchable memory and non-prefetchable memory transactions, the Memory Access Enable (MAE) bit must be set in the PCI Command (PCICMD) register.
  - For I/O transactions, the I/O Access Enable (IOAE) bit must be set in the PCI Command (PCICMD) register.
- Memory and IO requests from downstream ports that target the upstream port and the Bus Master Enable (BME) bit is cleared in the upstream port's PCICMD register.
- Memory and IO requests received on downstream ports and the Bus Master Enable (BME) bit is cleared in the downstream port's PCICMD register.
- A VGA route from a VGA enabled downstream port.

Configuration Requests (Routed by ID)

- Type 0 requests that arrive on a downstream port.
- Type 1 requests that arrive on a downstream port.
- Type 1 requests that do not route through the upstream port's PCI-to-PCI bridge.
- Type 1 requests that are converted to Type 0 requests at the upstream port but which do not target an enabled downstream port device number (i.e., target a PCI-to-PCI bridge device number that doesn't exist).
- Type 1 requests that route through the PES32T8 target a downstream port's link partner (i.e., are converted to a Type 0 request at the downstream port), and which do not target device zero. Note that this check may be disabled by the DDDNC bit in the SWCTL register. See section SWCTL - Switch Control (0x404) on page 9-51 for more information.

Completions (Routed by ID)

- Completions that attempt to route back onto the link on which they were received.
- Completions that do not have a valid route through the PES32T8.
- All completions that terminate within the PES32T8 (i.e., ones that target the upstream port bus number or any device/function on the virtual PCI bus within the switch) are treated as unexpected completions.

ID Routed Messages

- Messages that attempt to route back onto the link on which they were received.
- Messages that do not have a valid route through the PES32T8.
- Messages that target a downstream port device number that does not exist.
- A non Vendor Defined Type 1 message which targets an enabled PES32T8 port (i.e., PCI-to-PCI bridge). Vendor Defined Type 1 messages received by a PES32T8 port are silently discarded.
- A non Vendor Defined Type 1 message which is received by the upstream port.

Switch Specific Error Detection and Handling

This section describes PES32T8-specific error detection and handling. Since these mechanisms are outside of the PCIe base 1.1 specification, the PES32T8-specific initialization and error handling code may be required to take full advantage of these features.

Switch Time-Outs

The switch core discards any TLP that reaches the head of an IFB queue and is more than 64 seconds old. This includes posted, non-posted, completion and inserted TLPs. Although this feature is enabled by default, it may be disabled by setting the Enable Switch Time-outs (ETO) bit in a port's Switch Time-Out Control (SWTOCTL) register.

Whenever a TLP is discarded by a port due to a switch time-out, a bit corresponding to the type of TLP that was discarded is set in port's Switch Time-Out Status (SWTOSTS) register. In addition, a saturating count field corresponding to the type of TLP that was discarded is incremented in the port's Switch Time-Out Count (SWTOCNT) register. These saturating count fields are atomically cleared when read.

Corresponding to each TLP type that may be discarded in the SWTOSTS register is an associated field in the Switch Time-Out Reporting Control (SWTORCTL) register that controls the manner in which a dropped TLP of that type is reported. Error message reporting due to dropped TLPs is considered an internally generated error message and thus may be masked in the same manner as other internally generated error messages with the SERR Enable (SERRE) bit in the PCI Command (PCICMD) register and the error reporting enables (i.e., CERN, NFEREN, FEREN, and UREREN bits) in the PCI Express Device Control (PCIEDCTL) register.

If error reporting is enabled, an error message is generated when a status bit transitions from a zero to a one (i.e., is set) in the SWTOSTS register. A subsequent error of the same type is not reported until software clears the corresponding status bit and it is again set.

Following a fundamental reset, discarded posted and inserted TLPs are reported with an ERR_NONFATAL message. Discarded non-posted and completion TLPs are not reported by default since the requester's completion timer will detect the loss of a TLP of this type. The default error reporting policy my be modified by the root, serial EEPROM, or slave SMBus master.

If during processing of a TLP with broadcast routing a switch core time-out occurs, then the switch core will abort processing of the TLP. This may result in the broadcast TLP being transmitted on some but not all downstream pots.

End-to-End Parity Checking

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

Since deep sub-micron devices are known to be susceptible to single-event-upsets, a mechanism is desired that detects errors that occur within a PCI express switch. The PES32T8 parity protects all TLPs in the switch, thus enabling corruption that may occur inside of the device to be detected and reported even in systems that do not implement ECRC.

Data flowing into the PES32T8 is protected by the LCRC. Within the Data Link (DL) layer of the switch ingress port, the LCRC is checked and 32-bit DWord even parity is computed on the received TLP data. If an LCRC error is detected at this point, the link level retransmission protocol is used to recover from the error by forcing a retransmission by the link partner. As the TLP flows through the switch, its alignment or contents may be modified. In all such cases, parity is updated and not recomputed. Hence, any error that occurs is propagated and not masked by a parity regeneration. When the TLP reaches the DL layer of the switch egress port, parity is checked and in parallel a LCRC is computed. If the TLP is parity error free, then the LCRC and TLP contents are known to be correct and the LCRC is used to protect the packet through the lower portion of the DL layer, PHY layer, and link transmission.

If a parity error is detected by the DL layer of an egress port, then the TLP is nullified by inverting the computed LCRC and ending the packet with an EDB symbol. Nullified TLPs received by the link-partner are discarded. In addition to nullifying the TLP, the End-to-End Parity Error (EEPE) bit is set in the Switch parity Error Status (SWPESTS) register, and the saturating End-to-End Parity Error Count (EEPEC) field is incremented in the Switch Parity Error Count (SWPECNT) register. The EEPEC field is atomically cleared when read.

All internal memories used to store TLP data within the PES32T8 are Dword even parity protected. Parity errors in these memories are propagated and reported using the end-to-end parity protection mechanism.

The End-to-End Parity Error Reporting (EEPE) field in the Switch Parity Error Reporting Control (SWPERCTL) register controls the manner in which end-to-end parity errors are reported. If error reporting is enabled, an error message is generated when the EEPE bit in the SWPESTS register transitions from a zero to a one. A subsequent error of the same type is not reported until software clears the corresponding status bit and it is again set. End-to-end parity checking at a port may be disabled by setting the Disable End-to-End Parity Checking (DEEPC) bit in the Switch Parity Error Control (SWPECTL) register.

Parity generation can never be disabled at a port. However, to facilitate parity generation and checking testing a mechanism exists to generate bad parity. When the Generate Bad End-to-End Parity (GBEEP) bit is set in the SWPECTL register, bad (i.e., inverted) parity is generated for all Dwords of a TLP when the length field in the TLP header matches the value of the Length (LENGTH) field in the SWPECTL register. TLPs whose header length field does not match the LENGTH field are passed to the switch core with correct parity.

Following a fundamental reset, end-to-end parity checking is enabled and errors are reported with an ERR_NONFATAL message to the root. The default error reporting policy may be modified by the root, serial EEPROM, or slave SMBus master.

In addition to TLPs that flow through the switch, cases exist in which TLPs are produced and consumed by the switch (e.g., a configuration requests and responses). Whenever a TLP is produced by the switch, parity is computed as the TLP is generated. Thus, error protection is provided on produced TLPs as they flow through the switch. In addition, parity is checked on all consumed TLPs. If an error is detected, the TLP is discarded and an error is reported using the mechanism described above.

This means that a parity error reported at a switch port cannot be definitively used to identify the location at which the error occurred as the error may have occurred when parity as generated at another port, in the switch core, or may have been generated locally (i.e., for ingress TLPs to the switch core which are consumed by the port such as Type 0 configuration read requests on the upstream port).

TLP Processing

The PES32T8 supports two forms of very basic processing on TLPs that flow through the switch. Since TLP processing modifies the contents of a TLP, it may not be used in systems that employ ECRC since ECRC is not recomputed after TLP modifications. When the Force Relaxed Ordering (FRO) bit is set in the TLP Processing Control (TLPPCTL) register, the value of the relaxed ordering attribute is set to the value dictated by the Relaxed Ordering Modification (ROM) field in the TLPPCTL register. This transformation is only performed on TLPs in which the relaxed ordering attribute is applicable.

The relaxed ordering attribute is applicable to all TLPs except: configuration requests, I/O requests, memory requests that are Message Signaled Interrupts (MSIs), and Message requests (except where specifically permitted). Since MSIs cannot be distinguished from memory write transactions by the switch, the relaxed ordering attribute of MSIs will be modified.

When the Force No-Snoop (FNS) bit is set in the TLP Processing Control (TLPPCTL) register, the value of the no-snoop attribute is set to the value dictated by the No-Snoop Modification (NSM) field in the TLPPCTL register. This transformation is only performed on TLPs in which the no-snoop attribute is applicable. The no-snoop attribute is applicable to all TLPs except: configuration requests, I/O requests, memory requests that are Message Signaled Interrupts (MSIs), and Message requests (except where specifically permitted). Since MSIs cannot be distinguished from memory write transactions by the switch, the no-snoop attribute of MSIs will be modified.

Notes

Port 1 Performance Characteristics

As shown in Figure 1.1, port zero and port one share the same bifurcating PCIe stack. Since port zero is an upstream port and port one is a downstream port, the performance characteristics of port one differs from that of the other downstream ports. Unlike other downstream ports, transfers between port zero and port one are by default store-and-forward, not cut-through. This results in the latency of these transfers being proportional to the size of the transferred TLP and significantly higher than that of other ports.

Cut-through between ports zero and one can be enabled at the expense of U-Bus and D-Bus bus bandwidth by resetting the Cut-Through Disable (P01CTDIS) bit in the Switch Cut Through Control (SWCTCTL) register. The performance trade-off is largely dependant on the application. High throughput applications should stay with the default configuration while applications that require low latency should reset the P01CTDIS bit.

When port one is enabled, broadcast transactions are split into two transfers. First broadcast data is transferred between the upstream port and all of the downstream ports except port one. When this completes, broadcast data is transferred between the upstream port and port one. During the transfer between the upstream port and port one, no other transfer can take place on the U-Bus or D-Bus.

When port one is merged with port zero, broadcasts occur in one transfer as would be expected by the architectural description provided in this chapter.

Notes

Link Operation

Notes

Introduction

The PES32T8 contains eight x4 ports which may be merged in pairs to form x8 ports. The default link width of each port is x4 and the SerDes lanes are statically assigned to a port.

Polarity Inversion

Each port of the PES32T8 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 6 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 not to be inverted.

Link Width Negotiation

The PES32T8 supports the optional link variable width negotiation feature outlined in the PCIe specification. During link training, each merged x8 port is capable of negotiating to a x8, x4, x2 or x1 link width and each unmerged 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. 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 merged port link negotiates to a width less than x8, the unused group of four lanes are powered down to save power. In addition, unused SerDes in a four lane group are put in a low power state (i.e. L1 state). When an unmerged 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 merged or unmerged port is disabled, all SerDes lanes associated with that port are powered down.

Lane Reversal

The PCIe specification describes an optional lane reversal feature. The PES32T8 supports the automatic lane reversal feature outlined in the PCIe specification. The operation of lane reversal is dependant on the maximum link width selected by the MAXLNKWDTH field. Lane reversal mapping for the various non-trivial x4 non-merged port maximum link width configurations supported by the PES32T8 are illustrated in Figures 4.1 and 4.2. Lane reversal mapping for the various non-trivial x8 merged port maximum link width configurations supported by the PES32T8 are illustrated in Figures 4.3 through 4.5.

Notes

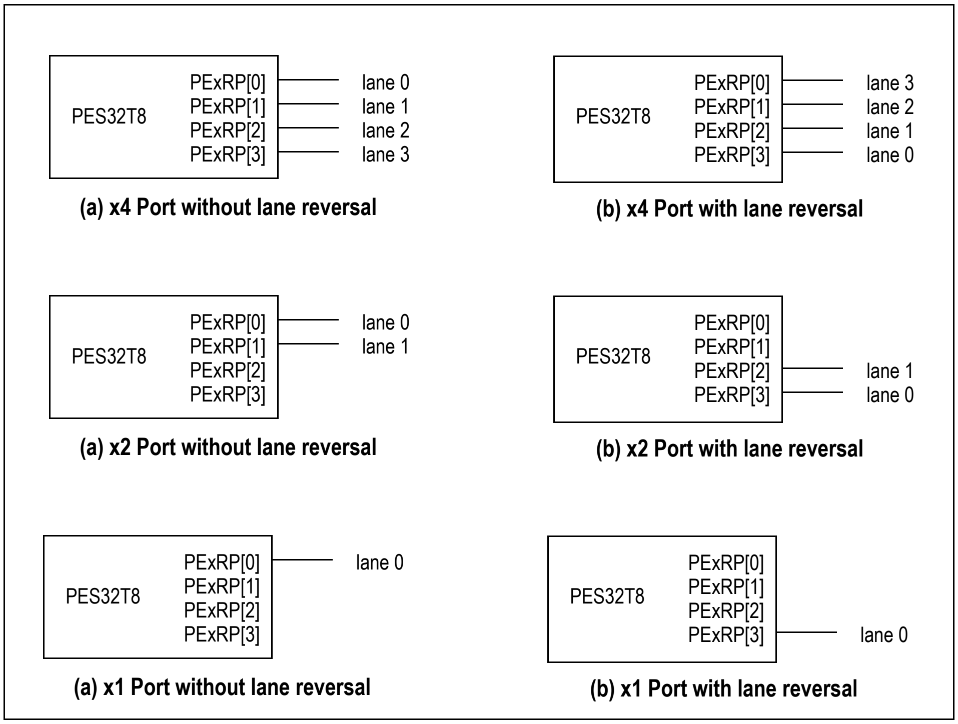

PES32T8
PExRP[0]
PExRP[1]
PExRP[2]
PExRP[3]
lane 0
lane 1
lane 2
lane 3
(a) x4 Port without lane reversal
lane 3
lane 2
lane 1
lane 0
(b) x4 Port with lane reversal
(a) x2 Port without lane reversal
(b) x2 Port with lane reversal
(a) x1 Port without lane reversal
(b) x1 Port with lane reversal

Figure 4.1 Unmerged Port Lane Reversal for Maximum Link Width of x4 (MAXLNKWDTH[5:0]=0x4)

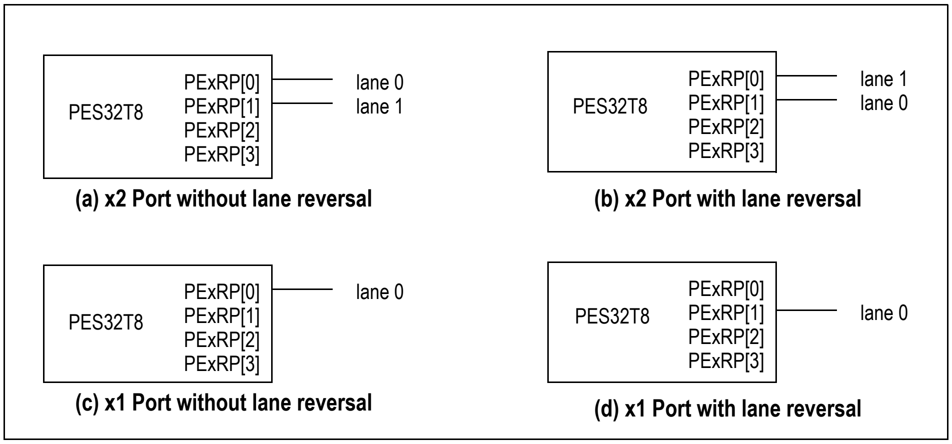

PES32T8
PExRP[0]
PExRP[1]
PExRP[2]
PExRP[3]
lane 0
lane 1
(a) x2 Port without lane reversal
lane 1
lane 0
(b) x2 Port with lane reversal
(c) x1 Port without lane reversal
(d) x1 Port with lane reversal

Figure 4.2 Unmerged Port Lane Reversal for Maximum Link Width of x2 (MAXLNKWDTH[5:0]=0x2)

Notes

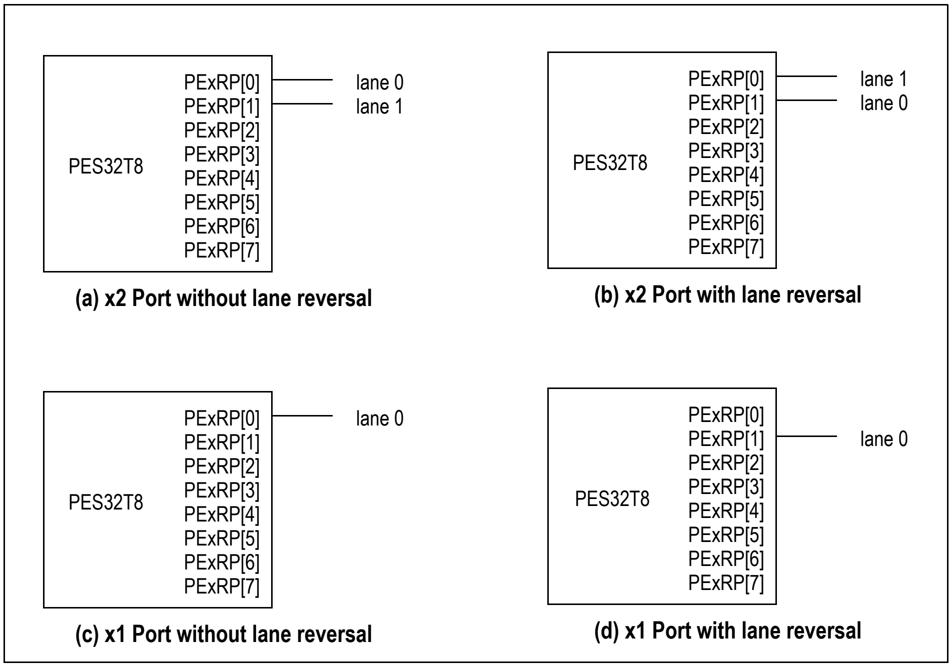


Figure 4.3 Merged Port Lane Reversal for Maximum Link Width of x2 (MAXLNKWDTH[5:0]=0x2)

Notes

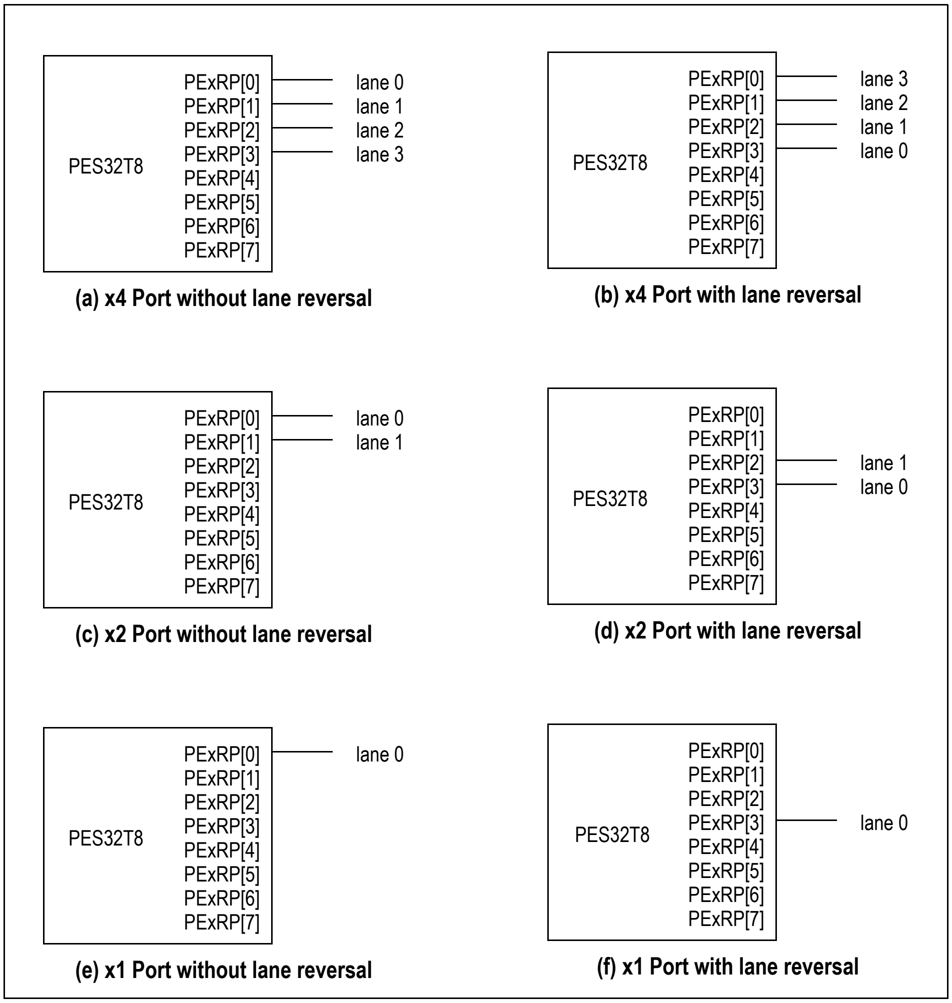
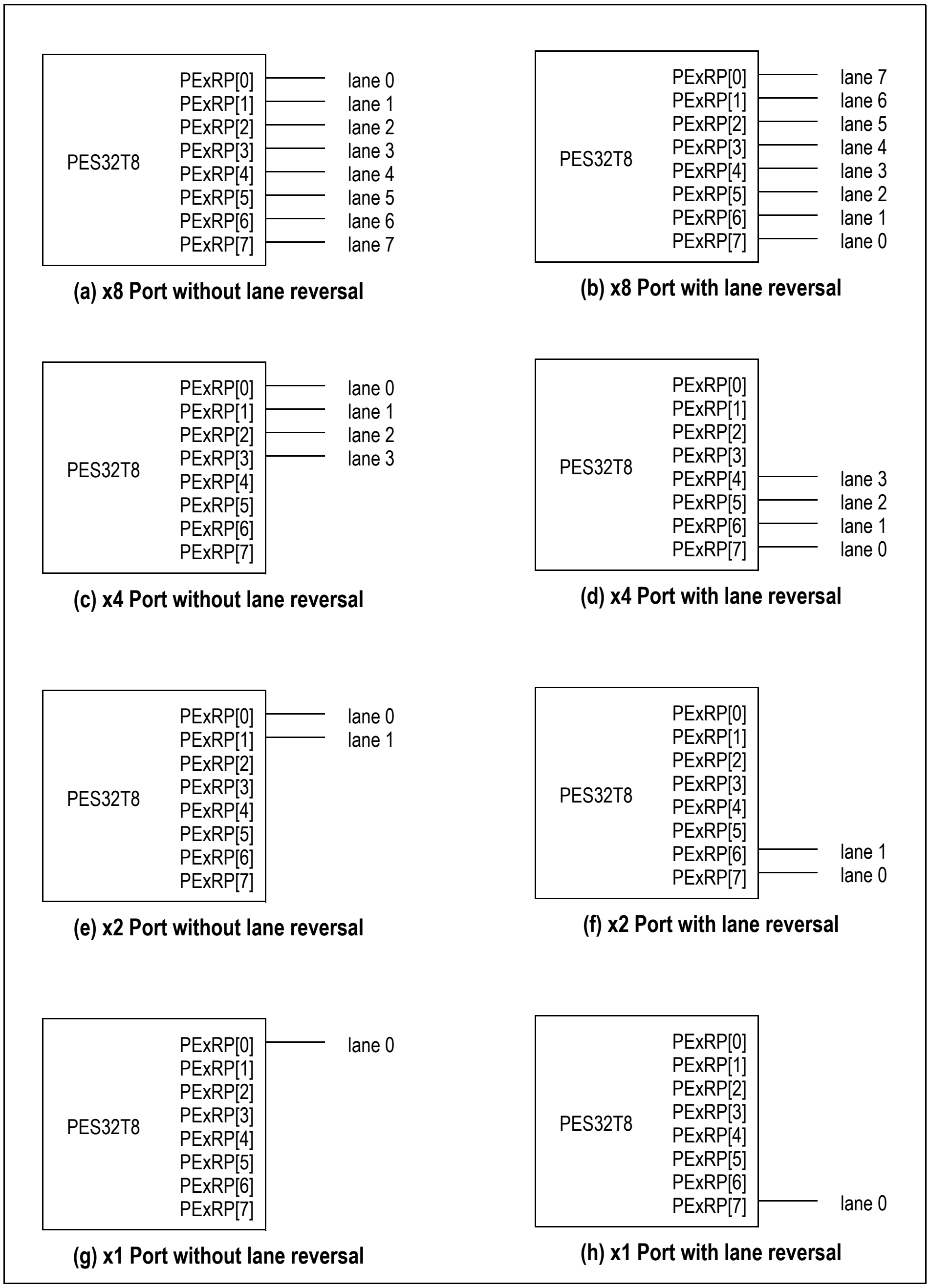


Figure 4.4 Merged Port Lane Reversal for Maximum Link Width of x4 (MAXLNKWDTH[5:0]=0x4)

Notes



(a) x8 Port without lane reversal

(b) x8 Port with lane reversal

(c) x4 Port without lane reversal

(d) x4 Port with lane reversal

(e) x2 Port without lane reversal

(f) x2 Port with lane reversal

(g) x1 Port without lane reversal

(h) x1 Port with lane reversal

Figure 4.5 Merged Port Lane Reversal for Maximum Link Width of x8 (MAXLNKWDTH[5:0]=0x8)

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.

