# KeyStone Architecture
# Peripheral Component Interconnect Express (PCIe)

# User Guide

TEXAS INSTRUMENTS

# Release History

| Release | Date | Description/Comments |
|---|---|---|
| D | September 2013 | Added "Byte Strobe Requirements" section (Page 2-25) |
| | | Corrected "Endian Mode" column in "Big Endian Byte Swap for Outbound Transactions" table (Page 2-26) |
| | | Updated description in "POSTED_WR_EN" field of CMD_STATUS register (Page 3-6) |
| | | Added section of Power Domain and Module State Transitions Considerations (Page 2-34) |
| | | Updated description of ACK_FREQ field in ACK_FREQ register. (Page 3-124) |
| | | Updated description of MSI_IRQ register (Page 3-13) |
| | | Updated PCIe local configuration registers offset to take account of 0x1000 address space offset (Page 3-62) |
| | | Updated the description of REPLAY_TIMER bif field in SYS_NUM register (Page 3-127) |
| | | Added one note in Inbound Translation section for 64-bit addressing usage in RC mode (Page 2-13) |
| | | Added PHY loopback configuration steps for KeyStone I devices (Page 2-32) |
| | | Corrected RX_LOS field description in SERDES_CFG0/1 registers: set 4h to be Enabled instead of 1h (Page 3-59) |
| C | November 2012 | Added note for the SerDes related registers that are available only in KeyStone I devices (Page 3-4) |
| | | Added SerDes configuration for KeyStone II devices section (Page 2-9) |
| | | Updated FUNC field description in PID register (Page 3-5) |
| | | Updated the description of BAR mask register usage (Page 2-16) |
| | | Updated the description of BAR usage in RC mode (Page 2-16) |
| | | Corrected the byte swapping in mode D and mode C in big endian byte swap table for outbound transaction (Page 2-26) |
| | | Updated field descriptions of SERDES_CFG0 and SERDES_CFG1 registers. (Page 3-58) |
| B | March 2012 | Corrected the description of CFG_TX_SWING (bit[18]) in PL_GEN2 register to be 0=Full Swing; 1=Low Swing (Page 3-133) |
| | | Updated sticky field in the Registers section (Page 3-1) |
| | | Corrected that the "soft reset resets the entire PCIESS except the sticky bits in MMRs" instead of "except the MMRs" (Page 2-34) |
| | | Added one ECRC usage note in ECRC Generation and Checking section (Page 2-48) |
| | | Added the common clock descriptions to the clock control section (Page 2-4) |
| | | Deleted the statements of 100 MHz and 250 MHz in the clock section (Page 1-5) |
| | | Corrected MSIn_IRQ_STATUS description from "write with a 0 to clear" to "write with 1 to clear" (Page 3-17) |
| | | Deleted "250 MHz functional clock frequency operation (PIPE clock frequency)" from Features section to avoid confusion (Page 1-2) |
| | | Added BAR Mask Registers to Configuration Type 1 Registers section (Page 3-79) |
| | | Added BAR Mask Registers to Configuration Type 0 Registers section (Page 3-65) |
| | | Modified master port and slave port access descriptions (Page 1-5) |
| | | Added pcieAddr definition to EDMA transfer examples (Page 2-23) |
| | | Modified buffer offset computation in EDMA transfer examples (Page 2-23) |
| | | Added descriptions and examples of BAR Mask registers (Page 2-16) |
| | | Modified MST_PRIV field description in PRIORITY register (Page 3-12) |
| A | December 2010 | Added SERDES Configuration section (Page 2-5) |
| | | Modified the Clock Control description (Page 2-4) |
| | | Modified the Initialization Sequence for EP mode (Page 2-28) |
| | | Modified the Initialization Sequence for RC mode (Page 2-27) |
| | | Modified the Reset Consideration description (Page 2-34) |
| SPRUGS6 | November 2010 | Initial Release |

# Contents

## Preface                       ø-xxi

**Chapter 1**

## Introduction                         1-1

**Chapter 2**

## Architecture                        2-1

**Chapter 3**

## *Registers*  3-1

**Appendix A**

### *Encoding of LTSSM State in DEBUG Registers* <span style="float:right">A-1</span>

# List of Tables

## List of Figures

## List of Examples

# Preface

## About This Manual

This User Guide describes the features, architecture, and details of the Peripheral Component Interconnect Express (PCIe).

## Notational Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in `screen font`.
- Information you must enter is in **`boldface screen font`**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

**Note**—Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

**CAUTION**—Indicates the possibility of service interruption if precautions are not taken.

**WARNING**—Indicates the possibility of damage to equipment if precautions are not taken.

## Related Documentation from Texas Instruments

| | |
|---|---|
| *C66x CorePac User Guide* | SPRUGW0 |
| *TMS320C6000 DSP CPU and Instruction Set Reference Guide* | SPRU189 |
| *TMS320C6000 Programmer's Guide* | SPRU198 |
| *TMS320C6000 Code Composer Studio Tutorial* | SPRU301 |
| *TMS320C6000 DSP Peripherals Overview Reference Guide* | SPRU190 |

## Trademarks

TMS320C66x and C66x, VLYNQ, and Packet Accelerator are trademarks of Texas Instruments Incorporated.

All other brand names and trademarks mentioned in this document are the property of Texas Instruments Incorporated or their respective owners, as applicable.

**Chapter 1**

# Introduction

This document describes the peripheral component interconnect express (PCI Express) module.

## 1.1 Purpose of the Peripheral

The PCI Express (PCIe) module is a multi-lane I/O interconnect providing low pin count, high reliability, and high-speed data transfer at rates of up to 5.0 Gbps per lane per direction, for serial links on backplanes and printed wiring boards. It is a 3rd Generation I/O Interconnect technology succeeding ISA and PCI bus that is designed to be used as a general-purpose serial I/O interconnect in multiple market segments, including desktop, mobile, server, storage and embedded communications.

## 1.2 Terminology Used in This Document

The following is a brief explanation of some terms used in this document.

| Term | Definition |
|---|---|
| ASPM | Active state power management |
| CRC | Cyclic redundancy check |
| DBI | Direct bus interface |
| DLLP | Data link layer packet |
| EP | End point |
| MMR | Memory mapped register |
| PCI | Peripheral component interconnect |
| PCIe | PCI Express |
| PCIESS | PCI Express subsystem |
| PCISIG | PCI special interest group |
| PIPE | Physical interface for PCI Express |
| PLL | Phase-locked loop |
| PME | Power management event |
| POR | Power of reset |
| RC | Root Complex |
| SerDes | Serializer/Deserializer |
| TC | Traffic class |
| TLP | Transaction layer packet |
| VBUSM | Virtual Bus Multi-issue |
| VC | Virtual channel |
| VPD | Vital product data |

## 1.3 Features

The PCI Express module supports the following features:

- Dual operation mode: Root Complex (RC) and End Point (EP)
- Supports a single bidirectional link interface (i.e., a single ingress port and a single egress port) with a maximum of two lanes width (×2)
- Operated at a raw speed of 2.5 Gbps or 5.0 Gbps per lane per direction
- Maximum outbound payload size of 128 bytes
- Maximum inbound payload size of 256 bytes
- Maximum remote read request size of 256 bytes
- Ultra-low transmit and receive latency
- Support for dynamic-width conversion
- Automatic lane reversal

- Polarity inversion on receive
- Single virtual channel (VC)
- Single traffic class (TC)
- Single function in End Point (EP) mode
- Automatic credit management
- ECRC generation and checking
- PCI device power management with the exception of D3 cold with Vaux
- PCI Express active state power management (ASPM) state L0s and L1
- PCI Express link power management states, except L2 state
- PCI Express advanced error reporting
- PCI Express messages for both transmit and receive
- Filtering for posted, non-posted, and completion traffic
- Configurable BAR filtering, I/O filtering, configuration filtering, and completion lookup/timeout
- Access to configuration space registers and external application memory-mapped registers through BAR0 and through configuration access
- Legacy interrupts reception (RC) and generation (EP)
- MSI generation and reception
- PHY loopback in RC mode

The PCI Express module does not support the following features:
- No support for using ×2 link as two ×1 links
- No support for multiple VCs
- No support for multiple TCs
- No support for function-level reset
- No support for PCI Express beacon for in-band wake
- No built-in hardware support for hot plug.
- No support for vendor messaging
- No support for IO access in inbound direction in RC or EP mode
- No support for addressing modes other than incremental for burst transactions. Thus, the PCIe addresses cannot be in cacheable memory space
- No auxiliary power to maintain controller state to come out of D3 cold state
- No support for L2 link state

# 1.4 Functional Block Diagram

The block diagram of the PCI Express subsystem (PCIESS) is shown in Figure 1-1.