Notes

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 IFB queues are advertised.

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, 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 occurs:

- 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 PES32T8 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 D3hot 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.
- Link Down

  A transitional link down pseudo-state prior to L0. This pseudo-state is associated with the LTSSM Detect, Polling, Configuration, Disabled, Loopback, and Hot-Reset states.

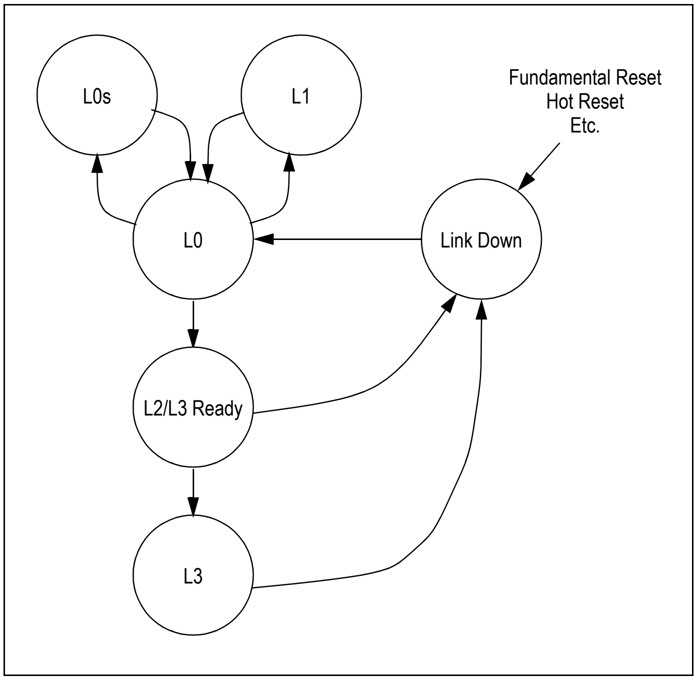


Figure 4.6 PES32T8 ASPM Link Sate Transitions

Active State Power Management

The operation of Active State Power Management (ASPM) is independent of 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 PES32T8 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 upstream switch port has the following L0s entry conditions:

- The receive lanes of all of the switch downstream ports which are not in a low power state (i.e., D3) and whose link is not down are in the L0s state.
- The switch has no TLPs to transmit on the upstream port or there are no available flow control credits to transmit a TLP.
- There are no DLLPs pending for transmission on the upstream port.

The downstream switch ports have the following L0s entry conditions:

- The receive lanes of the switch upstream port are in the L0s state.
- The switch has no TLPs to transmit on the downstream port or there are no available flow control credits to transmit a TLP.
- There are no DLLPs pending for transmission on the downstream port.

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 PES32T8's upstream port. If the link partner acknowledges the transition, the L1 state is entered. Otherwise, the L0s state is entered.

The upstream switch port will only request entry into the L1 state when all of the downstream ports which are not in a low power state (i.e., D3) and whose links are not down are in the L1 state.

Notes

Link Status

Associated with each port is a Port Link Up (PxLINKUP) status output and a Port Activity (PxACTIVE) status output. These outputs are provided on I/O Expander 4. 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 PxLINKUP and PxACTIVE status outputs may be used to provide a visual indication of system state and activity or for debug. The PxLINKUP 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 PxACTIVE output is asserted whenever any TLP, other than a vendor defined message, is transmitted or received on the corresponding port's link. Whenever a PxACTIVE 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.

General Purpose I/O

Notes

Introduction

The PES32T8 has 16 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 determines 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
0	PE2RSTN	Reset output for downstream port 2	Output
1	PE4RSTN	Reset output for downstream port 4	Output
2	IOEXPINTN0	SMBus I/O expander interrupt 0	Input
3	IOEXPINTN1	SMBus I/O expander interrupt 1	Input
4	IOEXPINTN2	SMBus I/O expander interrupt 2	Input
5	IOEXPINTN3	SMBus I/O expander interrupt 3	Input
6	Reserved		Input
7	GPEN	General purpose event output	Output
8	PE1RSTN	Reset output for downstream port 1	Output
9	PE3RSTN	Reset output for downstream port 3	Output
10	PE5RSTN	Reset output for downstream port 5	Output
11	PE6RSTN	Reset output for downstream port 6	Output
12	PE7RSTN	Reset output for downstream port 7	Output

Table 5.1 General Purpose I/O Pin Alternate Function

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.

Note: 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 PES32T8.

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.

Notes

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 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.

Notes

SMBus Interfaces

Notes

Introduction

The PES32T8 contains two SMBus interfaces. The slave SMBus interface provides full access to all software visible registers in the PES32T8, 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.

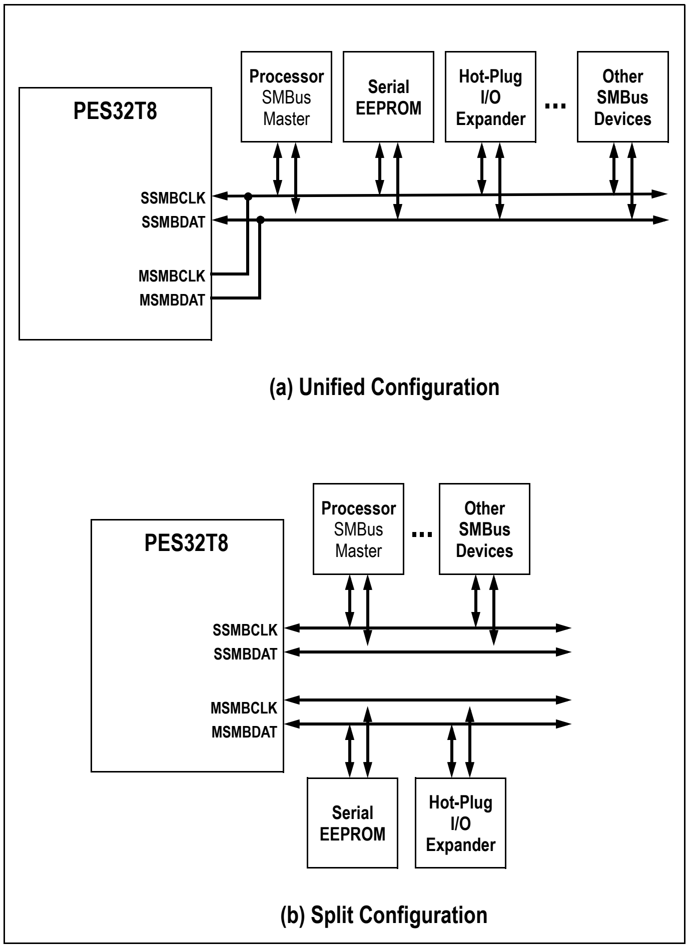


Figure 6.1 SMBus Interface Configuration Examples

In the unified configuration, shown in Figure 6.1(a), the master and slave SMBuses are tied together and the PES32T8 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 PES32T8 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 PES32T8 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. 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 the Fundamental Reset section on page 2-5). During a fundamental reset initialization sequence, the state of the Master SMBus Slow Mode (MSMBSMODE) signal is examined. If this signal is asserted, then the Master SMBus Clock Prescalar (MSMBCP) field in the SMBus Control (SMBUSCTL) register is initialized to support 100 KHz SMBus operation. If the signal is negated, then the MSMBCP 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 PES32T8. Any PES32T8 software visible register in any port may be initialized with values stored in the serial EEPROM.

Each software visible register in the PES32T8 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 PES32T8 are listed in Table 9.1 of Chapter 9. Since configuration blocks are used to store only the value of those registers that are initialized, a serial EEPROM much smaller than the total size of all of the configuration spaces may be used to initialize the device.

Any serial EEPROM compatible with those listed in Table 6.2 may be used to store the PES32T8 initialization values. Some of these devices are larger than the total size of all of the PCI configuration spaces in the PES32T8 that may be initialized and thus may not be fully utilized.

Notes

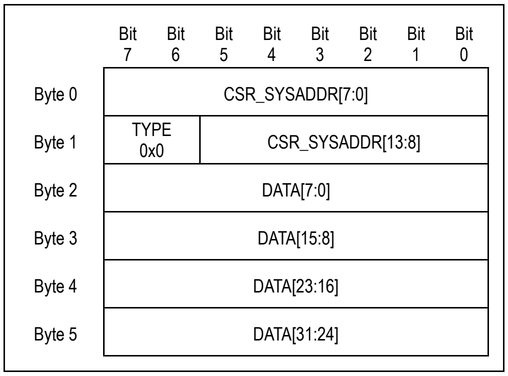
Serial EEPROM	Size
24C32	4 KB
24C64	8 KB
24C128	16 KB
24C256	32 KB
24C512	64 KB

Table 6.2 PES32T8 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 PES32T8. A double word initialization sequence occupies six byes in the serial EEPROM and is used to initialize a single double word quantity in the PES32T8.

A single double word initialization sequence consists of three fields and its format is shown in Figure 6.2. 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.



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	CSR_SYSADDR[7:0]							
Byte 1	TYPE 0x0		CSR_SYSADDR[13:8]					
Byte 2	DATA[7:0]							
Byte 3	DATA[15:8]							
Byte 4	DATA[23:16]							
Byte 5	DATA[31:24]							

Figure 6.2 Single Double Word Initialization Sequence Format

The second type of configuration block is the sequential double word initialization sequence. It is similar to a single double word initialization sequence except that it contains a double word count that allows multiple sequential double words to be initialized in one configuration block.

A sequential double word initialization sequence consists of four required fields and one to 65535 double word initialization data fields. The format of a sequential double word initialization sequence is shown in Figure 6.3. 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.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	CSR_SYSADDR[7:0]							
Byte 1	TYPE 0x1		CSR_SYSADDR[13:8]					
Byte 2	NUMDW[7:0]							
Byte 3	NUMDW[15:8]							
Byte 4	DATA0[7:0]							
Byte 5	DATA0[15:8]							
Byte 6	DATA0[23:16]							
Byte 7	DATA0[31:24]							
⋮	⋮							
Byte 4n+4	DATAn[7:0]							
Byte 4n+ 5	DATAn[15:8]							
Byte 4n+6	DATAn[23:16]							
Byte 4n+7	DATAn[31:24]							

Figure 6.3 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, Configuration Registers), 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 6.4. 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.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	CHECKSUM[7:0]							
Byte 1	TYPE 0x3		Reserved (must be zero)					

Figure 6.4 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[1]. The 1's complement of this sum is placed in the checksum field.

The checksum is verified in the following manner:

[1] This includes the byte containing the TYPE field.

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[1]. 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 Table 6.3.

Error	Action Taken
Configuration Done Sequence checksum mismatch with that computed by the PES32T8	Set RSTHALT bit in SWCTL register ICSERR bit is set in the SMBUSSTS register Abort initialization, set DONE bit in the SMBUSSTS register
Invalid configuration block type (only invalid type is 0x2)	Set RSTHALT bit in SWCTL register ICSERR bit is set in the SMBUSSTS register Abort initialization, set DONE bit in the SMBUSSTS register
An unexpected NACK is observed during a master SMBus transaction	Set RSTHALT bit in SWCTL register NAERR bit is set in the SMBUSSTS register Abort initialization, set DONE bit in the SMBUSSTS register
Master SMBus interface loses 16 consecutive arbitration attempts	Set RSTHALT bit in SWCTL register LAERR bit is set in the SMBUSSTS register Abort initialization, set DONE bit in the SMBUSSTS register
A misplaced START or STOP condition is detected by the master SMBus interface	Set RSTHALT bit in SWCTL register OTHERERR bit is set in the SMBUSSTS register Abort initialization, set DONE bit in the SMBUSSTS register

Table 6.3 Serial EEPROM Initialization Errors

Programming the Serial EEPROM

The serial EEPROM may be programmed prior to board assembly or in-system via the slave SMBus interface or a PCIe root. Programming the serial EEPROM via the slave SMBus is described in section Serial EEPROM Read or Write Operation on page 6-15. 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, 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), 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.

[1] This includes the checksum byte as well as the byte that contains the type and reserved field.

Initiating a serial EEPROM read or write operation when the BUSY bit is set produces undefined results. SMBus errors may occur when accessing the serial EEPROM. If an error occurs, 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 PES32T8 utilizes external SMBus/I2C-bus I/O expanders connected to the master SMBus interface for hot-plug and port status signals. The PES32T8 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. The PES32T8 supports up to five external I/O expanders. Table 6.4 summarizes the allocation of functions to I/O expanders. I/O expanders zero through three are used to provide hot-plug I/O signals while I/O expander four is used to provide link status and activity LED control.

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 2 hot-plug
	Upper	Port 4 hot-plug
1	Lower	Port 3 hot-plug
	Upper	Port 5 hot-plug
2	Lower	Port 6 hot-plug
	Upper	Power good inputs
3	Lower	Port 7 hot-plug
	Upper	Port 1 hot-plug
4	Lower	Link status
	Upper	Link activity

Table 6.4 I/O Expander Function Allocation

During the PES32T8 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[0:4]ADDR) field. The IOE[0:3]ADDR fields are contained in the I/O Expander Address 0 (IOEXPADDR0) register while the IOE[4]ADDR field is contained in the SMBus I/O Expander Address 1 (IOEXPADDR1) 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 a IOExADDR field is modified, SMBus write transactions are issued to the corresponding I/O expander by the PES32T8 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 are set to their negated value (e.g., the power indicator is turned off, the link is down, there is no activity, etc.).

The following I/O expander configuration sequence is issued by the PES32T8 to I/O expanders zero, one and three (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 PES32T8 to I/O expander two (i.e., the one that contains hot-plug signals and power good inputs).

- 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 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 all inputs 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 PES32T8 to I/O expander four.

- 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 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 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).

While the I/O expander is enabled, the PES32T8 maintains the I/O bus expander signals and the PES32T8 internal view of the hot-plug signals in a consistent state. This means that whenever that I/O bus expander state and the PES32T8 internal view of the signal state differs, an SMBus transaction is initiated by the PES32T8 to resolve the state conflict.

Notes

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 PES32T8 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 PES32T8 will initiate an SMBus transaction to modify the state of the I/O expander hot-plug outputs.

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 PES32T8 I/O expander interrupt input. Since the PES32T8 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 PES32T8 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 PES32T8 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 PES32T8 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 PES32T8 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 PES32T8 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 reflects the current state, as viewed by the PES32T8, 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 PES32T8 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 IOEXPTINF register allows an I/O expander test mode to be entered. When this bit is set, the PES32T8 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 PES32T8 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. 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 Tables 6.5 through 6.9.

Notes

System Design Recommendations

1. 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 PES32T8 will initialize the I/O expanders when normal device operation begins following the completion of the fundamental reset sequence.
2. 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.
3. 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.
4. 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.
5. I/O expander outputs are not modified when the device transitions from normal operation to a fundamental reset. In systems where I/O expander output values must be reset during a fundamental reset, a PCA9539 I/O expander should be used.

I/O Expander 0

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	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 6.5 I/O Expander 0 Signals

[1] 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)[1]	I	P3APN	Port 3 attention push button input
1 (I/O-0.1)	I	P3PDN	Port 3 presence detect input
2 (I/O-0.2)	I	P3PFN	Port 3 power fault input
3 (I/O-0.3)	I	P3MRLN	Port 3 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P3AIN	Port 3 attention indicator output
5 (I/O-0.5)	O	P3PIN	Port 3 power indicator output
6 (I/O-0.6)	O	P3PEP	Port 3 power enable output
7 (I/O-0.7)	O	P3ILOCKP	Port 3 electromechanical interlock
8 (I/O-1.0)	I	P5APN	Port 5 attention push button input
9 (I/O-1.1)	I	P5PDN	Port 5 presence detect input
10 (I/O-1.2)	I	P5PFN	Port 5 power fault input
11 (I/O-1.3)	I	P5MRLN	Port 5 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P5AIN	Port 5 attention indicator output
13 (I/O-1.5)	O	P5PIN	Port 5 power indicator output
14 (I/O-1.6)	O	P5PEP	Port 5 power enable output
15 (I/O-1.7)	O	P5ILOCKP	Port 5 electromechanical interlock

Table 6.6 I/O Expander 1 Signals

[1] 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)[1]	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

Table 6.7 I/O Expander 2 Signals (Part 1 of 2)

SMBus I/O Expander Bit	Type	Signal	Description
8 (I/O-1.0)	I	-	Unused
9 (I/O-1.1)	I	P1PWRGDN	Port 1 power good input
10 (I/O-1.2)	I	P2PWRGDN	Port 2 power good input
11 (I/O-1.3)	I	P3PWRGDN	Port 3 power good input
12 (I/O-1.4)	I	P4PWRGDN	Port 4 power good input
13 (I/O-1.5)	I	P5PWRGDN	Port 5 power good input
14 (I/O-1.6)	I	P6PWRGDN	Port 6 power good input
15 (I/O-1.7)	I	P7PWRGDN	Port 7 power good input

Table 6.7 I/O Expander 2 Signals (Part 2 of 2)

[1] I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

I/O Expander 3

SMBus I/O Expander Bit	Type	Signal	Description
0 (I/O-0.0)[1]	I	P7APN	Port 7 attention push button input
1 (I/O-0.1)	I	P7PDN	Port 7 presence detect input
2 (I/O-0.2)	I	P7PFN	Port 7 power fault input
3 (I/O-0.3)	I	P7MRLN	Port 7 manually-operated retention latch (MRL) input
4 (I/O-0.4)	O	P7AIN	Port 7 attention indicator output
5 (I/O-0.5)	O	P7PIN	Port 7 power indicator output
6 (I/O-0.6)	O	P7PEP	Port 7 power enable output
7 (I/O-0.7)	O	P7ILOCKP	Port 7 electromechanical interlock
8 (I/O-1.0)	I	P1APN	Port 1 attention push button input
9 (I/O-1.1)	I	P1PDN	Port 1 presence detect input
10 (I/O-1.2)	I	P1PFN	Port 1 power fault input
11 (I/O-1.3)	I	P1MRLN	Port 1 manually-operated retention latch (MRL) input
12 (I/O-1.4)	O	P1AIN	Port 1 attention indicator output
13 (I/O-1.5)	O	P1PIN	Port 1 power indicator output
14 (I/O-1.6)	O	P1PEP	Port 1 power enable output
15 (I/O-1.7)	O	P1ILOCKP	Port 1 electromechanical interlock

Table 6.8 I/O Expander 3 Signals

[1] 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)[1]	O	P0LINKUP	Port 0 link up status output
1 (I/O-0.1)	O	P1LINKUP	Port 1 link up status output
2 (I/O-0.2)	O	P2LINKUP	Port 2 link up status output
3 (I/O-0.3)	O	P3LINKUP	Port 3 link up status output
4 (I/O-0.4)	O	P4LINKUP	Port 4 link up status output
5 (I/O-0.5)	O	P5LINKUP	Port 5 link up status output
6 (I/O-0.6)	O	P6LINKUP	Port 6 link up status output
7 (I/O-0.7)	O	P7LINKUP	Port 7 link up status output
8 (I/O-1.0)	O	P0ACTIVE	Port 0 activity output
9 (I/O-1.1)	O	P1ACTIVE	Port 1 activity output
10 (I/O-1.2)	O	P2ACTIVE	Port 2 activity output
11 (I/O-1.3)	O	P3ACTIVE	Port 3 activity output
12 (I/O-1.4)	O	P4ACTIVE	Port 4 activity output
13 (I/O-1.5)	O	P5ACTIVE	Port 5 activity output
14 (I/O-1.6)	O	P6ACTIVE	Port 6 activity output
15 (I/O-1.7)	O	P7ACTIVE	Port 7 activity output

Table 6.9 I/O Expander 4 Signals

[1] I/O-x.y corresponds to the notation used for PCA9555 port x I/O pin y.

Slave SMBus Interface

The slave SMBus interface provides the PES32T8 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 the Fundamental Reset section on page 2-5). 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 6.10.

Address Bit	Address Bit Value
1	SSMBADDR[1]
2	SSMBADDR[2]
3	SSMBADDR[3]
4	0

Table 6.10 Slave SMBus Address When a Static Address is Selected

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

Table 6.10 Slave SMBus Address When a Static Address is Selected

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 6.5 and described in Table 6.11.

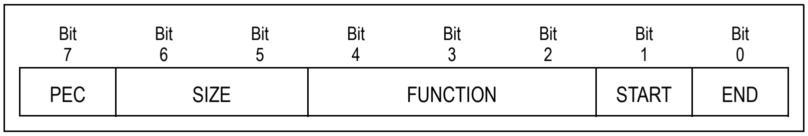


Figure 6.5 Slave SMBus Command Code Format

Bit	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 6.11 Slave SMBus Command Code Fields (Part 1 of 2)

Notes

Bit	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 6.11 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, each will be described 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, the command is NACKed. This indicates to the master that the transaction should be retried.

CSR Register Read or Write Operation

Table 6.12 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

Byte Position	Field Name	Description
0	CCODE	**Command Code.** Slave Command Code field described in Table 6.11.
1	BYTCNT	**Byte Count.** The byte count field is only transmitted for block type SMBus transactions. SMBus word and byte accesses do not contain this field. The byte count field indicates the number of bytes following the byte count field when performing a write or setting up for a read. The byte count field is also used when returning data to indicate the number of following bytes (including status). *Note that the byte count field does not include the PEC byte if PEC is enabled.*
2	CMD	**Command.** This field encodes fields related to the CSR register read or write operation.
3	ADDRL	**Address Low.** Lower 8-bits of the doubleword CSR system address of register to access.
4	ADDRU	**Address Upper.** 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	**Data Lower.** Bits [7:0] of data doubleword.

Table 6.12 CSR Register Read or Write Operation Byte Sequence (Part 1 of 2)

Byte Position	Field Name	Description
6	DATALM	**Data Lower Middle.** Bits [15:8] of data doubleword.
7	DATAUM	**Data Upper Middle.** Bits [23:16] of data doubleword.
8	DATAUU	**Data Upper.** Bits [31:24] of data doubleword.