**Figure 1-1**　　**PCI Express Subsystem Block Diagram**
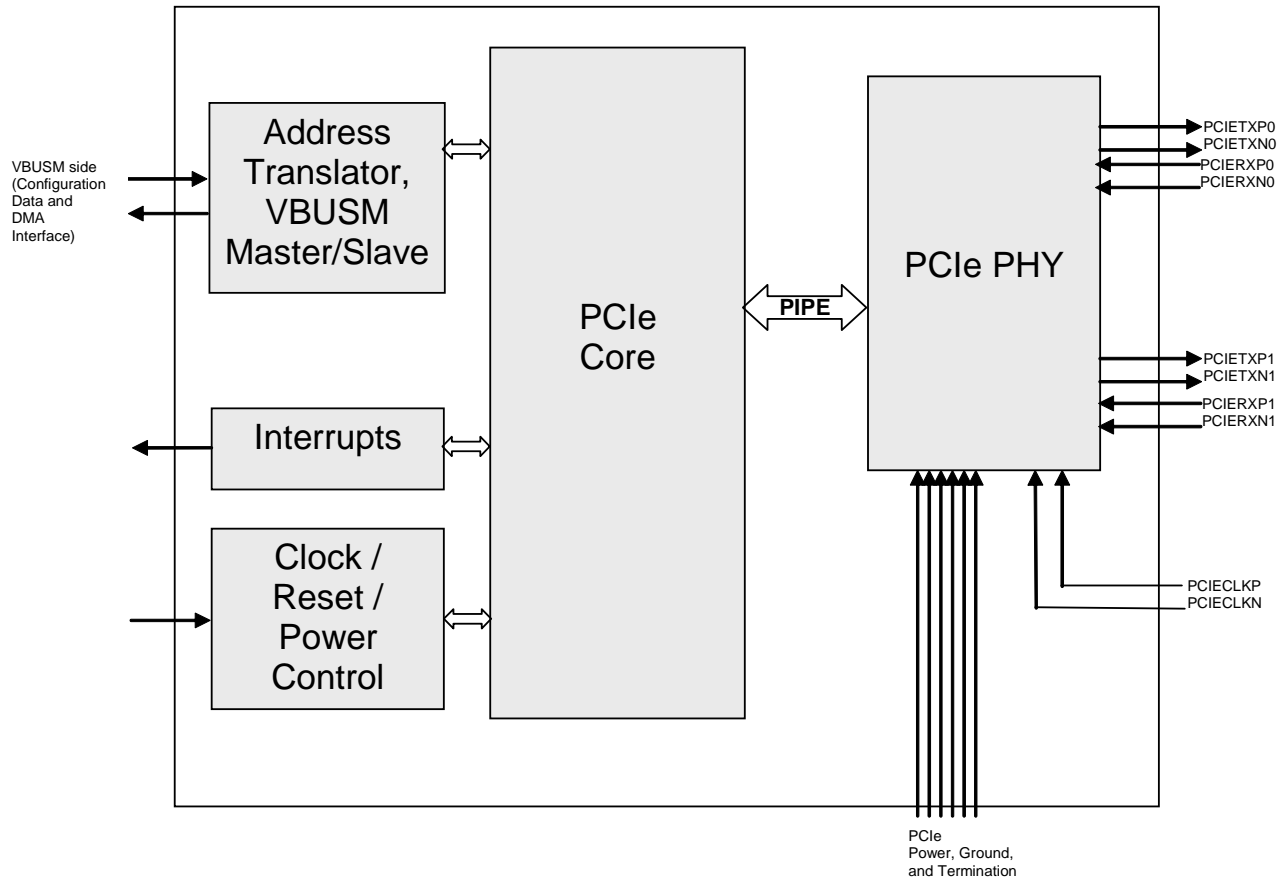


## 1.4.1 PCI Express Core Module

The PCI Express core contains the Transaction Layer, Data Link Layer, and MAC part of the PHY. The PCIe core is a Dual Mode core allowing it to operate as RC or EP. As an end point, it can operate as a legacy end point or native PCIe end point. There are two bootstrap pins PCIESSMODE[1:0] used to decide the default mode at power up (00: EP, 01: Legacy EP, 10: RC). Software can overwrite this by writing to the PCIESSMODE bits in the DEVSTAT register (see the device-specific data manual for details). PCIESSEN is another bootstrap value to determine whether the power domain of PCIe should be ON or not after the boot.

## 1.4.2 PCI Express PHY Interface

The PCIe PHY (SerDes) contains the analog portion of the PHY, which is the transmission line channel that is used to transmit and receive data. It contains a phase locked loop, analog transceiver, phase interpolator-based clock/data recovery, parallel-to-serial converter, serial-to-parallel converter, scrambler, configuration, and test logic.

### 1.4.3  VBUSM (Configuration and DMA Access Interface)

The PCIe has one 128-bit VBUSM master port and one 128-bit VBUSM slave port connected to the CPU side. The master port is for inbound transfer requests and the slave port is for both PCIe MMR access and outbound data transfer.

### 1.4.4  Clock, Reset, Power Control Logic

Several clock domains exist within the PCIESS. These clocks are functional clocks used by the PCIe controller and interface bridges as well as receive and transmit clocks used to clock data in and out respectively. The clocks required for clocking data and PHY functional clocks are generated by the PHY through the supplied input differential clock.

The PCIESS supports the conventional reset mechanism that is specified within the PCI Express Specification. The reset shown on the block diagram pertains to a hardware reset (cold or warm reset).

In addition to automatic power down mode exercised and entered by the hardware (Active State Power Management) when no activity is present, the PCIESS supports higher level power down mode that is controlled by user software.

### 1.4.5  Interrupts

The PCIESS is capable of generating up to 14 interrupts (INTA/B/C/D-legacy interrupts combined, MSI, error, power management, and reset) connected to the Interrupt Controller. The user software is required to acknowledge the serviced interrupt by writing to the corresponding vector onto the EOI register.

### 1.4.6  PCIe Power/Ground/Termination

Several power supplies, grounds, and termination exist to properly power up the PHY.

### 1.4.7  Differential Data Lines

A pair of differential data lines exists, for both transmit and receive paths, for each lane.

## 1.5  Supported Use Case Statement

The PCIe subsystem has only one interface link and this link can be used in ×1 or ×2 lane arrangements connecting to only one device. In other words, there exists no support for connecting to two devices in an ×1 arrangement because it has the support of a single interface link. This also means that it can not be used as a switch.

## 1.6  Industry Standard(s) Compliance Statement

The PCI Express module complies with the following standards:
- PCI Express base specification (revision 2.0)
- Synopsys DWC PCIe Dual Core (version 3.51a)
- TI SerDes 1.1.03

# Architecture

This chapter describes the details of the architecture of the PCI Express module and how it is structured.

## 2.1 Protocol Description(s)

### 2.1.1 PCI Express Topology

The PCI Express fabric looks like a tree structure with nodes connected to each other via point-to-point links. The root node is called the *root complex* (RC), the leaf nodes are called *end points* (EP) and the nodes that connect multiple devices to each other are called *switches* (SW). The RC can have multiple downstream ports but that still requires multiple instances of the PCI Express protocol stack for each RC port. Having one RC port in the PCI Express subsystem (PCIESS) does not imply that more than one EP can be connected to the PCIESS without adding a PCI Express switch. In addition, a RC port with a single port of multiple lanes cannot be used as two ports with the lanes split between them.