Table 6.12 CSR Register Read or Write Operation Byte Sequence (Part 2 of 2)

The format of the CMD field is shown in Figure 6.6 and described in Table 6.13.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WERR	RERR	0	OP	BEUU	BEUM	BELM	BELL

Figure 6.6 CSR Register Read or Write CMD Field Format

Bit Field	Name	Type	Description
0	BELL	Read/Write	**Byte Enable Lower**. When set, the byte enable for bits [7:0] of the data word is enabled.
1	BELM	Read/Write	**Byte Enable Lower Middle**. When set, the byte enable for bits [15:8] of the data word is enabled.
2	BEUM	Read/Write	**Byte Enable Upper Middle**. When set, the byte enable for bits [23:16] of the data word is enabled.
3	BEUU	Read/Write	Byte Enable Upper. When set, the byte enable for bits [31:24] of the data word is enabled.
4	OP	Read/Write	CSR Operation. This field encodes the CSR operation to be performed. 0 - CSR write 1 - CSR read
5	0	0	Reserved. Must be zero
6	RERR	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 6.13 CSR Register Read or Write CMD Field Description

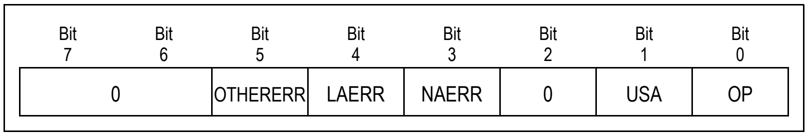
Serial EEPROM Read or Write Operation

Table 6.14 indicates the sequence of data as it is presented on the slave SMBus following the byte address of the Slave SMBus interface.

Byte Position	Field Name	Description
0	CCODE	**Command Code.** Slave Command Code field described in Table 6.11.
1	BYTCNT	**Byte Count.** The byte count field is only transmitted for block type SMBus transactions. SMBus word and byte accesses to not contain this field. The byte count field indicates the number of bytes following the byte count field when performing a write or setting up for a read. The byte count field is also used when returning data to indicate the number of following bytes (including status).
2	CMD	**Command.** This field contains information related to the serial EEPROM transaction
3	EEADDR	**Serial EEPROM Address.** This field specifies the address of the Serial EEPROM on the Master SMBus when the USA bit is set in the CMD field. Bit zero must be zero and thus the 7-bit address must be left justified.
4	ADDRL	**Address Low.** Lower 8-bits of the Serial EEPROM byte to access.
5	ADDRU	**Address Upper.** Upper 8-bits of the Serial EEPROM byte to access.
6	DATA	**Data.** Serial EEPROM value read or to be written.

Table 6.14 Serial EEPROM Read or Write Operation Byte Sequence

The format of the CMD field is shown in Figure 6.7 and described in Table 6.15.



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0		OTHERERR	LAERR	NAERR	0	USA	OP

Figure 6.7 Serial EEPROM Read or Write CMD Field Format

Bit Field	Name	Type[1]	Description
0	OP	RW	**Serial EEPROM Operation.** This field encodes the serial EEPROM operation to be performed. 0 - Serial EEPROM write 1 - Serial EEPROM read
1	USA	RW	**Use Specified Address**. When this bit is set the serial EEPROM SMBus address specified in the EEADDR 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 6.15 Serial EEPROM Read or Write CMD Field Description (Part 1 of 2)

Bit Field	Name	Type[1]	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 6.15 Serial EEPROM Read or Write CMD Field Description (Part 2 of 2)

[1] See Table 2 in the About This Manual chapter for a definition of these abbreviations.

Sample Slave SMBus Operation

Figures 6.8 through 6.13 illustrate sample Slave SMBus operations. Shaded items are driven by the PES32T8's slave SMBus interface and non-shaded items are driven by an SMBus host.

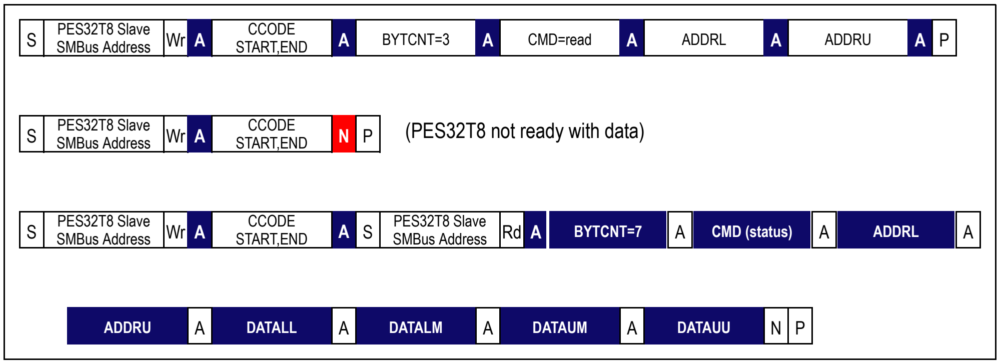


Figure 6.8 CSR Register Read Using SMBus Block Write/Read Transactions with PEC Disabled

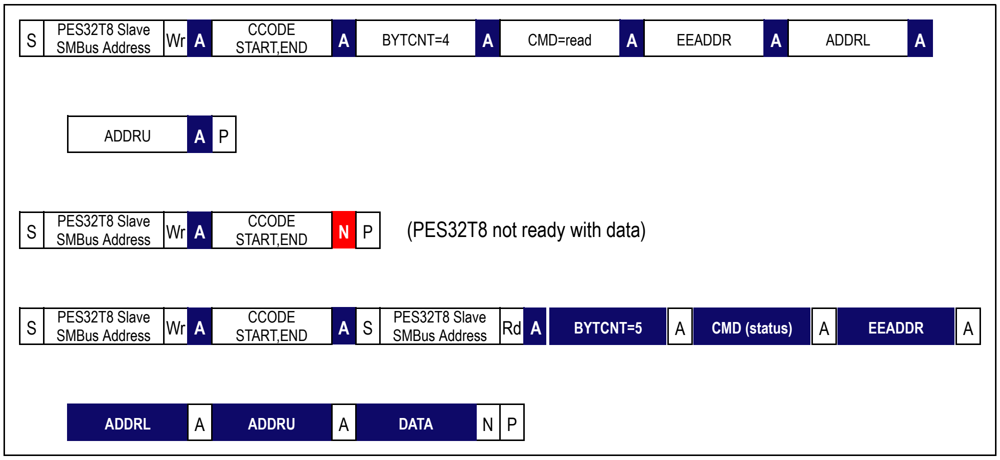


Figure 6.9 Serial EEPROM Read Using SMBus Block Write/Read Transactions with PEC Disabled

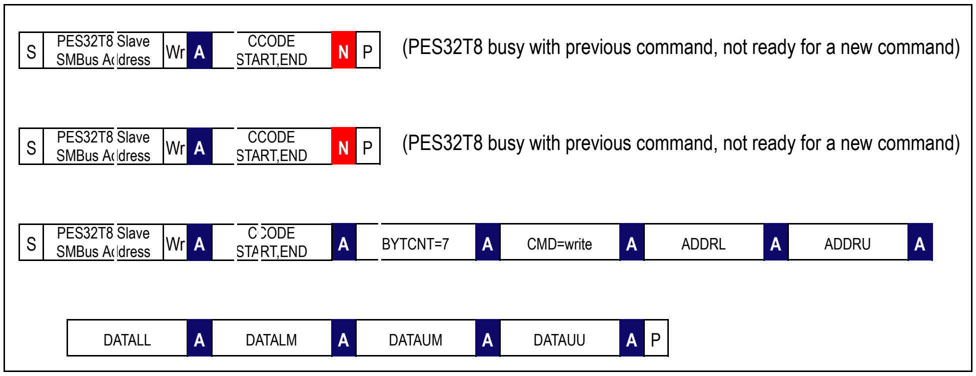

S
PES32T8 Slave SMBus Address
Wr
A
CCODE START,END
N
P
(PES32T8 busy with previous command, not ready for a new command)
S
PES32T8 Slave SMBus Address
Wr
A
CCODE START,END
N
P
(PES32T8 busy with previous command, not ready for a new command)
S
PES32T8 Slave SMBus Address
Wr
A
CCODE START,END
A
BYTCNT=7
A
CMD=write
A
ADDRL
A
ADDRU
A
DATALL
A
DATALM
A
DATAUM
A
DATAUU
A
P

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

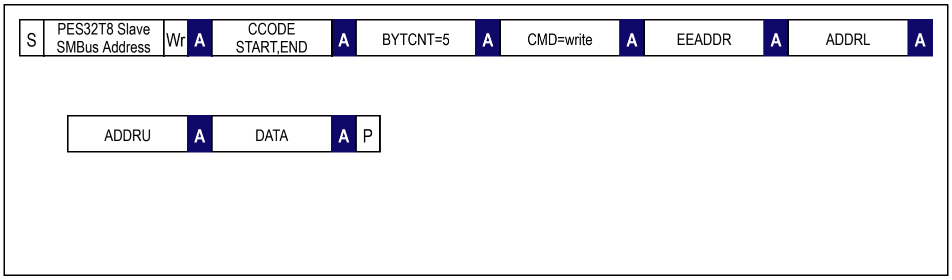


Figure 6.11 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Disabled

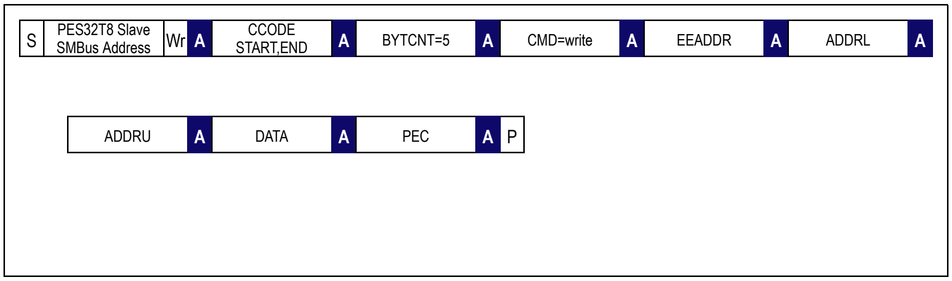


Figure 6.12 Serial EEPROM Write Using SMBus Block Write Transactions with PEC Enabled

Notes

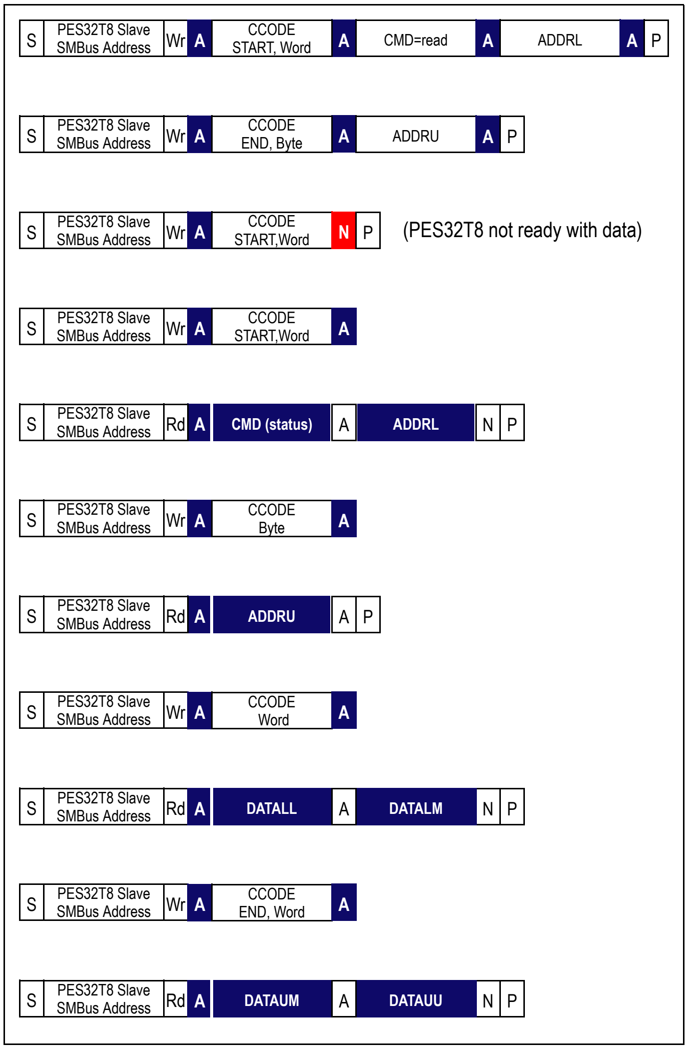

S | PES32T8 Slave SMBus Address | Wr | A | CCODE START, Word | A | CMD=read | A | ADDRL | A | P
S | PES32T8 Slave SMBus Address | Wr | A | CCODE END, Byte | A | ADDRU | A | P
S | PES32T8 Slave SMBus Address | Wr | A | CCODE START,Word | N | P
(PES32T8 not ready with data)
S | PES32T8 Slave SMBus Address | Wr | A | CCODE START,Word | A
S | PES32T8 Slave SMBus Address | Rd | A | CMD (status) | A | ADDRL | N | P
S | PES32T8 Slave SMBus Address | Wr | A | CCODE Byte | A
S | PES32T8 Slave SMBus Address | Rd | A | ADDRU | A | P
S | PES32T8 Slave SMBus Address | Wr | A | CCODE Word | A
S | PES32T8 Slave SMBus Address | Rd | A | DATALL | A | DATALM | N | P
S | PES32T8 Slave SMBus Address | Wr | A | CCODE END, Word | A
S | PES32T8 Slave SMBus Address | Rd | A | DATAUM | A | DATAUU | N | P

Figure 6.13 CSR Register Read Using SMBus Read and Write Transactions with PEC Disabled

Notes

Power Management

Notes

Introduction

Located in configuration space of each PCI-PCI bridge in the PES32T8 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 $D3_{hot}$ state allows the link associated with the bridge to enter the L1 state.

- The link associated with a port in the $D3_{hot}$ state will attempt to transition into L1 linkstate irrespective of the link or power management state of any other switch port.

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 PES32T8 supports the following device power management states: D0 Uninitialized, D0 Active, $D3_{hot}$, and $D3_{cold}$. A power management state transition diagram for the states supported by the PES32T8 is provided in Figure 7.1 and described in Table 7.1.

Transitioning a port's power management state from $D3_{hot}$ to D0 Uninitialized does not result in any logic being reset or re-initialization of register values. Thus, the default value of the No Soft Reset (NOSOFTRST) bit in the PCI Power Management Control and Status (PMCSR) register corresponds to the functional context being maintained in the $D3_{hot}$ state.

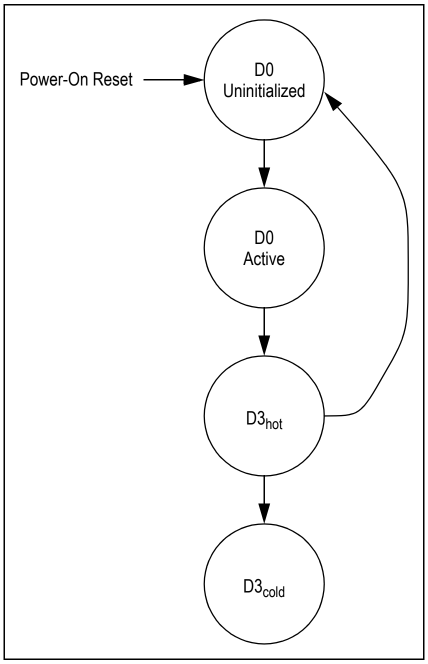


Figure 7.1 PES32T8 Power Management State Transition Diagram

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	$D3_{hot}$	The Power Management State (PMSTATE) field in the PCI Power Management Control and Status (PMCSR) register is written with the value that corresponds to the $D3_{hot}$ state.
$D3_{hot}$	D0 Uninitialized	The Power Management State (PMSTATE) field in the PCI Power Management Control and Status (PMCSR) register is written with the value that corresponds to D0 state.
$D3_{hot}$	$D3_{cold}$	Power is removed from the device.

Table 7.1 PES32T8 Power Management State Transition Diagram

The PES32T8 PCI-to-PCI bridges (i.e., ports) have the following behavior when in the $D3_{hot}$ 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 $D3_{hot}$ (e.g, generation of an ERR_NONFTAL 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 PES32T8 does not support generation of PME messages from the $D3_{cold}$ 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 $D3_{hot}$ state. This includes both the case when the downstream port is in the $D3_{hot}$ state or the entire switch is in the $D3_{hot}$ 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

The PME_Turn_Off / PME_TO_Ack protocol may be initiated by the root when the switch is in any power management state. When the PES32T8 receives a PME_Turn_Off message, it broadcasts the PME_Turn_Off message on all active downstream ports. The PES32T8 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 active downstream ports. This process is called PME_TO_Ack aggregation.

The aggregation of PME_TO_Ack messages on downstream ports is abandoned by the PES32T8 when it receives a TLP on its upstream port after it has received a PME_Turn_Off message on that port, but before it has responded with a PME_TO_Ack message. Once a PME_TO_Ack message has been scheduled for transmission on the upstream port, the PME_TO_Ack aggregation process has completed, and all received TLPs at that point may be discarded.

If the TLP that causes PME_TO_Ack aggregation to be abandoned targets the PES32T8, then the PES32T8 responds to the TLP normally. If the TLP that causes aggregation to be abandoned targets a downstream port and the port is in L0, then the TLP is transmitted on the downstream port. If the downstream port is not in L0 (i.e., it is in L2/L3 Ready), the switch transitions the link to Detect and then to L0. Once the link reaches L0, the TLP is transmitted on the downstream port.

When PME_TO_Ack aggregation is abandoned, the PES32T8 makes no attempt to abandon the PME_Turn_Off and PME_TO_Ack protocol on downstream ports. Devices downstream of the PES32T8 are allowed to respond with a PME_TO_Ack and to transition to L2/L3 Ready. When a TLP is received that targets the downstream port, the switch transitions the link to Detect and then to L0. Once the link reaches L0, the TLP is transmitted on the downstream port.

In order to avoid a 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 $D3_{cold}$ state, the PES32T8 is transitioned to the $D0_{uninitialized}$ state, it will resume generating PM_PME messages.

Power Budgeting Capability

The PES32T8 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 one of the other enhanced capabilities should be initialized to point to the power budgeting capability. The Next Pointer (NXTPTR) of the power budgeting capability should be adjusted if necessary.

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 9-49 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 Switch Control (SWCTL) 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 PES32T8 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 PES32T8 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 PES32T8 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 8.3, it will not be discussed further.

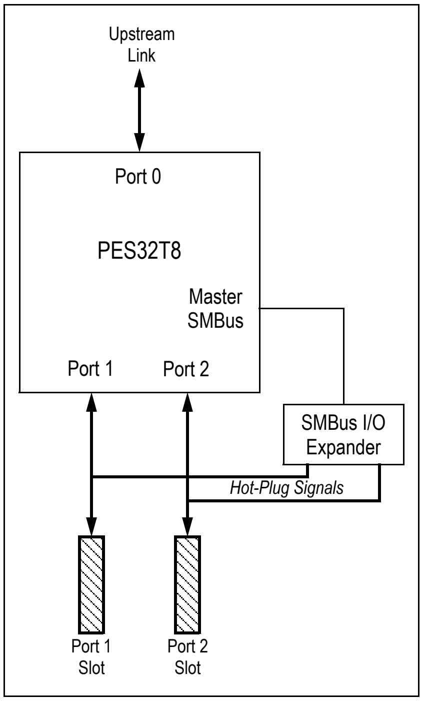


Figure 8.1 Hot-Plug on Switch Downstream Slots Application

Notes

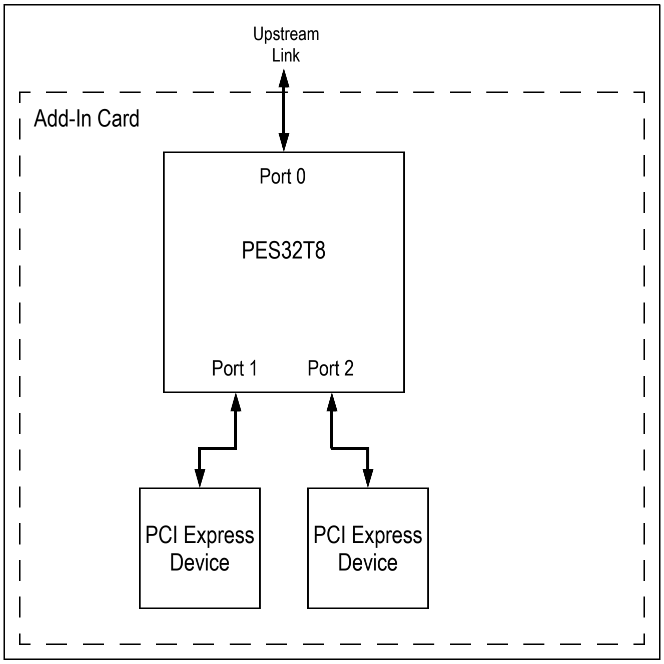


Figure 8.2 Hot-Plug with Switch on Add-In Card Application

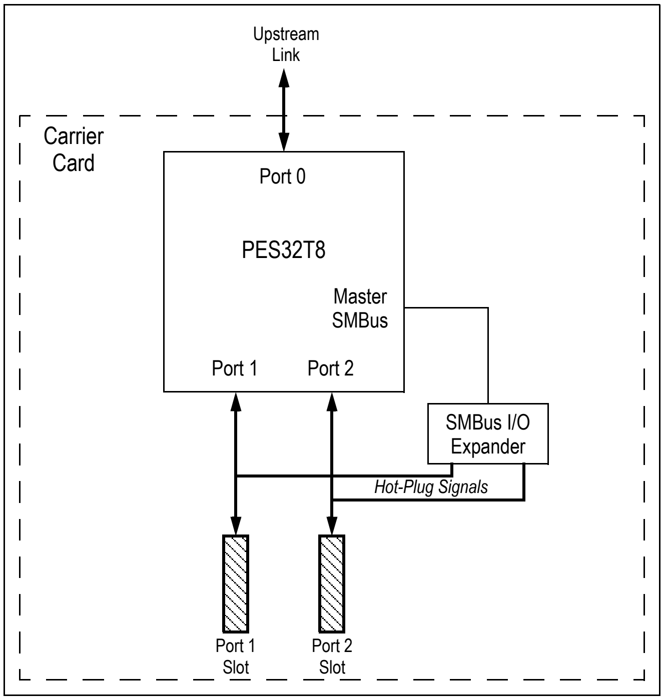


Figure 8.3 Hot-Plug with Carrier Card Application

The PCI Express Base Specification revision 1.1 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 PES32T8.

The remainder of this section discusses the use of the PES32T8 in an application involving an add-in card hot-plugged into a downstream slot. Associated with each downstream port in the PES32T8 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 PES32T8 allows sensor inputs and indicator outputs to be located next to the slot or on the plug in the module. Regardless of the physical location, the indicators are controlled by the PES32T8's downstream port. Table 12.1 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 PES32T8 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[1] 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[2]	O	Port x Reset Output.

Table 8.1 Downstream Port Hot-Plug Signals

[1] x corresponds to downstream port number (i.e., 1 through 7).

[2] 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 with a pulse length greater than 100 ms and less than 150 ms (i.e., it transactions from negated to asserted, maintains an asserted state for 100 to 150 ms, and then transitions back to negated). When the Toggle Electromechanical Interlock Control. (TEMICTL) bit in the HPCFGCTL register is set, writing a one to the EIC bit inverts the state of the PxILOCKP signal.

When the 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 RMRLWEMIL bit is cleared, the EIS bit state in the PCIESSTS register always returns the value of the corresponding PxILOCKP I/O expander signal output.

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.

Downstream port reset outputs are described in section Downstream Port Reset Outputs on page 2-8.

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 PES32T8 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:

- Attention Button Pressed (ABP)
- Power Fault Detected (PFD)
- MRL Sensor Changed (MRLSC)
- Presence Detected Changed (PDC)
- 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 $D3_{Hot}$ state, 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, both a PM_PME and an interrupt are generated. If the event interrupt is masked, neither a PM_PME nor interrupt are generated.

Note: 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 PES32T8 supports these systems by providing a General Purpose Event (GPEN) output as an alternate function of GPIO[7] 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[7], and GPIO[7] 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 PES32T8 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

Note: Logic gates in this diagram are intended to convey general concepts, not a direct implementation scheme.

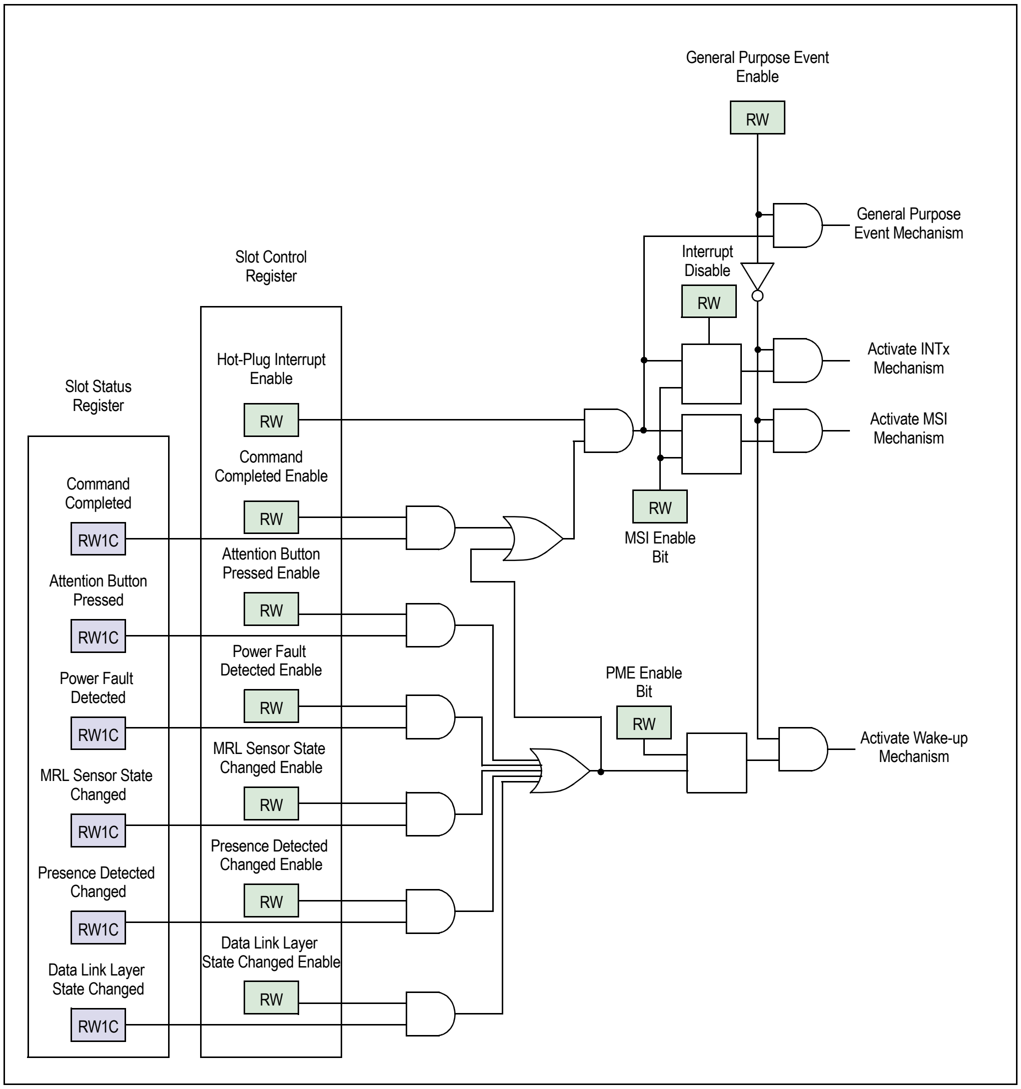

General Purpose Event Enable
RW
General Purpose Event Mechanism
Interrupt Disable
RW
Slot Control Register
Activate INTx Mechanism
Activate MSI Mechanism
Slot Status Register
Hot-Plug Interrupt Enable
RW
Command Completed Enable
RW
Command Completed
RW1C
RW
MSI Enable Bit
Attention Button Pressed Enable
RW
Attention Button Pressed
RW1C
Power Fault Detected Enable
RW
Power Fault Detected
RW1C
PME Enable Bit
RW
Activate Wake-up Mechanism
MRL Sensor State Changed Enable
RW
MRL Sensor State Changed
RW1C
Presence Detected Changed Enable
RW
Presence Detected Changed
RW1C
Data Link Layer State Changed Enable
RW
Data Link Layer State Changed
RW1C

Figure 8.4 PES32T8 Hot-Plug Event Signalling

Notes

Hot-Swap

The PES32T8 is hot-swap capable and meets the following requirements:

- All of the I/Os are tri-stated on reset (i.e., SerDes, GPIO, SMBuses, etc.).
- All I/O cells function predictably from early power. This means that the device is able to tolerate a non-monotonic ramp-up as well as a rapid ramp-up of the DC power.
- All I/O cells are able to tolerate a precharge voltage.
- Since no clock is present during physical connection, the device will maintain all outputs in a high-impedance state even when no clock is present.
- The I/O cells meet VI requirements for hot-swap.
- The I/O cells respect the required leakage current limits over the entire input voltage range.

In summary, the PES32T8 meets all of the I/O requirements necessary to build a PICMG compliant hot-swap board or system. The hot-swap I/O buffers of the PES32T8 may also be used to construct proprietary hot-swap systems. See the 89HPES32T8 Data Sheet on IDT's web site (www.IDT.com) for a detailed specification of I/O buffer characteristics.

Configuration Registers

Notes

Introduction

Each software-visible register in the PES32T8 is contained in the PCI configuration space of one of the ports. Thus, there are no registers in the PES32T8 that cannot be accessed by the root. Each software-visible register in the PES32T8 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 PES32T8 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)

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

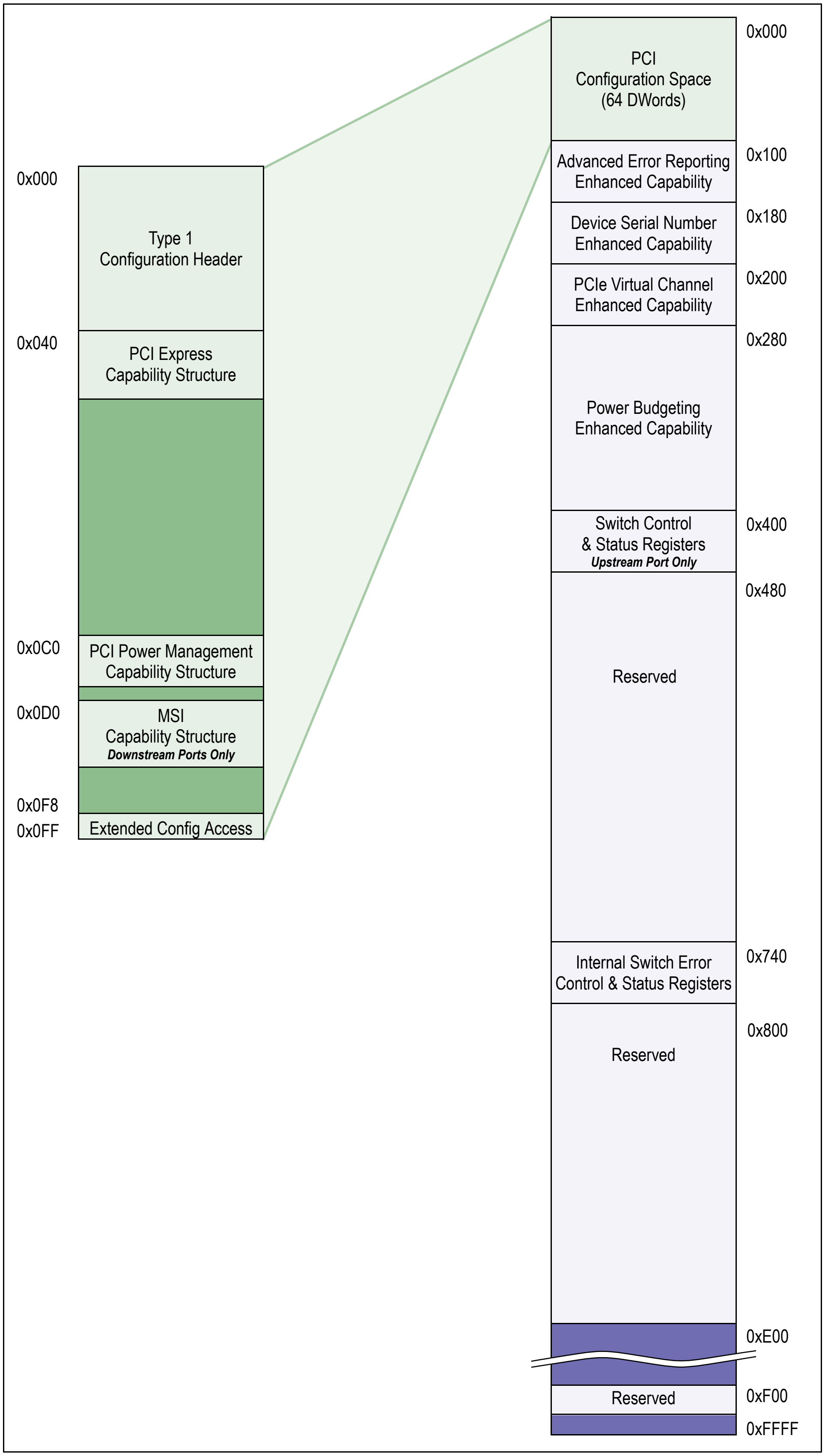


Figure 9.1 Port Configuration Space Organization

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	VID - Vendor Identification Register (0x000) on page 9-12
0x002	Word	P0_DID	DID - Device Identification Register (0x002) on page 9-12
0x004	Word	P0_PCICMD	PCICMD - PCI Command Register (0x004) on page 9-12
0x006	Word	P0_PCISTS	PCISTS - PCI Status Register (0x006) on page 9-13
0x008	Byte	P0_RID	RID - Revision Identification Register (0x008) on page 9-14
0x009	3 Bytes	P0_CCODE	CCODE - Class Code Register (0x009) on page 9-14
0x00C	Byte	P0_CLS	CLS - Cache Line Size Register (0x00C) on page 9-14
0x00D	Byte	P0_PLTIMER	PLTIMER - Primary Latency Timer (0x00D) on page 9-14
0x00E	Byte	P0_HDR	HDR - Header Type Register (0x00E) on page 9-15
0x00F	Byte	P0_BIST	BIST - Built-in Self Test Register (0x00F) on page 9-15
0x010	DWord	P0_BAR0	BAR0 - Base Address Register 0 (0x010) on page 9-15
0x014	DWord	P0_BAR1	BAR1 - Base Address Register 1 (0x014) on page 9-15
0x018	Byte	P0_PBUSN	PBUSN - Primary Bus Number Register (0x018) on page 9-15
0x019	Byte	P0_SBUSN	SBUSN - Secondary Bus Number Register (0x019) on page 9-15
0x01A	Byte	P0_SUBUSN	SUBUSN - Subordinate Bus Number Register (0x01A) on page 9-16
0x01B	Byte	P0_SLTIMER	SLTIMER - Secondary Latency Timer Register (0x01B) on page 9-16
0x01C	Byte	P0_IOBASE	IOBASE - I/O Base Register (0x01C) on page 9-16
0x01D	Byte	P0_IOLIMIT	IOLIMIT - I/O Limit Register (0x01D) on page 9-16
0x01E	Word	P0_SECSTS	SECSTS - Secondary Status Register (0x01E) on page 9-16
0x020	Word	P0_MBASE	MBASE - Memory Base Register (0x020) on page 9-17
0x022	Word	P0_MLIMIT	MLIMIT - Memory Limit Register (0x022) on page 9-17
0x024	Word	P0_PMBASE	PMBASE - Prefetchable Memory Base Register (0x024) on page 9-18
0x026	Word	P0_PMLIMIT	PMLIMIT - Prefetchable Memory Limit Register (0x026) on page 9-18
0x028	DWord	P0_PMBASEU	PMBASEU - Prefetchable Memory Base Upper Register (0x028) on page 9-18
0x02C	DWord	P0_PMLIMITU	PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C) on page 9-18
0x030	Word	P0_IOBASEU	IOBASEU - I/O Base Upper Register (0x030) on page 9-19
0x032	Word	P0_IOLIMITU	IOLIMITU - I/O Limit Upper Register (0x032) on page 9-19
0x034	Byte	P0_CAPPTR	CAPPTR - Capabilities Pointer Register (0x034) on page 9-19

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

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x038	DWord	P0_EROMBASE	EROMBASE - Expansion ROM Base Address Register (0x038) on page 9-19
0x03C	Byte	P0_INTRLINE	INTRLINE - Interrupt Line Register (0x03C) on page 9-19
0x03D	Byte	P0_INTRPIN	INTRPIN - Interrupt PIN Register (0x03D) on page 9-20
0x03E	Word	P0_BCTRL	BCTRL - Bridge Control Register (0x03E) on page 9-20
0x040	DWord	P0_PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 9-21
0x044	DWord	P0_PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 9-22
0x048	Word	P0_PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 9-23
0x04A	Word	P0_PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 9-24
0x04C	DWord	P0_PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 9-24
0x050	Word	P0_PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 9-26
0x052	Word	P0_PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 9-27
0x064	DWord	P0_PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 9-32
0x068	Word	P0_PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068 on page 9-32
0x06A	Word	P0_PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 9-32
0x06C	DWord	P0_PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 9-32
0x070	Word	P0_PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 9-32
0x072	Word	P0_PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 9-33
0x0C0	DWord	P0_PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 9-33
0x0C4	DWord	P0_PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 9-34
0x0F8	Word	P0_ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 9-37
0x0FC	Word	P0_ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 9-37
0x100	Dword	P0_AERCAP	AERCAP - AER Capabilities (0x100) on page 9-37
0x104	Dword	P0_AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 9-38
0x108	Dword	P0_AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 9-38
0x10C	Dword	P0_AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 9-40
0x110	Dword	P0_AERCES	AERCES - AER Correctable Error Status (0x110) on page 9-41
0x114	Dword	P0_AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 9-41
0x118	Dword	P0_AERCTL	AERCTL - AER Control (0x118) on page 9-42
0x11C	Dword	P0_AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 9-42

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

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x120	Dword	P0_AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 9-42
0x124	Dword	P0_AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 9-43
0x128	Dword	P0_AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 9-43
0x180	Dword	P0_SNUMCAP	SNUMCAP - Serial Number Capabilities (0x180) on page 9-43
0x184	Dword	P0_SNUMLDW	SNUMLDW - Serial Number Lower Doubleword (0x184) on page 9-43
0x188	Dword	P0_SNUMUDW	SNUMUDW - Serial Number Upper Doubleword (0x188) on page 9-43
0x200	DWord	P0_PCIEVCECAP	PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200) on page 9-44
0x204	DWord	P0_PVCCAP1	PVCCAP1 - Port VC Capability 1 (0x204) on page 9-44
0x208	DWord	P0_PVCCAP2	PVCCAP2 - Port VC Capability 2 (0x208) on page 9-44
0x20C	Word	P0_PVCCTL	PVCCTL - Port VC Control (0x20C) on page 9-45
0x20E	Word	P0_PVCSTS	PVCSTS - Port VC Status (0x20E) on page 9-45
0x210	DWord	P0_VCR0CAP	VCR0CAP - VC Resource 0 Capability (0x210) on page 9-45
0x214	DWord	P0_VCR0CTL	VCR0CTL- VC Resource 0 Control (0x214) on page 9-46
0x218	DWord	P0_VCR0STS	VCR0STS - VC Resource 0 Status (0x218) on page 9-47
0x220	DWord	P0_VCR0TBL0	VCR0TBL0 - VC Resource 0 Arbitration Table Entry 0 (0x220) on page 9-47
0x224	DWord	P0_VCR0TBL1	VCR0TBL1 - VC Resource 0 Arbitration Table Entry 1 (0x224) on page 9-48
0x228	DWord	P0_VCR0TBL2	VCR0TBL2 - VC Resource 0 Arbitration Table Entry 2 (0x228) on page 9-48
0x22C	DWord	P0_VCR0TBL3	VCR0TBL3 - VC Resource 0 Arbitration Table Entry 3 (0x22C) on page 9-48
0x280	Dword	P0_PWRBCAP	PWRBCAP - Power Budgeting Capabilities (0x280) on page 9-49
0x284	Dword	P0_PWRBDSEL	PWRBDSEL - Power Budgeting Data Select (0x284) on page 9-49
0x288	Dword	P0_PWRBD	PWRBD - Power Budgeting Data (0x288) on page 9-50
0x28C	Dword	P0_PWRBPBC	PWRBPBC - Power Budgeting Power Budget Capability (0x28C) on page 9-50
0x300	Dword	P0_PWRBDV0	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x304	Dword	P0_PWRBDV1	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x308	Dword	P0_PWRBDV2	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x30C	Dword	P0_PWRBDV3	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50

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

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x310	Dword	P0_PWRBDV4	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x314	Dword	P0_PWRBDV5	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x318	Dword	P0_PWRBDV6	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x31C	Dword	P0_PWRBDV7	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x400	DWord	SWSTS	SWSTS - Switch Status (0x400) on page 9-50
0x404	DWord	SWCTL	SWCTL - Switch Control (0x404) on page 9-51
0x408	DWord	HPCFGCTL	HPCFGCTL - Hot-Plug Configuration Control (0x408) on page 9-53
0x40C	DWord	GPR	GPR - General Purpose Register (0x40C) on page 9-54
0x418	DWord	GPIOFUNC	GPIOFUNC - General Purpose I/O Control Function (0x418) on page 9-54
0x41C	DWord	GPIOCFG	GPIOCFG - General Purpose I/O Configuration (0x41C) on page 9-54
0x420	DWord	GPIOD	GPIOD - General Purpose I/O Data (0x420) on page 9-55
0x424	DWord	SMBUSSTS	SMBUSSTS - SMBus Status (0x424) on page 9-55
0x428	DWord	SMBUSCTL	SMBUSCTL - SMBus Control (0x428) on page 9-56
0x42C	DWord	EEPROMINTF	EEPROMINTF - Serial EEPROM Interface (0x42C) on page 9-57
0x430	DWord	IOEXPINTF	IOEXPINTF - I/O Expander Interface (0x430) on page 9-57
0x434	DWord	IOEXPADDR0	IOEXPADDR0 - SMBus I/O Expander Address 0 (0x434) on page 9-58
0x438	DWord	IOEXPADDR1	IOEXPADDR1 - SMBus I/O Expander Address 1 (0x438) on page 9-59
0x450	DWord	GPECTL	GPECTL - General Purpose Event Control (0x450) on page 9-59
0x454	DWord	GPESTS	GPESTS - General Purpose Event Status (0x454) on page 9-60
0x458	DWord	UARBTC	UARBTC - U-Bus Arbiter Transfer Count (0x458) on page 9-61
0x45C	DWord	UARBCTC	UARBCTC - U-Bus Arbiter Current Transfer Count (0x45C) on page 9-61
0x460	DWord	DARBTC	DARBTC - D-Bus Arbiter Transfer Count (0x460) on page 9-62
0x464	DWord	DARBCTC	DARBCTC - D-Bus Arbiter Current Transfer Count (0x464) on page 9-62
0x46C	DWord	SWCTCTL	SWCTCTL - Switch Cut Through Control (0x46C) on page 9-62
0x4A8	DWord	SWTSCNTCTL	SWTSCNTCTL - Switch Time-Stamp Counter Control (0x4A8) on page 9-62
0x740	Dword	P0_SWPECTL	SWPECTL - Switch Parity Error Control (0x740) on page 9-63
0x744	Dword	P0_SWPESTS	SWPESTS - Switch Parity Error Status (0x744) on page 9-63
0x748	Dword	P0_SWPERCTL	SWPERCTL - Switch Parity Error Reporting Control (0x748) on page 9-63

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

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x74C	Dword	P0_SWPECNT	SWPECNT - Switch Parity Error Count (0x74C) on page 9-64
0x750	Dword	P0_SWTOCTL	SWTOCTL - Switch Time-Out Control (0x750) on page 9-64
0x754	Dword	P0_SWTOSTS	SWTOSTS - Switch Time-Out Status (0x754) on page 9-64
0x758	Dword	P0_SWTORCTL	SWTORCTL - Switch Time-Out Reporting Control (0x758) on page 9-65
0x75C	Dword	P0_SWTOCNT	SWTOCNT - Switch Time-Out Count (0x75C) on page 9-66
0x760	Dword	P0_SWTOTSCTL	SWTOTSCTL - Switch Time-Out Time-Stamp Control (0x760) on page 9-66

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

Notes

Downstream Ports (Ports 1 through 7)

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

Table 9.3 Downstream Ports 1 through 7 Configuration Space Registers (Part 1 of 4)

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x038	DWord	Px_EROMBASE	EROMBASE - Expansion ROM Base Address Register (0x038) on page 9-19
0x03C	Byte	Px_INTRLINE	INTRLINE - Interrupt Line Register (0x03C) on page 9-19
0x03D	Byte	Px_INTRPIN	INTRPIN - Interrupt PIN Register (0x03D) on page 9-20
0x03E	Word	Px_BCTRL	BCTRL - Bridge Control Register (0x03E) on page 9-20
0x040	DWord	Px_PCIECAP	PCIECAP - PCI Express Capability (0x040) on page 9-21
0x044	DWord	Px_PCIEDCAP	PCIEDCAP - PCI Express Device Capabilities (0x044) on page 9-22
0x048	Word	Px_PCIEDCTL	PCIEDCTL - PCI Express Device Control (0x048) on page 9-23
0x04A	Word	Px_PCIEDSTS	PCIEDSTS - PCI Express Device Status (0x04A) on page 9-24
0x04C	DWord	Px_PCIELCAP	PCIELCAP - PCI Express Link Capabilities (0x04C) on page 9-24
0x050	Word	Px_PCIELCTL	PCIELCTL - PCI Express Link Control (0x050) on page 9-26
0x052	Word	Px_PCIELSTS	PCIELSTS - PCI Express Link Status (0x052) on page 9-27
0x054	DWord	Px_PCIESCAP	PCIESCAP - PCI Express Slot Capabilities (0x054) on page 9-28
0x058	Word	Px_PCIESCTL	PCIESCTL - PCI Express Slot Control (0x058) on page 9-29
0x05A	Word	Px_PCIESSTS	PCIESSTS - PCI Express Slot Status (0x05A) on page 9-31
0x064	DWord	Px_PCIEDCAP2	PCIEDCAP2 - PCI Express Device Capabilities 2 (0x064) on page 9-32
0x068	Word	Px_PCIEDCTL2	PCIEDCTL2 - PCI Express Device Control 2 (0x068 on page 9-32
0x06A	Word	Px_PCIEDSTS2	PCIEDSTS2 - PCI Express Device Status 2 (0x06A) on page 9-32
0x06C	DWord	Px_PCIELCAP2	PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C) on page 9-32
0x070	Word	Px_PCIELCTL2	PCIELCTL2 - PCI Express Link Control 2 (0x070) on page 9-32
0x072	Word	Px_PCIELSTS2	PCIELSTS2 - PCI Express Link Status 2 (0x072) on page 9-33
0x074	DWord	Px_PCIESCAP2	PCIESCAP2 - PCI Express Slot Capabilities 2 (0x074) on page 9-33
0x078	Word	Px_PCIESCTL2	PCIESCTL2 - PCI Express Slot Control 2 (0x078) on page 9-33
0x07A	Word	Px_PCIESSTS2	PCIESSTS2 - PCI Express Slot Status 2 (0x07A) on page 9-33
0x0C0	DWord	Px_PMCAP	PMCAP - PCI Power Management Capabilities (0x0C0) on page 9-33
0x0C4	DWord	Px_PMCSR	PMCSR - PCI Power Management Control and Status (0x0C4) on page 9-34
0x0D0	DWord	Px_MSICAP	MSICAP - Message Signaled Interrupt Capability and Control (0x0D0) on page 9-35
0x0D4	DWord	Px_MSIADDR	MSIADDR - Message Signaled Interrupt Address (0x0D4) on page 9-35
0x0D8	DWord	Px_MSIUADDR	MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8) on page 9-36
0x0DC	DWord	Px_MSIMDATA	MSIMDATA - Message Signaled Interrupt Message Data (0x0DC) on page 9-36