**Figure 2-1  PCI Express Example Topology**
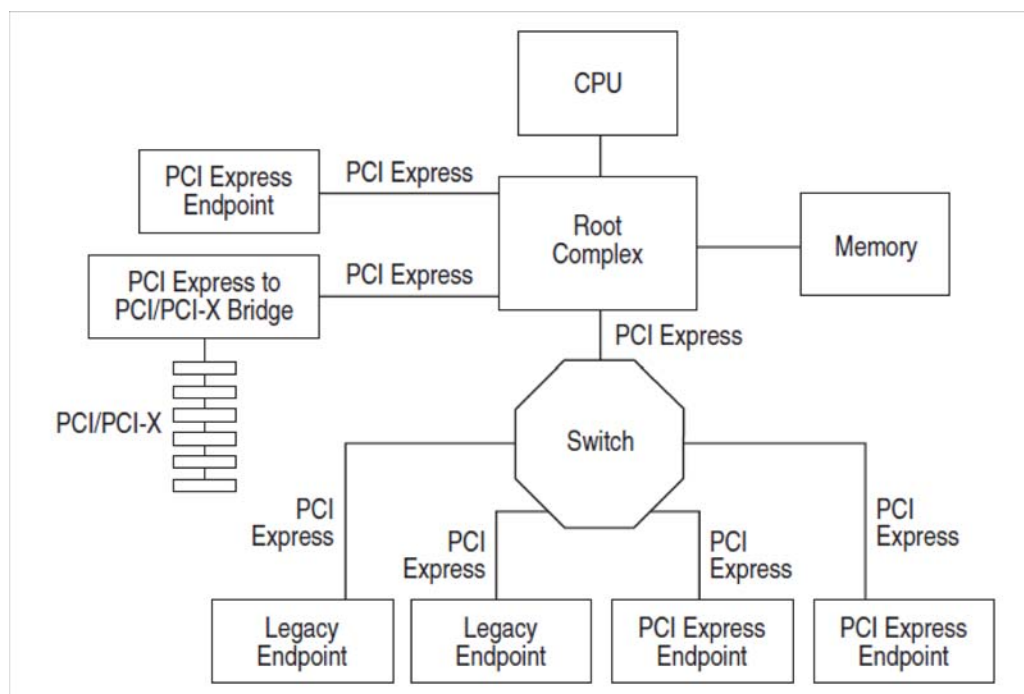


### 2.1.2 Serial Link

PCI Express is a point-to-point serial signaling protocol. Each link consists of one TX and one RX differential pair. On each link, depending upon the protocol version, 2.5 Gbps or 5.0 Gbps can be transported in each direction. Accounting for the 8b/10b encoding used over the serial link, the data rates translate to 2.0 and 4.0 Gbps of throughput at higher layers of PCI Express protocol. Data is transferred in packets, which include an address and a variable size data payload.

### 2.1.3  Supported PCI Express Transactions

All of the PCI Express transactions defined, Posted and Non-posted, are supported except the Locked Memory Read request transaction and its subsequent completion Locked response transaction. Inbound I/O read/write transactions also are not supported.

**Table 2-1      Supported PCI Express Transactions**

| Transaction Packet Types | Posted/Non-posted |
|---|---|
| Memory read | Non-posted |
| Memory write | Posted |
| IO read | Non-posted |
| IO write | Non-posted |
| Configuration read (type 0 and type 1) | Non-posted |
| Configuration write (type 0 and type 1) | Non-posted |
| Message request without data | Posted |
| Message request with data | Posted |
| Completion without data | - |
| Completion with data | - |
| **End of Table 2-1** | |

The non-posted transactions comprise of a transaction layer packet (TLP) from the requester to completer. The completer, at a later time, sends a completion TLP to the requester. The completion TLP is used to inform the requester that the completer has received the request. In addition, the completion TLP also contains the data if the transaction was a read transaction. Non-posted write transactions contain the data in request TLP.

For posted transactions, the request TLP is sent but there is no response TLP sent from the completer to the requester.

## 2.2 Clock Control

The PCIESS uses multiple clock domains. At the top level, there are only two functional clocks of relevance - a reference clock to the PCIe PHY and a clock for VBUSM interfaces of the subsystem.

The VBUSM interface functional clock is generated from the main PLL internally. There is no configuration of this clock needed by the user. Please see the device-specific data manual for the details.

The other clock input to the PCIESS is the differential clock. This differential clock is used by the PHY PLL to generate the necessary functional and bit clock required by the PHY. It is possible to use several different frequency values for the PHY reference clock.

To comply with the PCIe specifications, it is acceptable that a reference clock of 100 MHz that meets the requirements for REFCLK as described in the PCI Express Card Electromechanical Specification be driven as a differential signal into PCIe reference clock pads of the device. Greater jitter tolerance can be achieved by using one of the higher frequency reference clock values specified in the data manual.

Many PCIe connections, especially backplane connections, require a synchronous reference clock between the two link partners. To achieve this a common clock source, referred to as REFCLK in the PCI Express Card Electromechanical Specification, should be used by both ends of the PCIe link. If Spread Spectrum Clocking (SSC) is used it is required that a common reference clock be used by the link partners. Most commercially available platforms with PCIe backplanes use Spread Spectrum Clocking to reduce EMI.

If common clock architecture is not used, then the software driver must setup appropriate bit (COMMON_CLK_CFG bit in LINK_STAT_CTRL register) in the PCI Configuration registers to indicate so to the system. The number of training sequences is significantly increased if reference clock is not shared between devices on a PCIe link.

The PCIESS operation is completely dependent upon availability of clock from the PLL that is inside the PHY. All registers in the PCIESS are located in the clock domain that is dependent upon PLL to be in lock and properly configured. So, transactions initiated before ensuring that the PLL is locked, the subsystem will not operate. The device level registers are used to enable PLL and verify lock status. Please see the SerDes Configuration section for details.

## 2.3 SerDes Configuration for KeyStone I Devices

The physical layer SerDes has a built-in PLL, which is used for the clock recovery circuitry. The PLL is responsible for clock multiplication of a slow speed reference clock (refclkp/n). And the reference clock is the input to PCIe clock pins PCIECLKN and PCIECLKP. This reference clock has no timing relationship to the serial data and is asynchronous to any CPU system clock. The multiplied high-speed clock is routed only within the SerDes block. It is not distributed to the remaining blocks of the peripheral, nor is it a boundary signal to the core of the device. It is extremely important to have a good quality reference clock, and to isolate it and the PLL from all noise sources.

### 2.3.1 SerDes Configuration Registers

#### 2.3.1.1 PCIe SerDes Configuration Register (PCIE_SERDES_CFGPLL)

The SerDes macro is configured with the chip-level register PCIE_SERDES_CFGPLL. The memory-map address of this register in KeyStone devices is 0x02620358 and the default value is 0x000001C9.

**Table 2-2 PCIe SerDes Configuration Register (PCIE_SERDES_CFGPLL) Field Descriptions (Part 1 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-16 | Reserved | Reads return 0 and writes have no effect. |
| 15 | STD | Standard selection. Set to enable S-ATA operation. |
| 14-13 | CLKBYP | Clock bypass. Facilitates bypassing of the PLL with either refclkp/n or testclkt, and bypassing of the recovered receiver clock with testclkr. |
| 12-11 | LB | Loop bandwidth. Specify loop bandwidth settings. Jitter on the reference clock will degrade both the transmit eye and receiver jitter tolerance, thereby impairing system performance. Performance of the integrated PLL is optimized according to the jitter characteristics of the reference clock via the LB filed.<br>0 = Medium bandwidth. This setting is suitable for most systems that input the reference clock via the low jitter clock input buffer (LJCB), and is required for standards compliance (unless using an LC-based cleaner PLL).<br>1h = Ultra high bandwidth.<br>2h = Low bandwidth. In systems where the reference clock is directly input via a low jitter input cell, but is of lower quality, this setting may offer better performance. It will reduce the amount of reference clock jitter transferred through the PLL. However, it also increases the susceptibility to loop noise generated within the PLL itself. It is difficult to predict whether the improvement in the former will more than offset the degradation in the latter.<br>3h = High bandwidth. This is the setting appropriate for systems where the reference clock is cleaned through the ultra low jitter LC-based cleaner PLL. Standards compliance will be achieved even if the reference clock input to the cleaner PLL is outside the specification for the standard. |
| 10 | SLEEPPLL | Sleep PLL. Puts the PLL into the sleep state when high. |
| 9 | VRANGE | VCO range. Select between high and low range VCO. |
| 8 | ENDIVCLK | Enable DIVCLK output. Set to enable output of a divide by-5 of PLL clock. |

**Table 2-2      PCIe SerDes Configuration Register (PCIE_SERDES_CFGPLL) Field Descriptions  (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 7-1 | MPY | PLL multiply. Select PLL multiply factors between 4 and 25. <br>   10h = 4x <br>   14h = 5x <br>   18h = 6x <br>   20h = 8x <br>   21h = 8.25x <br>   28h = 10x <br>   30h = 12x <br>   32h = 12.5x <br>   3Ch = 15x <br>   40h = 16x <br>   42h = 16.5x <br>   50h = 20x <br>   58h = 22x <br>   64h = 25x <br>   Others = Reserved |
| 0 | ENPLL | Enable PLL. <br>   0 = Disable PLL. <br>   1 = Enable PLL. |
| **End of Table 2-2** | | |

### 2.3.1.2  PCIe SerDes Status Register (PCIE_SERDES_STS)

The status bits of PLL, transmitter and receiver port of the SerDes interface are mapped to the read-only chip-level PCIE_SERDES_STS register. It has both of the status bits for Lane0 and Lane1. The memory-map address of this register in KeyStone devices is 0x0262015C and the default value is 0x00000001.