Table 9.3 Downstream Ports 1 through 7 Configuration Space Registers (Part 2 of 4)

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x0F0	Dword	Px_SSIDSSVIDCAP	SSIDSSVIDCAP - Subsystem ID and Subsystem Vendor ID Capability (0x0F0) on page 9-36
0x0F4	Dword	Px_SSIDSSVID	SSIDSSVID - Subsystem ID and Subsystem Vendor ID (0x0F4) on page 9-36
0x0F8	Word	Px_ECFGADDR	ECFGADDR - Extended Configuration Space Access Address (0x0F8) on page 9-37
0x0FC	Word	Px_ECFGDATA	ECFGDATA - Extended Configuration Space Access Data (0x0FC) on page 9-37
0x100	Dword	Px_AERCAP	AERCAP - AER Capabilities (0x100) on page 9-37
0x104	Dword	Px_AERUES	AERUES - AER Uncorrectable Error Status (0x104) on page 9-38
0x108	Dword	Px_AERUEM	AERUEM - AER Uncorrectable Error Mask (0x108) on page 9-38
0x10C	Dword	Px_AERUESV	AERUESV - AER Uncorrectable Error Severity (0x10C) on page 9-40
0x110	Dword	Px_AERCES	AERCES - AER Correctable Error Status (0x110) on page 9-41
0x114	Dword	Px_AERCEM	AERCEM - AER Correctable Error Mask (0x114) on page 9-41
0x118	Dword	Px_AERCTL	AERCTL - AER Control (0x118) on page 9-42
0x11C	Dword	Px_AERHL1DW	AERHL1DW - AER Header Log 1st Doubleword (0x11C) on page 9-42
0x120	Dword	Px_AERHL2DW	AERHL2DW - AER Header Log 2nd Doubleword (0x120) on page 9-42
0x124	Dword	Px_AERHL3DW	AERHL3DW - AER Header Log 3rd Doubleword (0x124) on page 9-43
0x128	Dword	Px_AERHL4DW	AERHL4DW - AER Header Log 4th Doubleword (0x128) on page 9-43
0x180	Dword	Px_SNUMCAP	SNUMCAP - Serial Number Capabilities (0x180) on page 9-43
0x184	Dword	Px_SNUMLDW	SNUMLDW - Serial Number Lower Doubleword (0x184) on page 9-43
0x188	Dword	Px_SNUMUDW	SNUMUDW - Serial Number Upper Doubleword (0x188) on page 9-43
0x200	DWord	Px_PCIEVCECAP	PCIEVCECAP - PCI Express VC Enhanced Capability Header (0x200) on page 9-44
0x204	DWord	Px_PVCCAP1	PVCCAP1 - Port VC Capability 1 (0x204) on page 9-44
0x208	DWord	Px_PVCCAP2	PVCCAP2 - Port VC Capability 2 (0x208) on page 9-44
0x20C	Word	Px_PVCCTL	PVCCTL - Port VC Control (0x20C) on page 9-45
0x20E	Word	Px_PVCSTS	PVCSTS - Port VC Status (0x20E) on page 9-45
0x210	DWord	Px_VCR0CAP	VCR0CAP - VC Resource 0 Capability (0x210) on page 9-45
0x214	DWord	Px_VCR0CTL	VCR0CTL- VC Resource 0 Control (0x214) on page 9-46
0x218	DWord	Px_VCR0STS	VCR0STS - VC Resource 0 Status (0x218) on page 9-47
0x280	Dword	Px_PWRBCAP	PWRBCAP - Power Budgeting Capabilities (0x280) on page 9-49
0x284	Dword	Px_PWRBDSEL	PWRBDSEL - Power Budgeting Data Select (0x284) on page 9-49

Table 9.3 Downstream Ports 1 through 7 Configuration Space Registers (Part 3 of 4)

Notes

Cfg. Offset	Size	Register Mnemonic	Register Definition
0x288	Dword	Px_PWRBD	PWRBD - Power Budgeting Data (0x288) on page 9-50
0x28C	Dword	Px_PWRBPBC	PWRBPBC - Power Budgeting Power Budget Capability (0x28C) on page 9-50
0x300	Dword	Px_PWRBDV0	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x304	Dword	Px_PWRBDV1	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x308	Dword	Px_PWRBDV2	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x30C	Dword	Px_PWRBDV3	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x310	Dword	Px_PWRBDV4	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x314	Dword	Px_PWRBDV5	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x318	Dword	Px_PWRBDV6	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x31C	Dword	Px_PWRBDV7	PWRBDV[0..7] - Power Budgeting Data Value [0..7] (0x300) on page 9-50
0x740	Dword	Px_SWPECTL	SWPECTL - Switch Parity Error Control (0x740) on page 9-63
0x744	Dword	Px_SWPESTS	SWPESTS - Switch Parity Error Status (0x744) on page 9-63
0x748	Dword	Px_SWPERCTL	SWPERCTL - Switch Parity Error Reporting Control (0x748) on page 9-63
0x74C	Dword	Px_SWPECNT	SWPECNT - Switch Parity Error Count (0x74C) on page 9-64
0x750	Dword	Px_SWTOCTL	SWTOCTL - Switch Time-Out Control (0x750) on page 9-64
0x754	Dword	Px_SWTOSTS	SWTOSTS - Switch Time-Out Status (0x754) on page 9-64
0x758	Dword	Px_SWTORCTL	SWTORCTL - Switch Time-Out Reporting Control (0x758) on page 9-65
0x75C	Dword	Px_SWTOCNT	SWTOCNT - Switch Time-Out Count (0x75C) on page 9-66
0x760	Dword	Px_SWTOTSCTL	SWTOTSCTL - Switch Time-Out Time-Stamp Control (0x760) on page 9-66

Table 9.3 Downstream Ports 1 through 7 Configuration Space Registers (Part 4 of 4)

Notes

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	**Vendor Identification.** This field contains the 16-bit vendor ID value assigned to IDT. See section Vendor ID on page 1-5.

DID - Device Identification Register (0x002)

Bit Field	Field Name	Type	Default Value	Description
15:0	DID	RO	-	**Device Identification.** This field contains the 16-bit device ID assigned by IDT to this bridge. See section Device ID on page 1-5.

PCICMD - PCI Command Register (0x004)

Bit Field	Field Name	Type	Default Value	Description
0	IOAE	RW	0x0	**I/O Access Enable.** When this bit is cleared, the bridge does not respond to I/O accesses from the primary bus specified by IOBASE and IOLIMIT. 0x0 -(disable) Disable I/O space. 0x1 - (enable) Enable I/O space.
1	MAE	RW	0x0	**Memory Access Enable.** When this bit is cleared, the bridge does not respond to memory and prefetchable memory space access from the primary bus specified by MBASE, MLIMIT, PMBASE and PMLIMIT. 0x0 -(disable) Disable memory space. 0x1 - (enable) Enable memory space.
2	BME	RW	0x0	**Bus Master Enable.** When this bit is cleared, the bridge 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	**Special Cycle Enable.** Not applicable.
4	MWI	RO	0x0	**Memory Write Invalidate.** Not applicable.
5	VGAS	RO	0x0	**VGA Palette Snoop**. Not applicable.
6	PERRE	RW	0x0	**Parity Error Enable.** 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.

Notes

Bit Field	Field Name	Type	Default Value	Description
7	ADSTEP	RO	0x0	**Address Data Stepping.** Not applicable.
8	SERRE	RW	0x0	**SERR Enable.** Non-fatal and fatal errors detected by the bridge are reported to the Root Complex when this bit is set or the bits in the PCI Express Device Control register are set (see PCIEDCTL - PCI Express Device Control (0x048) on page 9-23). 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	**Fast Back-to-Back Enable.** Not applicable.
10	INTXD	RW	0x0	**INTx Disable.** 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	**INTx Status.** 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	**Capabilities List.** This bit is hardwired to one to indicate that the bridge implements an extended capability list item.
5	C66MHZ	RO	0x0	**66 MHz Capable.** Not applicable.
6	Reserved	RO	0x0	Reserved field.
7	FB2B	RO	0x0	**Fast Back-to-Back (FB2B).** Not applicable.
8	MDPED	RW1C	0x0	**Master Data Parity Error Detected.** This bit is set 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	**DEVSEL# TIming.** Not applicable.
11	STAS	RO	0x0	**Signalled Target Abort.** Not applicable since a target abort is never signalled.

Notes

Bit Field	Field Name	Type	Default Value	Description
12	RTAS	RO	0x0	**Received Target Abort.** Not applicable.
13	RMAS	RO	0x0	**Received Master Abort.** Not applicable.
14	SSE	RW1C	0x0	**Signalled System Error.** This bit is set when the bridge sends a ERR_FATAL or ERR_NONFATAL message and the SERR Enable (SERRE) bit is set in the PCICMD register. 0x0 -(noerror) no error. 0x1 - (error) This bit is set when a fatal or non-fatal error is signalled.
15	DPE	RW1C	0x0	**Detected Parity Error.** This bit is set by the bridge whenever it receives a poisoned TLP on the primary side regardless of the state of the PERRE bit in the PCI Command register.

RID - Revision Identification Register (0x008)

Bit Field	Field Name	Type	Default Value	Description
7:0	RID	RO	—	**Revision ID.** This field contains the revision identification number for the device. See section Revision ID on page 1-5.

CCODE - Class Code Register (0x009)

Bit Field	Field Name	Type	Default Value	Description
7:0	INTF	RO	0x00	**Interface.** This value indicates that the device is a PCI-PCI bridge that does not support subtractive decode.
15:8	SUB	RO	0x04	**Sub Class Code.** This value indicates that the device is a PCI-PCI bridge.
23:16	BASE	RO	0x06	**Base Class Code.** 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	**Cache Line Size.** This field has no effect on the bridge's functionality but may be read and written by software. This field is implemented for compatibility with legacy software.

PLTIMER - Primary Latency Timer (0x00D)

Bit Field	Field Name	Type	Default Value	Description
7:0	PLTIMER	RO	0x00	**Primary Latency Timer.** Not applicable.

Notes

HDR - Header Type Register (0x00E)

Bit Field	Field Name	Type	Default Value	Description
7:0	HDR	RO	0x01	**Header Type.** 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	**BIST.** 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	**Base Address Register.** Not applicable.

BAR1 - Base Address Register 1 (0x014)

Bit Field	Field Name	Type	Default Value	Description
31:0	BAR	RO	0x0	**Base Address Register.** Not applicable.

PBUSN - Primary Bus Number Register (0x018)

Bit Field	Field Name	Type	Default Value	Description
7:0	PBUSN	RW	0x0	**Primary Bus Number.** This field is used to record the bus number of the PCI bus segment to which the primary interface of the bridge is connected. This field has no functional effect within the PES32T8 but is implemented as a read/write register for software compatibility

SBUSN - Secondary Bus Number Register (0x019)

Bit Field	Field Name	Type	Default Value	Description
7:0	SBUSN	RW	0x0	**Secondary Bus Number.** This field is used to record the bus number of the PCI bus segment to which the secondary interface of the bridge is connected.

Notes

SUBUSN - Subordinate Bus Number Register (0x01A)

Bit Field	Field Name	Type	Default Value	Description
7:0	SUBUSN	RW	0x0	**Subordinate Bus Number.** The Subordinate Bus Number register is used to record the bus number of the highest numbered PCI bus segment which is behind (or subordinate to) the bridge.

SLTIMER - Secondary Latency Timer Register (0x01B)

Bit Field	Field Name	Type	Default Value	Description
7:0	SLTIMER	RO	0x0	**Secondary Latency Timer.** Not applicable.

IOBASE - I/O Base Register (0x01C)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RWL	0x1	**I/O Capability.** Indicates if the bridge supports 16-bit or 32-bit I/O addressing. 0x0 -(io16) 16-bit I/O addressing. 0x1 - (io32) 32-bit I/O addressing.
3:1	Reserved	RO	0x0	Reserved field.
7:4	IOBASE	RW	0xF	**I/O Base.** The IOBASE and IOLIMIT registers are used to control the forwarding of I/O transactions between the primary and secondary interfaces of the bridge. This field contains A[15:12] of the lowest I/O address aligned on a 4KB boundary that is below the primary interface of the bridge.

IOLIMIT - I/O Limit Register (0x01D)

Bit Field	Field Name	Type	Default Value	Description
0	IOCAP	RO	0x1	**I/O Capability.** Indicates if the bridge supports 16-bit or 32-bit I/O addressing. This bit always reflects the value of the IOCAP field in the IOBASE register.
3:1	Reserved	RO	0x0	Reserved field.
7:4	IOLIMIT	RW	0x0	**I/O Limit.** The IOBASE and IOLIMIT registers are used to control the forwarding of I/O transactions between the primary and secondary interfaces of the bridge. This field contains A[15:12] of the highest I/O address, with A[11:0] assumed to be 0xFFF, that is below the primary interface of the bridge.

SECSTS - Secondary Status Register (0x01E)

Bit Field	Field Name	Type	Default Value	Description
7:0	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
8	MDPED	RW1C	0x0	**Master Data Parity Error.** 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	**Signalled Target Abort Status.** Not applicable.
12	RTAS	RO	0x0	**Received Target Abort Status.** Not applicable.
13	RMAS	RO	0x0	**Received Master Abort Status.** Not applicable.
14	RSE	RW1C	0x0	**Received System Error.** This bit is controlled by the SERR enable bit in the Bridge Control (BCTRL) register. If the SERRE bit is cleared in BCTRL, 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	**Detected Parity Error.** This bit is set by the bridge whenever it receives a poisoned TLP on the secondary side regardless of the state of the PERRE bit in the PCI Command register

MBASE - Memory Base Register (0x020)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
15:4	MBASE	RW	0xFFF	**Memory Address Base.** The MBASE and MLIMIT registers are used to control the forwarding of non-prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the lowest address aligned on a 1MB boundary that is below the primary interface of the bridge.

MLIMIT - Memory Limit Register (0x022)

Bit Field	Field Name	Type	Default Value	Description
3:0	Reserved	RO	0x0	Reserved field.
15:4	MLIMIT	RW	0x0	**Memory Address Limit.** The MBASE and MLIMIT registers are used to control the forwarding of non-prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the highest address, with A[19:0] assumed to be 0xF_FFFF, that is below the primary interface of the bridge.

Notes

PMBASE - Prefetchable Memory Base Register (0x024)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RWL	0x1	**Prefetchable Memory Capability.** Indicates if the bridge supports 32-bit or 64-bit prefetchable memory addressing. 0x0 -(prefmem32) 32-bit prefetchable memory addressing. 0x1 - (prefmem64) 64-bit prefetchable memory addressing.
3:1	Reserved	RO	0x0	Reserved field.
15:4	PMBASE	RW	0xFFF	**Prefetchable Memory Address Base.** The PMBASE, PMBASEU, PMLIMIT and PMLIMITU registers are used to control the forwarding of prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the lowest memory address aligned on a 1MB boundary that is below the primary interface of the bridge. PMBASEU specifies the remaining bits.

PMLIMIT - Prefetchable Memory Limit Register (0x026)

Bit Field	Field Name	Type	Default Value	Description
0	PMCAP	RO	0x1	**Prefetchable Memory Capability.** Indicates if the bridge supports 32-bit or 64-bit prefetchable memory addressing. This bit always reflects the value in the PMCAP field in the PMBASE register.
3:1	Reserved	RO	0x0	Reserved field.
15:4	PMLIMIT	RW	0x0	**Prefetchable Memory Address Limit.** The PMBASE, PMBASEU, PMLIMIT and PMLIMITU registers are used to control the forwarding of prefetchable transactions between the primary and secondary interfaces of the bridge. This field contains A[31:20] of the highest memory address, with A[19:0] assumed to be 0xF_FFFF, that is below the primary interface of the bridge. PMLIMITU specifies the remaining bits

PMBASEU - Prefetchable Memory Base Upper Register (0x028)

Bit Field	Field Name	Type	Default Value	Description
31:0	PMBASEU	RW	0xFFFF_FFFF	**Prefetchable Memory Address Base Upper.** This field specifies the upper 32-bits of PMBASE when 64-bit addressing is used. When the PMCAP field in the PMBASE register is cleared, this field becomes read-only with a value of zero.

PMLIMITU - Prefetchable Memory Limit Upper Register (0x02C)

Bit Field	Field Name	Type	Default Value	Description
31:0	PMLIMITU	RW	0x0	**Prefetchable Memory Address Limit Upper.** This field specifies the upper 32-bits of PMLIMIT. When the PMCAP field in the PMBASE register is cleared, this field becomes read-only with a value of zero.

Notes

IOBASEU - I/O Base Upper Register (0x030)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOBASEU	RW	0xFFFF	**I/O Address Base Upper.** This field specifies the upper 16-bits of IOBASE. When the IOCAP field in the IOBASE register is cleared, this field becomes read-only with a value of zero.

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

Bit Field	Field Name	Type	Default Value	Description
15:0	IOLIMITU	RW	0x0	**Prefetchable IO Limit Upper.** This field specifies the upper 16-bits of IOLIMIT. When the IOCAP field in the IOBASE register is cleared, this field becomes read-only with a value of zero.

CAPPTR - Capabilities Pointer Register (0x034)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPPTR	RWL	0x40	**Capabilities Pointer.** This field specifies a pointer to the head of the capabilities structure.

EROMBASE - Expansion ROM Base Address Register (0x038)

Bit Field	Field Name	Type	Default Value	Description
31:0	EROMBASE	RO	0x0	**Expansion ROM Base Address.** 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	**Interrupt Line.** 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	**Interrupt Pin.** 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. This field should only be configured with values of 0x0 through 0x4. 0x0 -(none) Bridge does not generate any interrupts. 0x1 -(INTA) Bridge generates INTA interrupts. 0x2 -(INTB) Bridge generates INTB interrupts. 0x3 -(INTC) Bridge generates INTC interrupts. 0x4 -(INTD) Bridge generates INTD interrupts.

BCTRL - Bridge Control Register (0x03E)

Bit Field	Field Name	Type	Default Value	Description
0	PERRE	RW	0x0	**Parity Error Response Enable.** This bit controls the bridges response to poisoned TLPs on the secondary interface. 0x0 - (ignore) Ignore poisoned TLPs (i.e., parity errors) on the secondary interface. 0x1 - (report) Enable poisoned TLP (i.e., parity error) detection and reporting on the secondary interface of the bridge.
1	SERRE	RW	0x0	**System Error Enable.** This bit controls forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL from the secondary interface of the bridge to the primary interface. Note that error reporting must be enabled in the Command register or PCI Express Capability structure, Device Control register for errors to be reported on the primary interface. 0x0 - (ignore) Do not forward errors from the secondary to the primary interface. 0x1 - (report) Enable forwarding of errors from secondary to the primary interface.
2	ISAEN	RW	0x0	**ISA Enable.** This bit controls the routing of ISA I/O transactions. 0 - (disable) Forward downstream all I/O addresses in the address range defined by the I/O base and I/O limit registers. 1 - (enable) Forward upstream ISA I/O addresses in the address range defined by the I/O base and I/O limit registers that are in the first 64 KB of PCI I/O address space (top 768 bytes of each 1-KB block).
3	VGAEN	RW	0x0	**VGA Enable.** Controls the routing of processor-initiated transactions targeting VGA. 0 - (block) Do not forward VGA compatible addresses from the primary interface to the secondary interface 1 - (forward) Forward VGA compatible addresses from the primary to the secondary interface.

Notes

Bit Field	Field Name	Type	Default Value	Description
4	VGA16EN	RW	0x0	**VGA 16-bit Enable.** This bit only has an effect when the VGAEN bit is set in this register. This read/write bit enables system configuration software to select between 10-bit and 16-bit I/O space decoding for VGA transactions. 0 - (bit10) Perform 10-bit decoding. I/O space aliasing occurs in this mode. 1 -(bit16) Perform 16-bit decoding. No I/O space aliasing occurs in this mode.
5	Reserved	RO	0x0	Reserved field.
6	SRESET	RW	0x0	**Secondary Bus Reset.** Setting this bit triggers a secondary bus reset. In the upstream port, setting this bit initiates a upstream secondary bus reset. In a downstream 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	**Capability ID**. The value of 0x10 identifies this capability as a PCI Express capability structure.
15:8	NXTPTR	RWL	0xC0	**Next Pointer**. This field contains a pointer to the next capability structure.
19:16	VER	RWL	0x1	**PCI Express Capability Version.** This field indicates the PCI-SIG defined PCI Express capability structure version number. The default value of this field is 0x1 in PCIe 1.1 mode.
23:20	TYPE	RO	Upstream: 0x5 Downstream: 0x6	Port Type. This field identifies the type of switch port (upstream or downstream).
24	SLOT	RWL	0x0	**Slot Implemented.** This bit is set when the PCI Express link associated with this Port is connected to a slot. This field does not apply to an upstream port and should be set to zero.
29:25	IMN	RO	0x0	**Interrupt Message Number.** The function is allocated none (upstream ports) or only one (downstream ports) MSI. Therefore, this field is set to zero.
30	TCS	RWL	0x1	**TCS Routing Supported.** The PES32T8 supports TCS routing. The default value of this field is 0x0 in PCIe 1.1 mode.
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	**Maximum Payload Size Supported**. 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	**Phantom Functions Supported**. This field indicates the support for unclaimed function number to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers. The value is hardwired to 0x0 to indicate that no function number bits are used for phantom functions.
5	ETAG	RWL	0x1	**Extended Tag Field Support.** This field indicates the maximum supported size of the Tag field as a requester.
8:6	E0AL	RO	0x0	**Endpoint L0s Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L0s state to the L0 state. The value is hardwired to 0x0 as this field does not apply to a switch.
11:9	E1AL	RO	0x0	**Endpoint L1 Acceptable Latency.** This field indicates the acceptable total latency that an endpoint can withstand due to transition from the L1 state to the L0 state. The value is hardwired to 0x0 as this field does not apply to a switch.
12	ABP	RO	0x0	**Attention Button Present.** In PCIe base 1.1 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	**Attention Indicator Present.** In PCIe base 1.1 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	**Power Indicator Present.** In PCIe base 1.1 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	**Role Based Error Reporting.** This bit is set to indicate that the PES32T8 supports error reporting as defined in the PCIe base 1.1 and 2.0 specifications.
17:16	Reserved	RO	0x0	Reserved field.
25:18	CSPLV	RO	0x0	**Captured Slot Power Limit Value.** This field in combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. The value of this field is set by a Set_Slot_Power_Limit Message and is only applicable for the upstream port. This field is always zero in downstream ports.

Notes

Bit Field	Field Name	Type	Default Value	Description
27:26	CSPLS	RO	0x0	**Captured Slot Power Limit Scale.** This field specifies the scale used for the Slot Power Limit Value. The value of this field is set by a Set_Slot_Power_Limit Message and is only applicable for 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	**Correctable Error Reporting Enable**. This bit controls reporting of correctable errors.
1	NFEREN	RW	0x0	**Non-Fatal Error Reporting Enable**. This bit controls reporting of non-fatal errors.
2	FEREN	RW	0x0	**Fatal Error Reporting Enable**. This bit controls reporting of fatal errors.
3	URREN	RW	0x0	**Unsupported Request Reporting Enable**. This bit controls reporting of unsupported requests.
4	ERO	RO	0x0	**Enable Relaxed Ordering**. 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	**Max Payload Size.** This field sets maximum TLP payload size for the device. 0x0 -(s128) 128 bytes max payload size 0x1 -(s256) 256 bytes max payload size 0x2 -(s512) 512 bytes max payload size 0x3 -(s1024) 1024 bytes max payload size 0x4 -(s2048) 2048 bytes max payload size 0x5 -reserved (treated as 128 bytes) 0x6 -reserved (treated as 128 bytes) 0x7 -reserved (treated as 128 bytes)
8	ETFEN	RW	0x0	**Extended Tag Field Enable**. Since the bridge never generates a transaction that requires a completion, this bit has no functional effect on the device during normal operation. To aid in debug, when the SEQTAG field is set in the TLCTL register, this field controls whether tags are generated in the range from 0 through 31 or from 0 through 255.
9	PFEN	RO	0x0	**Phantom Function Enable**. The bridge does not support phantom function numbers. Therefore, this field is hardwired to zero.
10	AUXPMEN	RO	0x0	**Auxiliary Power PM Enable**. The device does not implement this capability.
11	ENS	RO	0x0	**Enable No Snoop.** The bridge does not generate transactions with the No Snoop bit set and passes transactions through the bridge with the No Snoop bit unmodified.

Notes

Bit Field	Field Name	Type	Default Value	Description
14:12	MRRS	RO	0x0	**Maximum Read Request Size.** 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	**Correctable Error Detected**. 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	**Non-Fatal Error Detected**. 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	**Fatal Error Detected**. 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	**Unsupported Request Detected**. 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	**Aux Power Detected**. Devices that require AUX power, set this bit when AUX power is detected.This device does not require AUX power, hence the value is hardwired to zero.
5	TP	RO	0x0	**Transactions Pending.** The 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.

PCIELCAP - PCI Express Link Capabilities (0x04C)

Bit Field	Field Name	Type	Default Value	Description
3:0	MAXLNK-SPD	RO	0x1	**Maximum Link Speed**. This field indicates the supported link speeds of the port. 1 - (gen1) 2.5 Gbps 2 - (gen2) 5 Gbps others - reserved

Notes

Bit Field	Field Name	Type	Default Value	Description
9:4	MAXLNK-WDTH	RWL	HWINIT	**Maximum Link Width**. This field indicates the maximum link width of the given PCI Express link. This field may be overridden to allow the link width to be forced to a smaller value. Setting this field to a invalid or reserved value results in x1 being used. The initial value of this field corresponds to the link width of the corresponding port. 0 - reserved 1 - (x1) x1 link width 2 - (x2) x2 link width 4 - (x4) x4 link width 8 - (x8) x8 link width 12 -(x12) x12 link width 16 -(x16) x16 link width 32- (x32) x32 link width others - reserved
11:10	ASPMS	RO	0x3	**Active State Power Management (ASPM) Support**. This field is hardwired to 0x3 to indicate L0s and L1 Support.
14:12	L0SEL	RWL	see text	**L0s Exit Latency**. 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 us to 2 us 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	**L1 Exit Latency**. This field indicates the L1 exit latency for the given PCI Express link. Transitioning from L1 to L0 always requires 2.3 uS. Therefore, a value 2 µs to less than 4 µs is reported with a default value of 0x2.
18	CPM	RWL	0x0	**Clock Power Management**. This bit indicates if the component tolerates removal of the reference clock via the "CLKREQ#" mechanism. The PES32T8 does not support the removal of reference clocks.
19	SDERR	RWL	Upstream: 0x0 Downstream: 0x1	**Surprise Down Error Reporting**. The PES32T8 downstream ports support surprise down error reporting.
20	DLLLA	RWL	Upstream: 0x0 Downstream: 0x1	**Data Link Layer Link Active Reporting**. The PES32T8 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.
21	LBN	RWL	Upstream: 0x0 Downstream: 0x1	**Link Bandwidth Notification Capability**. When set, this bit indicates support for the link bandwidth notification status and interrupt mechanisms. The PES32T8 downstream ports support the capability. This field is not applicable for the upstream port and must be zero. The default value of this field for all ports is 0x0 in PCIe 1.1 mode.
23:22	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
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 Number**. This field indicates the PCI express port number for the corresponding link.

PCIELCTL - PCI Express Link Control (0x050)

Bit Field	Field Name	Type	Default Value	Description
1:0	ASPM	RW	0x0	**Active State Power Management (ASPM) Control**. This field controls the level of ASPM supported by the link. The initial value corresponds to disabled. The value contained in Serial EEPROM may override this default value 0x0 - (disabled) disabled 0x1 - (l0s) L0s enable entry 0x2 - (l1) L1 enable entry 0x3 - (l0sl1) L0s and L1 enable entry
2	Reserved	RO	0x0	Reserved field.
3	RCB	RO	0x0	**Read Completion Boundary**. This field is not applicable and is hardwired to zero.
4	LDIS	RW	0x0	**Link Disable.** When set in a downstream port, this bit disables the link. 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 hardwired to zero. When the REGUNLOCK bit is set, writing a one to the LDIS bit disables the upstream link.
5	LRET	RW	0x0	**Link Retrain**. Writing a one to this field initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. This field always returns zero when read. It is permitted to set this bit while simultaneously modifying other fields in this register. When this 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 hardwired 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	**Common Clock Configuration**. 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	**Extended Sync**. 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	**Enable Clock Power Management**. The PES32T8 does not support this feature.

Notes

Bit Field	Field Name	Type	Default Value	Description
9	HWAWDTH-DIS	RW	0x0	**Hardware Autonomous Width Disable**. When set, this bit disables hardware from changing the link width for reasons other than attempting to correct for unreliable link operation by reducing the link width. This field is read-only zero in PCIe 1.1 mode.
10	LBWINTEN	RW	0x0	**Link Bandwidth Management Interrupt Enable**. When set, this bit enables the generation of an interrupt to indicate that the LBWSTS bit has been set in the PCIELSTS register. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is read-only zero in PCIe 1.1 mode.
11	LABWINTEN	RW	0x0	**Link Autonomous Bandwidth Interrupt Enable**. When set, this bit enables the generation of an interrupt to indicate that the LABWSTS bit has been set in the PCIELSTS register. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is read-only zero in PCIe 1.1 mode.
15:12	Reserved	RO	0x0	Reserved field.

PCIELSTS - PCI Express Link Status (0x052)

Bit Field	Field Name	Type	Default Value	Description
3:0	LS	RO	0x1	**Link Speed**. 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	**Link Width**. This field indicates the negotiated width of the link.
10	TERR	RO	0x0	**Training Error.** 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	**Link Training**. When set, this bit indicates that link training is in progress.
12	SCLK	RWL	HWINIT	**Slot Clock Configuration**. 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	DLLLA	RO	0x0	**Data Link Layer Link Active**. This bit indicates the status for the data link control and management state machine. This bit is always zero if the DLLLA bit in the PCIELCAP register is not set. 0x0 - (notactive) Data link layer not active state 0x1 - (active) Data link layer active state.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	LBWSTS	RW1C	0x0	**Link Bandwidth Management Status**. This bit is set to indicate that either of the following have occurred without the link transitioning through the DL_Down state. A link retraining initiated by setting the LRET bit in the PCIELCTL register has completed. The PHY has autonomously changed link speed or width to attempt to correct unreliable link operation either through an LTSSM time-out or a higher level process. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is read-only zero in PCIe 1.1 mode.
15	LABWSTS	RW1C	0x0	**Link Autonomous Bandwidth Status**. This bit is set to indicate that either that the PHY has autonomously changed link speed or width for reasons other than to attempt to correct unreliable link operation. This bit is set when a downstream switch port receives eight consecutive TS1 or TS2 ordered sets with the Autonomous Change bit set. If the LBN field in the PCIELCAP register is cleared, this field is hardwired to zero. This field is read-only zero in PCIe 1.1 mode.

PCIESCAP - PCI Express Slot Capabilities (0x054)

Bit Field	Field Name	Type	Default Value	Description
0	ABP	RWL	0x0	**Attention Button Present**. This bit is set when the Attention Button is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
1	PCP	RWL	0x0	**Power Control Present**. This bit is set when a Power Controller is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
2	MRLP	RWL	0x0	**MRL Sensor Present**. This bit is set when an MRL Sensor is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
3	ATTIP	RWL	0x0	**Attention Indicator Present**. This bit is set when an Attention Indicator is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
4	PWRIP	RWL	0x0	**Power Indicator Present**. This bit is set when an Power Indicator is implemented for the port. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
5	HPS	RWL	0x0	**Hot-Plug Surprise**. When set, this bit indicates that a device present in the slot may be removed from the system without notice. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	HPC	RWL	0x0	**Hot-Plug Capable**. This bit is set if the slot corresponding to the port is capable of supporting hot-plug operations. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
14:7	SPLV	RW	0x0	**Slot Power Limit Value**. In combination with the Slot Power Limit Scale, this field specifies the upper limit on power supplied by the slot. A Set_Slot_Power_Limit message is generated using this field whenever this register is written or when the link transitions from a 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	**Slot Power Limit Scale**. This field specifies the scale used for the Slot Power Limit Value (SPLV). 0x0 -(x1) 1.0x 0x1 -(xp1) 0.1x 0x2 -(xp01) 0.01x 0x3 -(xp001) 0.001x A Set_Slot_Power_Limit message is generated using this field whenever this register is written or when the link transitions from a 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	**Electromechanical Interlock Present**. This bit is set if an electromechanical interlock is implemented on the chassis for this slot. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.
18	NCCS	RO	0x0	**No Command Completed Support**. Software notification is always generated when an issued command is completed by the hot-plug controller. Therefore, this field is hardwired to zero.
31:19	PSLOTNUM	RWL	0x0	**Physical Slot Number**. This field indicates the physical slot number attached to this port. For devices interconnected on the system board, this field should be initialized to zero. This bit is read-only and has a value of zero when the SLOT bit in the PCIECAP register is cleared.

PCIESCTL - PCI Express Slot Control (0x058)

Bit Field	Field Name	Type	Default Value	Description
0	ABPE	RW	0x0	**Attention Button Pressed Enable**. This bit when set enables generation of a Hot-Plug interrupt or wake-up event on an attention button pressed event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
1	PFDE	RW	0x0	**Power Fault Detected Enable**. This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a power fault event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.

Notes

Bit Field	Field Name	Type	Default Value	Description
2	MRLSCE	RW	0x0	**MRL Sensor Change Enable**. This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a MRL sensor change event. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
3	PDCE	RW	0x0	**Presence Detected Changed Enable**. This bit when set enables the generation of a Hot-Plug interrupt or wake-up event on a presence detect change event.
4	CCIE	RW	0x0	**Command Complete Interrupt Enable**. This bit when set enables the generation of a Hot-Plug interrupt when a command is completed by the Hot-Plug Controller. 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	**Hot-Plug Interrupt Enable**. This bit when set enables generation of a Hot-Plug interrupt on enabled Hot-Plug events. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
7:6	AIC	RW	0x3	**Attention Indicator Control**. When read, this register returns the current state of the Attention Indicator. Writing to this register sets the indicator. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. This field is always zero if the ATTIP bit is cleared in the PCIESCAP register. 0x0 -(reserved) Reserved 0x1 -(on) On 0x2 -(blink) Blink 0x3 -(off) Off
9:8	PIC	RW	0x1	**Power Indicator Control**. When read, this register returns the current state f the Power Indicator. Writing to this register sets the indicator. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. This field is always zero if the PWRIP bit is cleared in the PCIESCAP register. 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	**Power Controller Control**. When read, this register returns the current state of the power applied to the slot. Writing to this register sets the power state of the slot. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register. 0x0 -(on) Power on 0x1 -(off) Power off

Notes

Bit Field	Field Name	Type	Default Value	Description
11	EIC	RW	0x0	**Electromechanical Interlock Control**. This field always returns a value of zero when read. If an electromechanical interlock is implemented, a write of a one to this field causes the state of the interlock to toggle and a write of a zero has no effect. This bit is read-only and has a value of zero when the corresponding capability is not enabled in the PCIESCAP register.
12	DLLLASCE	RW	0x0	**Data Link Layer Link Active State Change Enable**. 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	**Attention Button Pressed**. Set when the attention button is pressed.
1	PFD	RW1C	0x0	**Power Fault Detected**. Set when the Power Controller detects a power fault.
2	MRLSC	RW1C	0x0	**MRL Sensor Changed**. Set when an MRL Sensor state change is detected.
3	PSD	RW1C	0x0	**Presence Detected Changed**. Set when a Presence Detected change is detected.
4	CC	RW1C	0x0	**Command Completed**. This bit is set when the Hot-Plug Controller completes an issued command.
5	MRLSS	RO	0x0	**MRL Sensor State**. This field enclosed the current state of the MRL sensor. 0x0 -(closed) MRL closed 0x1 -(open) MRL open
6	PDS	RO	0x1	**Presence Detect State**. 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	**Electromechanical Interlock Status**. 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	DLLLASC	RW1C	0x0	**Data Link Layer Link Active State Change**. This bit is set when the state of the data link layer active field in the link status register changes state. 0x0 - (no change) No DLLA state change 0x1 - (changed) DLLA state change
15:9	Reserved	RO	0x0	Reserved field.

Notes

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.

PCIEDSTS2 - PCI Express Device Status 2 (0x06A)

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

PCIELCAP2 - PCI Express Link Capabilities 2 (0x06C)

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

PCIELCTL2 - PCI Express Link Control 2 (0x070)

Bit Field	Field Name	Type	Default Value	Description
3:0	TLS	RW	0x0	**Target Link Speed**. For downstream ports, this field sets an upper limit on the link operational speed by restricting the values advertised by the upstream component in its training sequences. For both upstream and downstream ports, this field is used to set the target compliance mode speed when software is using the ECOMP bit in this register to force a link into compliance mode. The PES32T8 only supports 2.5 Gbps operation. Setting this field to an unsupported value produces undefined results. 1 -(gen1) 2.5 Gbps 2 -(gen2) 5 Gbps others-reserved This field is read-only zero in PCIe 1.1 mode.
4	ECOMP	RW	0x0 Sticky	**Enter Compliance**. Software is permitted to force a link into compliance mode at the speed indicated by the TLS field by setting this bit in both components on a link and then initiating a hot reset on the link. This field is read-only zero in PCIe 1.1 mode.
5	HASD	RW	0x0	**Hardware Autonomous Speed Disable.** When set this bit prevents hardware from changing the link speed for any reason other than to correct unreliable link operation by reducing the link speed. This field is read-only zero in PCIe 1.1 mode.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:6	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.

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	**Capability ID**. The value of 0x1 identifies this capability as a PCI power management capability structure.
15:8	NXTPTR	RWL	Upstream: 0x0 Downstream: 0xD0	**Next Pointer**. 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	**Power Management Capability Version.** 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	**PME Clock.** Does not apply to PCI Express.
20	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
21	DEVSP	RWL	0x0	**Device Specific Initialization.** The value of zero indicates that no device specific initialization is required.
24:22	AUXI	RO	0x0	**AUX Current.** not used
25	D1	RO	0x0	**D1 Support.** This field indicates that the PES32T8 does not support D1.
26	D2	RO	0x0	**D2 Support.** This field indicates that the PES32T8 does not support D2.
31:27	PME	RWL	0b11001	**PME Support.** 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$_{cold}$. This functionality may be supported in the system by routing WAKE# around the switch. Modification of this field modifies the advertised capability value but does not modify the device behavior (i.e., PME is generated in the states noted in the default value).

PMCSR - PCI Power Management Control and Status (0x0C4)

Bit Field	Field Name	Type	Default Value	Description
1:0	PSTATE	RW	0x0	**Power State**. This field is used to determine the current power state and to set a new power state. 0x0 - (d0) D0 state 0x1 -(d1) D1 state (not supported by the PES32T8 and reserved) 0x2-(d2) D2 state (not supported by the PES32T8 and reserved) 0x3 -(d3) D3$_{hot}$ state
2	Reserved	RO	0x0	Reserved field.
3	NOSOFTRST	RWL	0x1	**No Soft Reset**. This bit indicates if the configuration context is preserved by the bridge when the device transitions from a D3hot to D0 power management state. 0x0 - (reset) State reset 0x1 - (preserved) State preserved
7:4	Reserved	RO	0x0	Reserved field.
8	PMEE	RW	0x0 Sticky	**PME Enable.** When this bit is set, PME message generation is enabled for the port. If a hot-plug wake-up event is desired when exiting the D3$_{cold}$ state, then this bit should be set during serial EEPROM initialization. A hot reset does not result in modification of this field.
12:9	DSEL	RO	0x0	**Data Select.** The optional data register is not implemented.
14:13	DSCALE	RO	0x0	**Data Scale.** The optional data register is not implemented.
15	PMES	RW1C	0x0 Sticky	**PME Status.** This bit is set if a PME is generated by the 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.

Notes

Bit Field	Field Name	Type	Default Value	Description
21:16	Reserved	RO	0x0	Reserved field.
22	B2B3	RO	0x0	**B2/B3 Support.** Does not apply to PCI Express.
23	BPCCE	RO	0x0	**Bus Power/Clock Control Enable.** Does not apply to PCI Express.
31:24	DATA	RO	0x0	**Data.** This optional field is not implemented.

Message Signaled Interrupt Capability Structure

MSICAP - Message Signaled Interrupt Capability and Control (0x0D0)

Bit Field	Field Name	Type	Default Value	Description
7:0	CAPID	RO	0x5	**Capability ID**. The value of 0x5 identifies this capability as a MSI capability structure.
15:8	NXTPTR	RWL	0x0	**Next Pointer**. This field contains a pointer to the next capability structure. This field is set to 0x0 indicating that it is the last capability.
16	EN	RW	0x0	**Enable.** This bit enables MSI. 0x0 -(disable) disabled 0x1 -(enable) enabled
19:17	MMC	RO	0x0	**Multiple Message Capable.** This field contains the number of requested messages.
22:20	MME	RW	0x0	**Multiple Message Enable.** Hardwired to one message.
23	A64	RO	0x1	**64-bit Address Capable.** The bridge is capable of generating messages using a 64-bit address.
31:24	Reserved	RO	0x0	Reserved field.

MSIADDR - Message Signaled Interrupt Address (0x0D4)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
31:2	ADDR	RW	0x0	**Message Address**. This field specifies the lower portion of the DWORD address of the MSI memory write transaction. The PES32T8 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

MSIUADDR - Message Signaled Interrupt Upper Address (0x0D8)

Bit Field	Field Name	Type	Default Value	Description
31:0	UADDR	RW	0x0	**Upper Message Address**. This field specifies the upper portion of the DWORD address of the MSI memory write transaction. If the contents of this field are non-zero, then 64-bit address is used in the MSI memory write transaction. If the contents of this field are zero, then the 32-bit address specified in the MSIADDR field is used. The PES32T8 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.

MSIMDATA - Message Signaled Interrupt Message Data (0x0DC)

Bit Field	Field Name	Type	Default Value	Description
15:0	MDATA	RW	0x0	**Message Data**. This field contains the lower 16-bits of data that are written when a MSI is 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	**Capability ID**. The value of 0xD identifies this capability as a SSID/SSVID capability structure.
15:8	NXTPTR	RWL	0x00	**Next Pointer**. 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	**SubSystem Vendor ID**. This field identifies the manufacturer of the add-in card or subsystem. SSVID values are assigned by the PCI-SIG to insure uniqueness.
31:16	SSID	RWL	0x0	**Subsystem ID**. This field identifies the add-in card or subsystem. SSID values are assigned by the vendor.

Notes

Extended Configuration Space Access Registers

ECFGADDR - Extended Configuration Space Access Address (0x0F8)

Bit Field	Field Name	Type	Default Value	Description
1:0	Reserved	RO	0x0	Reserved field.
7:2	REG	RW	0x0	**Register Number.** This field selects the configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 1.1
11:8	EREG	RW	0x0	**Extended Register Number.** This field selects the extended configuration register number as defined by Section 7.2.2 of the PCI Express Base Specification, Rev. 1.1
31:12	Reserved	RO	0x0	Reserved field.

ECFGDATA - Extended Configuration Space Access Data (0x0FC)

Bit Field	Field Name	Type	Default Value	Description
31:0	DATA	RW	0x0	**Configuration Data.** A read from this field will return the configuration space register value pointed to by the ECFGADDR register. A write to this field will update the contents of the configuration space register pointed to by the ECFGADDR register with the value written. For both reads and writes, the byte enables correspond to those used to access this field. 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	**Capability ID.** The value of 0x1 indicates an advanced error reporting capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x200	**Next Pointer.**

Notes

AERUES - AER Uncorrectable Error Status (0x104)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW1C	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW1C	0x0 Sticky	**Data Link Protocol Error Status.** This bit is set when a data link layer protocol error is detected.
5	SDOENERR	RW1C	0x0 Sticky	**Surprise Down Error Status.** This bit is set when a surprise down 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	**Poisoned TLP Status.** This bit is set when a poisoned TLP is detected.
13	FCPERR	RW1C	0x0 Sticky	**Flow Control Protocol Error Status.** This bit is set when a flow control protocol error is detected.
14	COMPTO	RO	0x0	**Completion Time-out Status.** A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	**Completer Abort Status.** The PES32T8 never responds to a non-posted request with a completer abort.
16	UECOMP	RW1C	0x0 Sticky	**Unexpected Completion Status.** This bit is set when an unexpected completion is detected.
17	RCVOVR	RW1C	0x0 Sticky	**Receiver Overflow Status.** This bit is set when a receiver overflow is detected.
18	MALFORMED	RW1C	0x0 Sticky	**Malformed TLP Status.** This bit is set when a malformed TLP is detected.
19	ECRC	RW1C	0x0 Sticky	**ECRC Status.** This bit is set when an ECRC error is detected.
20	UR	RW1C	0x0 Sticky	**UR Status.** This bit is set when an unsupported request is detected.
31:21	Reserved	RO	0x0	Reserved field.

AERUEM - AER Uncorrectable Error Mask (0x108)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x0 Sticky	**Data Link Protocol Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.

Notes

Bit Field	Field Name	Type	Default Value	Description
5	SDOENERR	RW	0x0 Sticky	**Surprise Down Error Status.** 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	**Poisoned TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
13	FCPERR	RW	0x0 Sticky	**Flow Control Protocol Error Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
14	COMPTO	RO	0x0	**Completion Time-out Status.** A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	**Completer Abort Status.** The PES32T8 never responds to a non-posted request with a completer abort.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
17	RCVOVR	RW	0x0 Sticky	**Receiver Overflow Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
18	MAL-FORMED	RW	0x0 Sticky	**Malformed TLP Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
19	ECRC	RW	0x0 Sticky	**ECRC Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
20	UR	RW	0x0 Sticky	**UR Mask.** When this bit is set, the corresponding bit in the AERUES register is masked. When a bit is masked in the AERUES register, the corresponding event is not logged in the advanced capability structure and an error is not reported to the root complex.
31:21	Reserved	RO	0x0	Reserved field.

Notes

AERUESV - AER Uncorrectable Error Severity (0x10C)

Bit Field	Field Name	Type	Default Value	Description
0	UDEF	RW	0x0 Sticky	**Undefined.** This bit is no longer used in this version of the specification.
3:1	Reserved	RO	0x0	Reserved field.
4	DLPERR	RW	0x1 Sticky	**Data Link Protocol Error Severity.** 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	**Surprise Down Error Status.** 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	**Poisoned TLP Status Severity.** 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	**Flow Control Protocol Error Severity.** 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	**Completion Time-out Status.** A switch port does not initiate non-posted requests on its own behalf. Therefore, this field is hardwired to zero.
15	CABORT	RO	0x0	**Completer Abort Status.** The PES32T8 never responds to a non-posted request with a completer abort.
16	UECOMP	RW	0x0 Sticky	**Unexpected Completion Severity.** 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	**Receiver Overflow Severity.** 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	**Malformed TLP Severity.** 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.