**Figure 2-2      PCIe SerDes Status Register (PCIE_SERDES_STS)**

| 31 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Reserved | | RDTCTIP 1 | TX_TEST FAIL1 | OCIP 1 | LOSDTCT 1 | SYNC 1 | RX_TEST FAIL1 | RDTCTIP 0 | TX_TEST FAIL0 | OCIP 0 | LOSDTCT 0 | SYNC 0 | RX_TEST FAIL0 | LOCK |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-1 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 2-3      PCIe SerDes Status Register (PCIE_SERDES_STS) Field Descriptions  (Part 1 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-13 | Reserved | Reads return 0 and writes have no effect. |
| 12 | RDTCTIP1 | Receiver detect in progress of Lane1. Driven high when PCIe receiver detection starts, and deasserted when receiver detection disabled. |
| 11 | TX_TESTFAIL1 | Test failure. Driven high when an error is encountered during a test sequence executed on transmitter port of Lane1. |
| 10 | OCIP1 | Offset compensation in progress of Lane1. Driven high asynchronously during offset compensation. |
| 9 | LOSDTCT1 | Loss of Signal detect of Lane1. Driven high asynchronously when a loss of signal (electrical idle) condition is detected. |
| 8 | SYNC1 | Symbol alignment of Lane1. When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed. |
| 7 | RX_TESTFAIL1 | Test failure. Driven high when an error is encountered during a test sequence executed on receiver port of Lane1. |
| 6 | RDTCTIP0 | Receiver detect in progress of Lane0. Driven high when PCIe receiver detection starts, and deasserted when receiver detection disabled. |

**Table 2-3    PCIe SerDes Status Register (PCIE_SERDES_STS) Field Descriptions  (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 5 | TX_TESTFAIL0 | Test failure. Driven high when an error is encountered during a test sequence executed on transmitter port of Lane0. |
| 4 | OCIP0 | Offset compensation in progress of Lane0. Driven high asynchronously during offset compensation. |
| 3 | LOSDTCT0 | Loss of Signal detect of Lane0. Driven high asynchronously when a loss of signal (electrical idle) condition is detected. |
| 2 | SYNC0 | Symbol alignment of Lane0. When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed. |
| 1 | RX_TESTFAIL0 | Test failure. Driven high when an error is encountered during a test sequence executed on receive port of Lane0. |
| 0 | LOCK | PLL lock. Driven high asynchronously between 2048-3071 refclkp/n cycles after the PLL has locked, which will occur within 200 refclkp/n cycles. While LOCK is low, the PLL output frequency may overshoot by no more than <5%, provided MPY is not changed to select a lower multiplication factor. The same percentage overshoot will be mirrored by the bus clocks originating from the SerDes. |
| **End of Table 2-3** | | |

> **Note—**The PCIe SerDes configuration / status registers are located in the chip configuration section of the device. Before writing to these registers, the kicker mechanism must be used to unlock these registers. For more information about using the kicker mechanism, please see the device-specific data manual.

## 2.3.2  Enabling the PLL

To enable the internal PLL, the ENPLL bit of PCIE_SERDES_CFGPLL must be set. After setting this bit, it is necessary to allow 1 μs for the regulator to stabilize. Thereafter, the PLL will take no longer than 200 reference clock cycles to lock to the required frequency, provided refclkp/n are stable.

The LOCK bit of PCIE_SERDES_STS register will be driven high by the digital lock detector. We can sample this bit during the PLL initialization to indicate the refclkp/n are stable.

### 2.3.3 Reference Clock Multiplication

The line rate versus PLL output clock frequency can be calculated. This is summarized in below table:

**Table 2-4        Line Rate vs. PLL Output Clock Frequency**

| Rate | Line Rate | PLL Output Frequency | RATESCALE |
|---|---|---|---|
| Full | x Gbps | 0.5x GHz | 0.5 |
| Half | x Gbps | x GHz | 1 |
| Quarter | x Gbps | 2x GHz | 2 |
| $refclkp/n \ = \ \dfrac{LINERATE \times RATESCALE}{MPY}$ | | | |
| **End of Table 2-4** | | | |

The PLL multiplier (MPY) should be configured based on refclkp/n and targeted PLL output frequency (line rate). The RATESCALE is set as 1 by default in the device level.