Notes

Bit Field	Field Name	Type	Default Value	Description
19	ECRC	RW	0x0 Sticky	**ECRC Severity.** 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	**UR Severity.** 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	**Receiver Error Status.** This bit is set when the physical layer detects a receiver error.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW1C	0x0 Sticky	**Bad TLP Status.** This bit is set when a bad TLP is detected.
7	BADDLLP	RW1C	0x0 Sticky	**Bad DLLP Status.** This bit is set when a bad DLLP is detected.
8	RPLYROVR	RW1C	0x0 Sticky	**Replay Number Rollover Status.** This bit is set when a replay number rollover has occurred indicating that the data link layer has abandoned replays and has requested that the link be retrained.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW1C	0x0 Sticky	**Replay Timer Time-Out Status.** This bit is set when the replay timer in the data link layer times out.
13	ADVISO-RYNF	RW1C	0x0 Sticky	**Advisory Non-Fatal Error Status.** This bit is set when an advisory non-fatal error is detected as described in Section 6.2.3.2.4 of the 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	**Receiver Error Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.
5:1	Reserved	RO	0x0	Reserved field.
6	BADTLP	RW	0x0 Sticky	**Bad TLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.

Notes

Bit Field	Field Name	Type	Default Value	Description
7	BADDLLP	RW	0x0 Sticky	**Bad DLLP Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.
8	RPLYROVR	RW	0x0 Sticky	**Replay Number Rollover Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.
11:9	Reserved	RO	0x0	Reserved field.
12	RPLYTO	RW	0x0 Sticky	**Replay Timer Time-Out Mask.** When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.
13	ADVISORYNF	RW	0x1 Sticky	**Advisory Non-Fatal Error Status.**When this bit is set, the corresponding bit in the AERCES register is masked. When a bit is masked in the AERCES register, the corresponding event is not reported to the root complex.
31:14	Reserved	RO	0x0	Reserved field.

AERCTL - AER Control (0x118)

Bit Field	Field Name	Type	Default Value	Description
4:0	FEPTR	RO	0x0 Sticky	**First Error Pointer.** This field contains a pointer to the bit in the AERUES register that resulted in the first reported error.
5	ECRCGC	RWL	0x1	**ECRC Generation Capable.** This bit indicates if the device is capable of generating ECRC.
6	ECRCGE	RW	0x0 Sticky	**ECRC Generation Enable.** When this bit is set, ECRC generation is enabled.
7	ECRCCC	RWL	0x1	**ECRC Check Capable.** This bit indicates if the device is capable of checking ECRC.
8	ECRCCE	RW	0x0 Sticky	**ECRC Check Enable.** 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	**Header Log.** This field contains the 1st doubleword of the TLP header that resulted in the first reported uncorrectable error.

AERHL2DW - AER Header Log 2nd Doubleword (0x120)

Bit Field	Field Name	Type	Default Value	Description
31:0	HL	RO	0x0 Sticky	**Header Log.** This field contains the 2nd doubleword of the TLP header that resulted in the first reported uncorrectable error.

Notes

AERHL3DW - AER Header Log 3rd Doubleword (0x124)

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

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

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

Device Serial Number Enhanced Capability

SNUMCAP - Serial Number Capabilities (0x180)

Bit Field	Field Name	Type	Default Value	Description
15:0	CAPID	RO	0x3	**Capability ID.** The value of 0x3 indicates a device serial number capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1 indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x0	**Next Pointer.**

SNUMLDW - Serial Number Lower Doubleword (0x184)

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

SNUMUDW - Serial Number Upper Doubleword (0x188)

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

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	**Capability ID.** The value of 0x2. indicates a virtual channel capability structure.
19:16	CAPVER	RO	0x1	**Capability Version.** The value of 0x1. indicates compatibility with version 1 of the specification.
31:20	NXTPTR	RWL	0x0	**Next Pointer.** 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	0x0	**Extended VC Count.** The value 0x0 indicates only implementation of the default VC.
3	Reserved	RO	0x0	Reserved field.
6:4	LPEVCCNT	RO	0x0	**Low Priority Extended VC Count.** The value of 0x0 indicates only implementation of the default VC.
7	Reserved	RO	0x0	Reserved field.
9:8	REFCLK	RO	0x0	**Reference Clock.** WRR is not implemented.
11:10	PATBLSIZ	RO	Upstream: 0x2 Downstream: 0x0	**Port Arbitration Table Entry Size.** This field indicates the size of the port arbitration table in the device. For the upstream port, the is set to 0x2 to indicate a table with 4-bit entries. For downstream ports, this value is set to 0x0. 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	**VC Arbitration Capability.** 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) bit 1 - weighted round robin (WRR) with 32 phases bit 2 - weighted round robin (WRR) with 64 phases bit 3 - weighted round robin (WRR) with 128 phases bits 4 through 7 - reserved
23:8	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
31:24	VCATBLOFF	RO	0x0	**VC Arbitration Table Offset.** 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	**Load VC Arbitration Table.** 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	VCARBSEL	RO	0x0	**VC Arbitration Select.** 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). Since the device supports only VC0, this field has no functional effect.
31:4	Reserved	RO	0x0	Reserved field.

PVCSTS - Port VC Status (0x20E)

Bit Field	Field Name	Type	Default Value	Description
0	VCATS	RO	0x0	**VC Arbitration Table Status.** This bit indicates the coherency status of the VC arbitration table. Since the device supports only VC0, this field has no functional effect and is always zero.
31:1	Reserved	RO	0x0	Reserved field.

VCR0CAP - VC Resource 0 Capability (0x210)

Bit Field	Field Name	Type	Default Value	Description
7:0	PARBC	RO	Upstream: 0x3 Downstream: 0x1	**Port Arbitration Capability.** 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. The upstream port supports hardware fixed round robin and weighted round robin with 32 phases. Downstream ports support only hardware fixed round robin. bit 0 - hardware fixed round robin bit 1 - weighted round robin with 32 phases bit 2 - weighted round robin with 64 phases bit 3 - weighted round robin with 128 phases bit 4 - time-based weighted round robin with 128 phases bit 5 - weighted round robin with 256 phases
13:8	Reserved	RO	0x0	Reserved field.

Notes

Bit Field	Field Name	Type	Default Value	Description
14	APS	RO	0x0	**Advanced Packet Switching.** Not supported.
15	RJST	RO	0x0	**Reject Snoop Transactions.** No supported for switch ports.
22:16	MAXTS	RO	0x0	**Maximum Time Slots.** Since this VC does not support time-based WRR, this field is not valid.
23	Reserved	RO	0x0	Reserved field.
31:24	PATBLOFF	RO	Upstream: 0x2 Downstream: 0x0	**Port Arbitration Table Offset.** 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). The upstream port has a port arbitration table. Downstream ports do not have a port arbitration table.

VCR0CTL- VC Resource 0 Control (0x214)

Bit Field	Field Name	Type	Default Value	Description
7:0	TCVCMAP	bit 0 RO bits 1 through 7 RW	0xFF	**TC/VC Map.** This field indicates the TCs that are mapped to the VC resource. Each bit corresponds to a TC. When a bit is set, the corresponding TC is mapped to the VC.
15:8	Reserved	RO	0x0	Reserved field.
16	LPAT	RW	0x0	**Load Port Arbitration Table.** 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 only has an effect in the upstream port. This bit always returns 0 when read.
19:17	PARBSEL	RW	0x0	**Port Arbitration Select.** 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 t he Port Arbitration Capability field of the VC resource.
23:20	Reserved	RO	0x0	Reserved field.
26:24	VCID	RO	0x0	**VC ID.** This field assigns a VC ID to the VC resource. Since the PES32T8 implements only a single VC, this field is hardwired to zero.
30:27	Reserved	RO	0x0	Reserved field.
31	VCEN	RO	0x1	**VC Enable.** This field, when set, enables a virtual channel. Since the PES32T8 implements only a single VC, this field is hardwired to one (enabled).

Notes

VCR0STS - VC Resource 0 Status (0x218)

Bit Field	Field Name	Type	Default Value	Description
15:0	Reserved	RO	0x0	Reserved field.
16	PATS	RO	0x0	**Port Arbitration Table Status.** 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. This field is always zero for downstream ports.
17	VCNEG	RO	0x0	**VC Negotiation Pending.** Since the PES32T8 implements only a single VC (i.e., the default VC) this field indicates the status of the process of flow control initialization.
31:18	Reserved	RO	0x0	Reserved field.

VCR0TBL0 - VC Resource 0 Arbitration Table Entry 0 (0x220)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE0	RW	0x1	**Phase 0.** 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 through 0xF - reserved
7:4	PHASE1	RW	0x2	**Phase 1.** This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE2	RW	0x3	**Phase 2.** This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE3	RW	0x4	**Phase 3.** This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE4	RW	0x5	**Phase 4.** This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE5	RW	0x6	**Phase 5.** This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE6	RW	0x7	**Phase 6.** This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE7	RW	0x1	**Phase 7.** This field contains the port ID for the corresponding port arbitration period.

Notes

VCR0TBL1 - VC Resource 0 Arbitration Table Entry 1 (0x224)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE8	RW	0x2	**Phase 8.** This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE9	RW	0x3	**Phase 9.** This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE10	RW	0x4	**Phase 10.** This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE11	RW	0x5	**Phase 11.** This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE12	RW	0x6	**Phase 12.** This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE13	RW	0x7	**Phase 13.** This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE14	RW	0x1	**Phase 14.** This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE15	RW	0x2	**Phase 15.** This field contains the port ID for the corresponding port arbitration period.

VCR0TBL2 - VC Resource 0 Arbitration Table Entry 2 (0x228)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE16	RW	0x3	**Phase 16.** This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE17	RW	0x4	**Phase 17.** This field contains the port ID for the corresponding port arbitration period.
11:8	PHASE18	RW	0x5	**Phase 18.** This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE19	RW	0x6	**Phase 19.** This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE20	RW	0x7	**Phase 20.** This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE21	RW	0x1	**Phase 21.** This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE22	RW	0x2	**Phase 22.** This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE23	RW	0x3	**Phase 23.** This field contains the port ID for the corresponding port arbitration period.

VCR0TBL3 - VC Resource 0 Arbitration Table Entry 3 (0x22C)

Bit Field	Field Name	Type	Default Value	Description
3:0	PHASE24	RW	0x4	**Phase 24.** This field contains the port ID for the corresponding port arbitration period.
7:4	PHASE25	RW	0x5	**Phase 25.** This field contains the port ID for the corresponding port arbitration period.

Notes

Bit Field	Field Name	Type	Default Value	Description
11:8	PHASE26	RW	0x6	**Phase 26.** This field contains the port ID for the corresponding port arbitration period.
15:12	PHASE27	RW	0x7	**Phase 27.** This field contains the port ID for the corresponding port arbitration period.
19:16	PHASE28	RW	0x1	**Phase 28.** This field contains the port ID for the corresponding port arbitration period.
23:20	PHASE29	RW	0x2	**Phase 29.** This field contains the port ID for the corresponding port arbitration period.
27:24	PHASE30	RW	0x3	**Phase 30.** This field contains the port ID for the corresponding port arbitration period.
31:28	PHASE31	RW	0x4	**Phase 31.** 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	**Capability ID.** 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	**Capability Version.** 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	**Next Pointer.**

PWRBDSEL - Power Budgeting Data Select (0x284)

Bit Field	Field Name	Type	Default Value	Description
7:0	DVSEL	RW	0x0	**Data Value Select.** 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	**Data.** 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	**System Allocated.** 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[0..7] - Power Budgeting Data Value [0..7] (0x300)

Bit Field	Field Name	Type	Default Value	Description
31:0	DV	RW	Undefined Sticky	**Data Value.** 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 Switch Control (SWCTL) 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.

Switch Control and Status Registers

SWSTS - Switch Status (0x400)

Bit Field	Field Name	Type	Default Value	Description
3:0	SWMODE	RO	HWINIT	**Switch Mode.** These configuration pins determine the PES32T8 switch operating mode. These pins should be static and not change after the negation of PERSTN. 0x0 - Normal switch mode 0x1 - Normal switch mode with Serial EEPROM initialization 0x2 - through 0xF Reserved
4	Reserved	RO	0x0	Reserved field.
5	CCLKDS	RO	HWINIT	**Common Clock Downstream.** This bit reflects the value of the CCLKDS signal sampled during the fundamental reset.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	CCLKUS	RO	HWINIT	**Common Clock Upstream.** This bit reflects the value of the CCLKUS signal sampled during the fundamental reset.
7	MSMB-SMODE	RO	HWINIT	**Master SMBus Slow Mode.** This bit reflects the value of the MSMBSMODE signal sampled during the fundamental reset.
8	REFCLKM	RO	HWINIT	**PCI Express Reference Clock Mode Select.** This bit reflects the value of the REFCLKM signal sampled during the fundamental reset.
9	RSTHALT	RO	HWINIT	**Reset Halt.** This bit reflects the value of the RSTHALT signal sampled during the fundamental reset.
10:11	PEMODE	RWL	HWINIT	**PCI Express Base Specification Mode.** This field selects the PCIe base specification operating mode for the PES32T8. 0x0 - reserved 0x1 - (pebase1p1) PCIe 1.1 base specification compliant mode. 0x2 - reserved 0x3 - reserved
12	P01MERGE	RO	0x0	**Port 0 and 1 Merge.** This bit reflects the value of the P01MERGE signal sampled during the fundamental reset.
13	P23MERGE	RO	0x0	**Port 2 and 3 Merge.** This bit reflects the value of the P23MERGE signal sampled during the fundamental reset.
14	P45MERGE	RO	0x0	**Port 4 and 5 Merge.** This bit reflects the value of the P45MERGE signal sampled during the fundamental reset.
15	P67MERGE	RO	0x0	**Port 6 and 7 Merge.** This bit reflects the value of the P67MERGE signal sampled during the fundamental reset.
19:16	Reserved	RO	0x0	Reserved field.
22:20	LOCKMODE	RO	0x0	**Lock Mode.** 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 -(port7ocked) Upstream port is locked with port 7.
27:23	Reserved	RO	0x0	Reserved field.
31:28	MARKER	RW	0x0 Sticky	**Marker.** 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.

SWCTL - Switch Control (0x404)

Bit Field	Field Name	Type	Default Value	Description
0	FRST	RW	0x0	**Fundamental Reset.** 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 2-5 for the behavior of this bit.

Notes

Bit Field	Field Name	Type	Default Value	Description
1	HRST	RW	0x0	**Hot Reset.** 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 2-6 for the behavior of this bit.
2	RSTHALT	RW	HWINIT Sticky	**Reset Halt.** 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	REGUN-LOCK	RW	0x0 Sticky	**Register Unlock.** When this bit is set, the contents of registers and fields of type Read and Write when Unlocked (RWL) are modified when written 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	**Power Budgeting Data Value Unlock.** 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	DLDHRST	RW	0x0 Sticky	**Disable Link Down Hot Reset.** 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	**Disable Hot Reset Serial EEPROM Initialization.** When this bit is set, step 6 “serial EEPROM initialization” is skipped in the hot reset sequence described in section Hot Reset on page 2-6 regardless of the selected switch operating mode.
7	DRO	RW	0x0 Sticky	**Disable Relaxed Ordering.** 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.
8	DP2P	RW	0x0 Sticky	**Disable Peer-to-Peer Transactions.** When this bit is set, all peer-to-peer transactions are disabled. In this mode, transactions received on a downstream port which are not destined to the upstream port are treated as an unsupported requests.
13:9	Reserved	RO	0x0	Reserved field.
14	CTDIS	RW	0x0 Sticky	**Disable Cut-Through Routing.** When this bit is set, cut through routing of TLPs is disabled between all ports (i.e., they are routed in a stored and forwarded manner). When this bit is cleared, TLPs are routed in a cut-through manner when possible.

Notes

Bit Field	Field Name	Type	Default Value	Description
15	LOCKIGNORE	RW	0x0 Sticky	**Ignore Locked Transactions** When this bit is set, all bus locking side-effects associated with locked transactions (e.g., MRdLk) are ignored and the TLPs are treated by the PES32T8 as normal TLPs (e.g., are routed normally through the switch).
31:16	Reserved	RO	0x0	Reserved field.

HPCFGCTL - Hot-Plug Configuration Control (0x408)

Bit Field	Field Name	Type	Default Value	Description
0	IPXAPN	RW	0x0 Sticky	**Invert Polarity of PxAPN.** When this bit is set, the polarity of the PxAPN input is inverted in all ports.
1	IPXPDN	RW	0x0 Sticky	**Invert Polarity of PxPDN.** When this bit is set, the polarity of the PxPDN input is inverted in all ports.
2	IPXPFN	RW	0x0 Sticky	**Invert Polarity of PxPFN.** When this bit is set, the polarity of the PxPFN input is inverted in all ports.
3	IPXMRLN	RW	0x0 Sticky	**Invert Polarity of PxMRLN.** When this bit is set, the polarity of the PxMRLN input is inverted in all ports.
4	IPXAIN	RW	0x0 Sticky	**Invert Polarity of PxAIN.** When this bit is set, the polarity of the PxAIN output is inverted in all ports.
5	IPXPIN	RW	0x0 Sticky	**Invert Polarity of PxPIN.** When this bit is set, the polarity of the PxPIN output is inverted in all ports.
6	IPXPEP	RW	0x0 Sticky	**Invert Polarity of PxPEP.** When this bit is set, the polarity of the PxPEP output is inverted in all ports.
7	IPXILOCKP	RW	0x0 Sticky	**Invert Polarity of PxILOCKP.** When this bit is set, the polarity of the PxILOCKP output is inverted in all ports.
8	IPXPWRGDN	RW	0x0 Sticky	**Invert Polarity of PxPWRGDN.** When this bit is set, the polarity of the PxPWRGDN input is inverted in all ports.
10:9	Reserved	RW	0x0	Reserved field.
11	MRLPWROFF	RW	0x1 Sticky	**When the MRL Automatic Power Off**. When this bit is set and the Manual Retention Latch Present (MRLP) bit is set in the PCI Express Slot Capability (PCIESCAP) register, then power to the slot is automatically turned off when the MRL sensor indicates that the MRL is open. This occurs regardless of the state of the Power Controller Control (PCC) bit in the PCI Express Slot Control (PCIESCTL) register.
12	RMRLWEMIL	RW	0x0 Sticky	**Replace MRL Status with EMIL Status.** When this bit is set, the PxMRLN signal inputs are used as electromechanical lock state inputs.
13	TEMICTL	RW	0x0 Sticky	**Toggle Electromechanical Interlock Control.** When this bit is cleared, the Electromechanical Interlock (PxILOCKP) 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 PxILOCKP output.

Notes

Bit Field	Field Name	Type	Default Value	Description
15:14	RSTMODE	RW	0x0 Sticky	**Reset Mode.** 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	**Slot Power to Reset Negation.** 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	**Reset Negation.** 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	**Data.** General purpose register available for software use.

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

Bit Field	Field Name	Type	Default Value	Description
15:0	GPIOFUNC	RW	0x0 Sticky	**GPIO Function.** 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.
31:16	Reserved	RO	0x0	Reserved field.

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

Bit Field	Field Name	Type	Default Value	Description
15:0	GPIOCFG	RW	0x0 Sticky	**GPIO Configuration.** Each bit in this field controls the corresponding GPIO pin. When a bit is configured as a general purpose I/O pin and the corresponding bit in this field is set, then the pin is configured as a GPIO output. When a bit is configured as a general purpose I/O pin and the corresponding bit in this field is 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.
31:16	Reserved	RO	0x0	Reserved field.

Notes

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

Bit Field	Field Name	Type	Default Value	Description
15:0	GPIOD	RW	HWINIT Sticky	**GPIO Data.** Each bit in this field controls the corresponding GPIO pin. Reading this field returns the current value of each GPIO pin regardless of GPIO pin mode (i.e., alternate function or GPIO pin). Writing a value to this field causes the corresponding pins which are configured as GPIO outputs to change state to the value written.
31:16	Reserved	RO	0x0	Reserved field.

SMBUSSTS - SMBus Status (0x424)

Bit Field	Field Name	Type	Default Value	Description
0	Reserved	RO	0x0	Reserved field.
7:1	SSMBADDR	RO	HWINIT	**Slave SMBus Address.** This field contains the SMBus address assigned to the slave SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	MSMBADDR	RO	HWINIT	**Master SMBus Address.** This field contains the SMBus address assigned to the master SMBus interface.
23:16	Reserved	RO	0x0	Reserved field.
24	EEPROM-DONE	RO	0x0	**Serial EEPROM Initialization Done.** 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	**No Acknowledge Error.** This bit is set if an unexpected NACK is observed during a master SMBus transaction. The setting of this bit may indicate the following: that the addressed device does not exist on the SMBus (i.e., addressing error); data is unavailable or the device is busy; an invalid command was detected by the slave; or invalid data was detected by the slave.
26	LAERR	RW1C	0x0	**Lost Arbitration Error.** When the master SMBus interface loses arbitration for the SMBus, it automatically re-arbitrates for the SMBus. If the master SMBus interface loses 16 consecutive arbitration attempts, then the transaction is aborted and this bit is set.
27	OTHERERR	RW1C	0x0	**Other Error.** This bit is set if a misplaced START or STOP condition is detected by the master SMBus interface.
28	ICSERR	RW1C	0x0	**Initialization Checksum Error.** This bit is set if an invalid checksum is computed during Serial EEPROM initialization or when a configuration done command is not found in the serial EEPROM.
29	URIA	RW1C	0x0	**Unmapped Register Initialization Attempt.** 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.

Notes

SMBUSCTL - SMBus Control (0x428)

Bit Field	Field Name	Type	Default Value	Description
15:0	MSMBCP	RW	HWINIT Sticky	**Master SMBus Clock Prescalar.** This field contains a clock prescalar value used during master SMBus transactions. The prescalar clock period is equal to 32 ns multiplied by the value in this field. When the field is cleared to zero or one, the clock is stopped. The initial value of this field is 0x0139 when the master SMBus is configured to operate in slow mode (i.e., 100 KHz) in the boot configuration and to 0x0053[1] when it is configured to operate in fast mode (i.e., 400 KHz).
16	MSMBIOM	RW	0x0 Sticky	**Master SMBus Ignore Other Masters.** When this bit is set, the master SMBus proceeds with transactions regardless of whether it won or lost arbitration.
17	ICHECKSUM	RW	0x0 Sticky	**Ignore Checksum Errors.** When this bit is set, serial EEPROM initialization checksum errors are ignored (i.e., the checksum always passes).
19:18	SSMBMODE	RW	0x0 Sticky	**Slave SMBus Mode.** The salve SMBus contains internal glitch counters on the SSMBCLK and SSMBDAT signals that wait approximately 1uS before sampling or driving these signals. This field allows the glitch counter time to be reduced or entirely removed. In some systems, this may permit high speed slave SMBus operation. 0x0 -(normal) Slave SMBus normal mode. Glitch counters operate with 1uS delay. 0x1 -(fast) Slave SMBus interface fast mode. Glitch counters operate with 100nS delay. 0x2 -(disabled) Slave SMBus interface with glitch counters disabled. Glitch counters operate with zero delay which effectively removes them. 0x3 -reserved.
21:20	MSMBMODE	RW	0x0 Sticky	**Master SMBus Mode.** The master SMBus contains internal glitch counters on the MSMBCLK and MSMBDAT signals that wait approximately 1uS before sampling or driving these signals. This field allows the glitch counter time to be reduced or entirely removed. In some systems, this may permit high speed master SMBus operation. 0x0 -(normal) Master SMBus normal mode. Glitch counters operate with 1uS delay. 0x1 -(fast) Master SMBus interface fast mode. Glitch counters operate with 100nS delay. 0x2 -(disabled) Master SMBus interface with glitch counters disabled. Glitch counters operate with zero delay which effectively removes them. 0x3 -reserved.
31:22	Reserved	RO	0x0	Reserved field.

[1] The MSMBCLK low minimum pulse width is equal to half the period programmed in this field. The value of 0x53, which corresponds to ~373 KHz, allows the min low pulse width to be satisfied. In systems where this timing parameter is not critical, the operating frequency may be increased.

Notes

EEPROMINTF - Serial EEPROM Interface (0x42C)

Bit Field	Field Name	Type	Default Value	Description
15:0	ADDR	RW	0x0	**EEPROM Address.** This field contains the byte address in the Serial EEPROM to be read or written.
23:16	DATA	RW	0x0	**EEPROM Data.** A write to this field will initiates a serial EEPROM read or write operation, as selected by the OP field, to the address specified in the ADDR field. When a write operation is selected, the value written to this field is the value written to the serial EEPROM. When a read operation is selected, the value written to this field is ignored and the value read from the serial EEPROM may be read from this field when the DONE bit is set.
24	BUSY	RO	0x0	**EEPROM Busy.** This bit is set when a serial EEPROM read or write operation is in progress. 0x0 -(idle) serial EEPROM interface idle 0x1 -(busy) serial EEPROM interface operation in progress
25	DONE	RW1C	0x0	**EEPROM Operation Completed.** This bit is set when a serial EEPROM operation has completed. 0x0 - (not done) interface is idle or operation in progress 0x1 - (done) operation completed
26	OP	RW	0x0	**EEPROM Operation Select.** This field selects the type of EEPROM operation to be performed when the DATA field is written 0x0 -(write) serial EEPROM write 0x1 -(read) serial EEPROM read
31:27	Reserved	RO	0x0	Reserved field.

IOEXPINTF - I/O Expander Interface (0x430)

Bit Field	Field Name	Type	Default Value	Description
15:0	IOEDATA	RW	0x0	**I/O Expander Data.** 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 and RELOADIOEX bits are 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	RELOADIOEX	RW	0x0	**Reload I/O Expander Signals.** 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 IOEDATA field. This bit always returns a zero when read.

Notes

Bit Field	Field Name	Type	Default Value	Description
25	IOEXTM	RW	0x0	**IO Expander Test Mode.** 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 PES32T8 core are ignored and values supplied to the I/O expander correspond to value written to the IOEDATA field.
29:26	SELECT	RO	0x0	**I/O Expander Select.** 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 through 0x7 - reserved
30	Reserved	RO	0x0	Reserved field.
31	DONE	RW1C	0x0	**I/O Expander Operation Done**. This bit is set when any of the following conditions occurs: - 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	**I/O Expander 0 Address.** This field contains the SMBus address assigned to I/O expander 0 on the master SMBus interface.
8	Reserved	RO	0x0	Reserved field.
15:9	IOE1ADDR	RWL	0x0 Sticky	**I/O Expander 1 Address.** This field contains the SMBus address assigned to I/O expander 1 on the master SMBus interface.
16	Reserved	RO	0x0	Reserved field.
23:17	IOE2ADDR	RWL	0x0 Sticky	**I/O Expander 2 Address.** This field contains the SMBus address assigned to I/O expander 2 on the master SMBus interface.
24	Reserved	RO	0x0	Reserved field.
31:25	IOE3ADDR	RWL	0x0 Sticky	**I/O Expander 3 Address.** This field contains the SMBus address assigned to I/O expander 3 on the master SMBus interface.

Notes

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	**I/O Expander 4 Address.** This field contains the SMBus address assigned to I/O expander 4 on the master SMBus interface.
31:8	Reserved	RO	0x0	Reserved field.

GPECTL - General Purpose Event Control (0x450)

Bit Field	Field Name	Type	Default Value	Description
0	IGPE	RW	0x0 Sticky	**Invert General Purpose Event Enable Signal Polarity.** When this bit is set, the polarity of all General Purpose Event (GPEN) signals is inverted. 0x0 - (normal) GPEN signals are active low 0x1 - (invert) GPEN signals are active high
1	P1GPEE	RW	0x0 Sticky	**Port 1 General Purpose Event Enable.** 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	**Port 2 General Purpose Event Enable.** 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	**Port 3 General Purpose Event Enable.** 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	**Port 4 General Purpose Event Enable.** 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	**Port 5 General Purpose Event Enable.** 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].
6	P6GPEE	RW	0x0 Sticky	**Port 6 General Purpose Event Enable.** 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].

Notes

Bit Field	Field Name	Type	Default Value	Description
7	P7GPEE	RW	0x0 Sticky	**Port 7 General Purpose Event Enable.** 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].
31:8	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	**Port 1 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
2	P2GPES	RO	0x0	**Port 2 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
3	P3GPES	RO	0x0	**Port 3 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
4	P4GPES	RO	0x0	**Port 4 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
5	P5GPES	RO	0x0	**Port 5 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.

Notes

Bit Field	Field Name	Type	Default Value	Description
6	P6GPES	RO	0x0	**Port 6 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
7	P7GPES	RO	0x0	**Port 7 General Purpose Event Status.** 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[7] and GPIO[7] is asserted only if enabled to operate as an alternate function.
31:8	Reserved	RO	0x0	Reserved field.

UARBTC - U-Bus Arbiter Transfer Count (0x458)

Bit Field	Field Name	Type	Default Value	Description
7:0	D2UTC	RW	0x01 Sticky	**Downstream to Upstream Transfer Count.** This field contains the downstream to upstream transfer count. The D2UCTC field in the UARBCTC register is set to this value after each arbitration period.
15:8	P2PTC	RW	0x01 Sticky	**Peer to Peer Transfer Count.** This field contains the peer to peer transfer count. The P2PCTC field in the UARBCTC register is set to this value after each arbitration period.
23:16	U2STC	RW	0x01 Sticky	**Upstream to Self Transfer Count.** This field contains the upstream to self transfer count. The U2SCTC field in the UARBCTC register is set to this value after each arbitration period.
31:24	D2STC	RW	0x01 Sticky	**Downstream to Self Transfer Count.** This field contains the downstream to self transfer count. The D2SCTC field in the UARBCTC register is set to this value after each arbitration period.

UARBCTC - U-Bus Arbiter Current Transfer Count (0x45C)

Bit Field	Field Name	Type	Default Value	Description
7:0	D2UCTC	RO	0x01	**Downstream to Upstream Current Transfer Count.** This field contains the current downstream to upstream transfer count.
15:8	P2PCTC	RO	0x01	**Peer to Peer Current Transfer Count.** This field contains the current peer to peer transfer count.
23:16	U2SCTC	RO	0x01	**Upstream to Self Current Transfer Count.** This field contains the current upstream to self transfer count.
31:24	D2SCTC	RO	0x01	**Downstream to Downstream Current Transfer Count.** This field contains the current downstream to self transfer count.

Notes

DARBTC - D-Bus Arbiter Transfer Count (0x460)

Bit Field	Field Name	Type	Default Value	Description
7:0	U2DTC	RW	0x01 Sticky	**Upstream to Downstream Transfer Count.** This field contains the upstream to downstream transfer count. The U2DCTC field in the DARBCTC register is set to this value after each arbitration period.
15:8	BDQ2DTC	RW	0x01 Sticky	**Bus Decoupler Queue to Downstream Transfer Count.** This field contains the bus decoupler queue to downstream transfer count. The BDQ2DTC field in the DARBCTC register is set to this value after each arbitration period.
31:16	Reserved	RO	0x0	Reserved field.

DARBCTC - D-Bus Arbiter Current Transfer Count (0x464)

Bit Field	Field Name	Type	Default Value	Description
7:0	U2DCTC	RO	0x01	**Upstream to Downstream Current Transfer Count.** This field contains the current upstream to downstream transfer count.
15:8	BDQ2DTC	RO	0x01	**Bus Decoupler Queue to Downstream Current Transfer Count.** This field contains the current bus decoupler queue to downstream transfer count.
31:16	Reserved	RO	0x0	Reserved field.

SWCTCTL - Switch Cut Through Control (0x46C)

Bit Field	Field Name	Type	Default Value	Description
0	P01CTDIS	RW	0x1 Sticky	**Port 0 and 1 Cut-Through Disable.** When this bit is set, cut through routing of TLPs is disabled between ports 0 and 1 (i.e., they are routed in a stored and forwarded manner). When this bit is cleared, TLPs are routed in a cut-through manner. This bit has no effect on any port other than zero and one.
31:1	Reserved	RO	0x0	Reserved field.

SWTSCNTCTL - Switch Time-Stamp Counter Control (0x4A8)

Bit Field	Field Name	Type	Default Value	Description
31:0	PRESETVAL	RW	0x0	**Time-Stamp Preset Value.** A write to this register will cause bits 34 through three of the time-stamp counter associated with all ports to take on the value written and bits two through zero to be set to zero.

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	**Disable End-to-End Parity Checking.** 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	**Generate Bad End-to-End Parity.** 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	**Error Match Length.** 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	**End-to-End Parity Error.** This bit is set when an end-to-end parity error is detected at the port.
31:1	Reserved	RO	0x0	Reserved field.

SWPERCTL - Switch Parity Error Reporting Control (0x748)

Bit Field	Field Name	Type	Default Value	Description
1:0	EEPE	RW	0x2	**End-to-End Parity Error Reporting.** 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.

Notes

SWPECNT - Switch Parity Error Count (0x74C)

Bit Field	Field Name	Type	Default Value	Description
7:0	EEPEC	RCW	0x0 Sticky	**End-to-End Parity Error Count.** 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.

SWTOCTL - Switch Time-Out Control (0x750)

Bit Field	Field Name	Type	Default Value	Description
0	ETO	RW	0x1 Sticky	**Enable Switch Time-outs.** When this bit is set, switch time-outs for this port are enabled. In this mode, a TLP will be discarded if it has been in this port's input queue for more than the specified switch core time-out limit. See section Switch Time-Outs on page 3-16.
31:1	Reserved	RO	0x0	Reserved field.

SWTOSTS - Switch Time-Out Status (0x754)

Bit Field	Field Name	Type	Default Value	Description
0	PTLPTO	RW1C	0x0	**Posted TLP Time-Out.** This bit is set when a TLP is discarded from the port's IFB posted queue because of a time-out.
1	NPTLPTO	RW1C	0x0	**Non-Posted TLP Time-Out.** This bit is set when a TLP is discarded from the port's IFB non-posted queue because of a time-out.
2	CPTLPTO	RW1C	0x0	**Completion TLP Time-Out.** This bit is set when a TLP is discarded from the port's IFB completion queue because of a time-out.
3	ITLPTO	RW1C	0x0	**Inserted TLP Time-Out.** This bit is set when a TLP is discarded from the port's IFB insertion queue because of a time-out.
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	**Posted TLP Time-Out Reporting.** 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	**Non-Posted TLP Time-Out Reporting.** 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	**Completion TLP Time-Out Reporting.** 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	**Inserted TLP Time-Out Reporting.** 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	**Posted TLP Time-Out Count.** This field is incremented each time a TLP is discarded from the port's IFB posted queue because of a time-out. This counter saturates at its maximum value. Reading this field causes it to be cleared.
15:8	NPTLPTOC	RCW	0x0 Sticky	**Non-Posted TLP Time-Out Count.** This field is incremented each time a TLP is discarded from the port's IFB non-posted queue because of a time-out. This counter saturates at its maximum value. Reading this field causes it to be cleared.
23:16	CTLPTOC	RCW	0x0 Sticky	**Completion TLP Time-Out Count.** This field is incremented each time a TLP is discarded from the port's IFB completion queue because of a time-out. This counter saturates at its maximum value. Reading this field causes it to be cleared.
31:24	ITLPTOC	RCW	0x0 Sticky	**Inserted TLP Time-Out Count.** This field is incremented each time a TLP is discarded from the port's IFB insertion queue because of a time-out. This counter saturates at its maximum value. Reading this field causes it to be cleared.

SWTOTSCTL - Switch Time-Out Time-Stamp Control (0x760)

Bit Field	Field Name	Type	Default Value	Description
21:0	INC	RW	0x1 Sticky	**Time Stamp Increment.** This field contains the amount by which the time-stamp counter is incremented each cycle. Increasing the value in this field proportionally decreases the switch core time-out value. The intended use of this field is to accelerate summations.
30:22	TCOUNT	RW	0x1DD Sticky	**Terminal Count.** This field contains the value associated with bits 24 to 32 which signify a terminal count. When the time-stamp counter is greater than or equal to this value, then time-stamp epoch values contained in bits 33 and 34 of the time-stamp are incremented.The default value of 0x1DD corresponds to an epoch interval of ~32 S. The switch core time-out value is equal to two times the epoch interval. Thus, the default value corresponds to a switch core time-out value of 64 S.
31	Reserved	RO	0x0	Reserved field.

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 PES32T8: AC-coupled and DC-coupled (also called AC and DC pins). The Boundary Scan interface in the PES32T8 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 PES32T8 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 PES32T8.

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 PES32T8's many external signal pins. The JTAG TAP Controller can also be used for identifying the device part number. The JTAG logic of the PES32T8 is depicted in Figure 10.1.

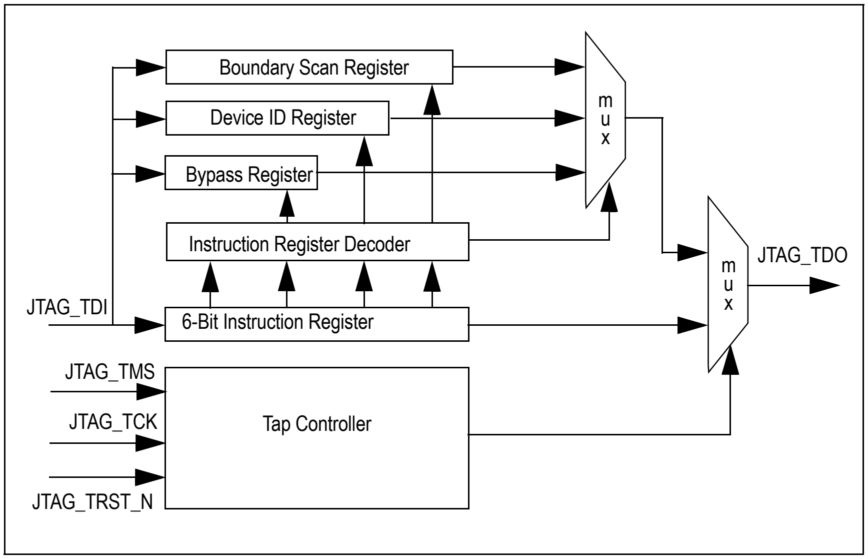


Figure 10.1 Diagram of the JTAG Logic

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.

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.

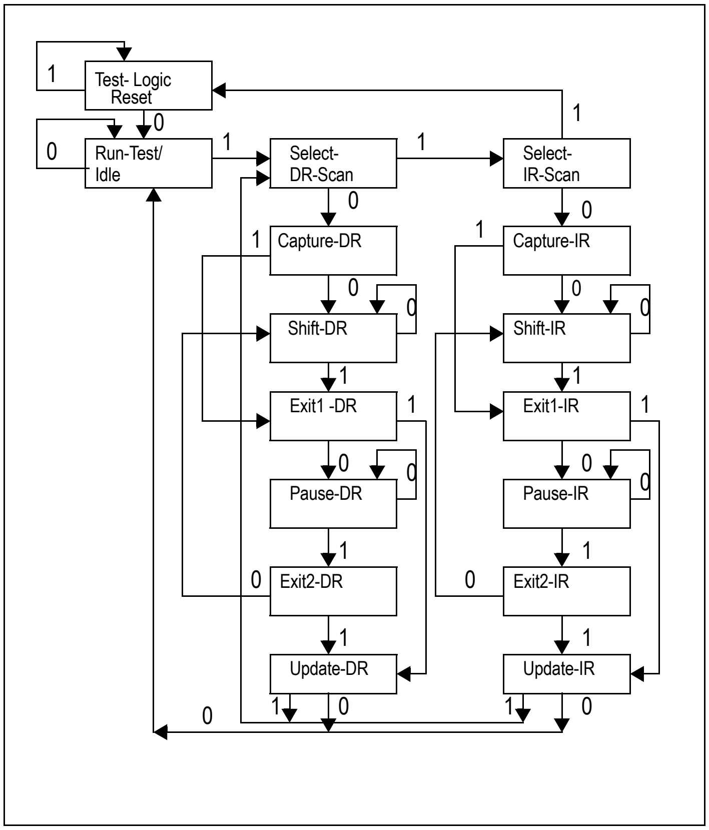


Figure 10.2 State Diagram of PES32T8's TAP Controller

Boundary Scan Chain

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

Table 10.2 Boundary Scan Chain (Part 1 of 2)

Notes

Function	Pin Name	Type[1]	Boundary Cell[2]
PCI Express Interface (cont.)	PEREFCLKN[2:1]	I	—
	PEREFCLKP[2:1]	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[15:0]	I/O	O/C
System Pins	CCLKDS	I	O
	CCLKUS	I	O
	MSMBSMODE	I	O
	P01MERGEN		O
	P23MERGEN		O
	P45MERGEN		O
	P67MERGEN		O
	PERSTN	I	O
	RSTHALT	I	O
	SWMODE[3:0]	I	—
EJTAG / JTAG	JTAG_TCK	I	—
	JTAG_TDI	I	—
	JTAG_TDO	O	—
	JTAG_TMS	I	—
	JTAG_TRST_N	I	—

Table 10.2 Boundary Scan Chain (Part 2 of 2)

[1] I = Input, O = Output

[2] 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.

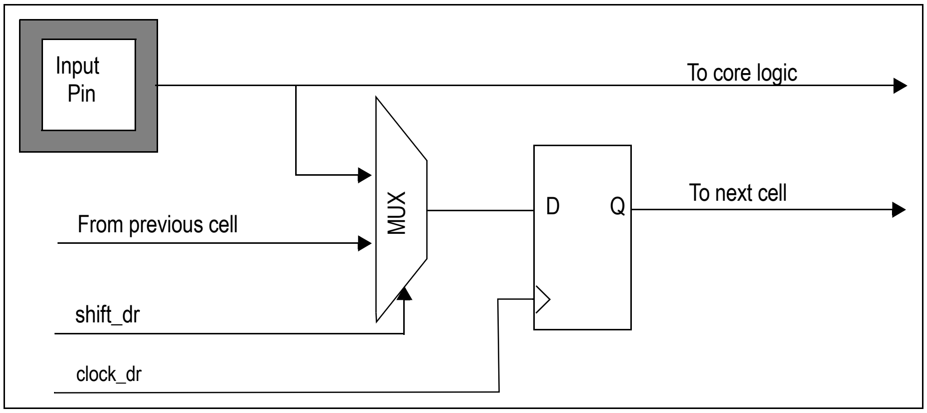


Figure 10.3 Diagram of Observe-only Input Cell

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

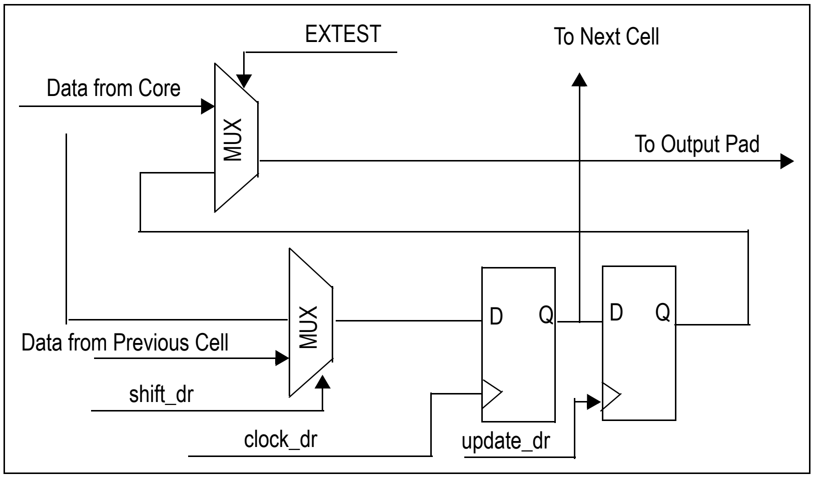


Figure 10.4 Diagram of Output Cell

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

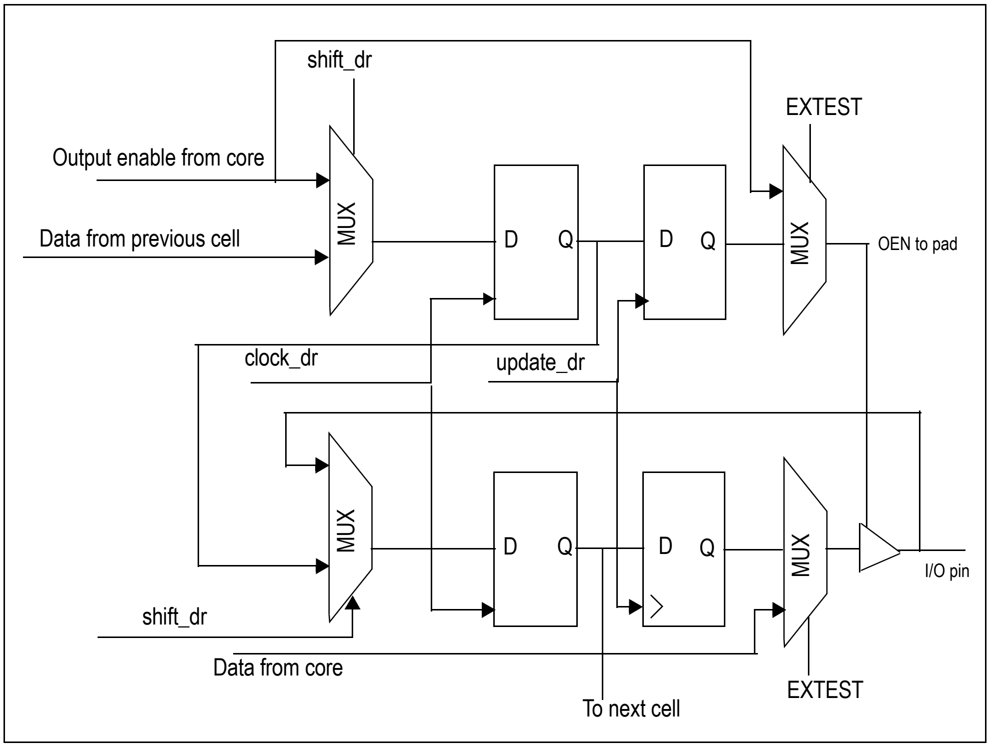


Figure 10.5 Diagram of Bidirectional Cell

The bidirectional cells are composed of only two boundary scan cells. They contain one output enable cell and one capture cell, which contains only one register. The input to this single register is selected via a mux that is selected by the output enable cell when EXTEST is disabled. When the Output Enable Cell is driving a high out to the pad (which enables the pad for output) and EXTEST is disabled, the Capture Cell will be configured to capture output data from the core to the pad.

However, in the case where the Output Enable Cell is low (signifying a tri-state condition at the pad) or EXTEST is enabled, the Capture Cell will capture input data from the pad to the core. The configuration is shown graphically in Figure 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 regis*ter 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 PES32T8'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 PES32T8. 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.

Therefore, instead of having to shift many times to get a value through the PES32T8, 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	**Manufacturer Identity** (11 bits) This field identifies the manufacturer as IDT.	R	0x33
27:12	Part_number	**Part Number** (16 bits) This field identifies the silicon as PES32T8.	R	0x802F
31:28	Version	**Version** (4 bits) This field identifies the silicon revision of the PES32T8.	R	silicon-dependent

Table 10.4 System Controller Device Identification Register

Version	Part Number	Mnfg. ID	LSB
xxxx	1000|0000|0010|1111	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.

Notes

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

Trademarks

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

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/.