For example, the typical refclkp/n is 100 MHz, the targeted output frequency of the PLL is 2.5 GHz (in both Gen1 and Gen2 modes), and the RATESCALE is 1, then the PLL multiplier should be configured as 25×.

**Note—**The targeted PLL output frequency (line rate) is 2.5 GHz in both PCIe Gen1 and Gen2 modes. The PCIe PHY performs data bus width conversion from 8-bit to 16-bit when the module switches from Gen1 speed to Gen2 speed. The output frequency of the PLL stays constant irrespective of whether it is operating in Gen1 or Gen2 mode. Set the DIR_SPD bit to 1 in the PL_GEN2 register during the initialization can switch the PCIe link speed mode from Gen1 (2.5Gbps) to Gen2 (5.0Gbps).

## 2.4  SerDes Configuration for KeyStone II Devices

SerDes module information for KeyStone II devices is not provided in this user guide. Please check for availability of the SerDes User Guide for KeyStone II Devices on the device product page.

## 2.5  Signal Descriptions

The PCIESS pin names with signal direction and description are shown in Table 2-5.

**Table 2-5      PCI Express Signal Description**

| Pin Name | Type | Description |
|---|---|---|
| PCIECLKN | Input | PCIe Clock Input to Drive PCIe SerDes |
| PCIECLKP | Input | |
| PCIERXN0 | Input | PCIe Receive Data Lane 0 |
| PCIERXP0 | Input | |
| PCIERXN1 | Input | PCIe Receive Data Lane 1 |
| PCIERXP1 | Input | |
| PCIETXN0 | Output | PCIe Transmit Data Lane 0 |
| PCIETXP0 | Output | |
| PCIETXN1 | Output | PCIe Transmit Data Lane 1 |
| PCIETXP1 | Output | |
| **End of Table 2-5** | | |

## 2.6  Pin Multiplexing

The PCIESS functional pins are not pin multiplexed with other signals. Only the module configuration bits (PCIESSMODE[1:0] and PCIESSEN) are latched at power up by GPIO and Timer pins. Please see the device-specific data manual for details of how pin multiplexing affects the PCIe module.

## 2.7  Address Translation

PCI Express TLP transactions use PCIe addresses. There is a mapping requirement between a PCIe address and a local internal bus address and to accommodate this address mapping, built-in hardware address translation mechanisms exist. That is, if the *Type* field of an outgoing or received TLP indicates the use of address routing, then an outbound or inbound address translation is required and is performed accordingly to map internal bus address to PCIe address or vice-versa using hardware address translators. Address translations for outbound and inbound transactions are discussed below.

PCIe recognizes four address spaces, Memory, I/O, Configuration, and Message. Messages do not consume any Memory or I/O resource. I/O addressing is used by PCIe that supports legacy device operation capabilities; this is not a mandatory feature by PCIe and when supported the I/O spaces is memory mapped to system memory. This implies that only two types of device address spaces, Configuration and Memory spaces, need to map between PCIe address and internal bus address.

The device address space is divided into two spaces (Range 0 and Range 1). Address space (Range 0) that is used for PCIe configuration task is referred to as Configuration space, and a second address space (Range 1) that is used for accessing memory (non-configuration related) is referred to as Memory space.

> **Note**—All I/O accesses use address routing. There is no support for I/O access in inbound direction. That is, when it comes to a transaction that performs I/O access only TLPs with outbound transactions are supported.

## 2.7.1 Outbound Address Translation

The PCIe subsystem allows mapping of internal bus address to/from PCIe address on outbound TLPs. This is accomplished by using outbound address translation logic. For each outbound read/write request, the address translation module within the PCIESS can convert an internal bus address to a PCIe address of memory read/write type.

The address translation logic uses information in address translation registers to perform the mapping. The registers, OB_SIZE, OB_OFFSET_INDEXn and OB_OFFSETn_HI are used in conjunction with outbound address translator.

**Figure 2-3    Outbound Address Translation**
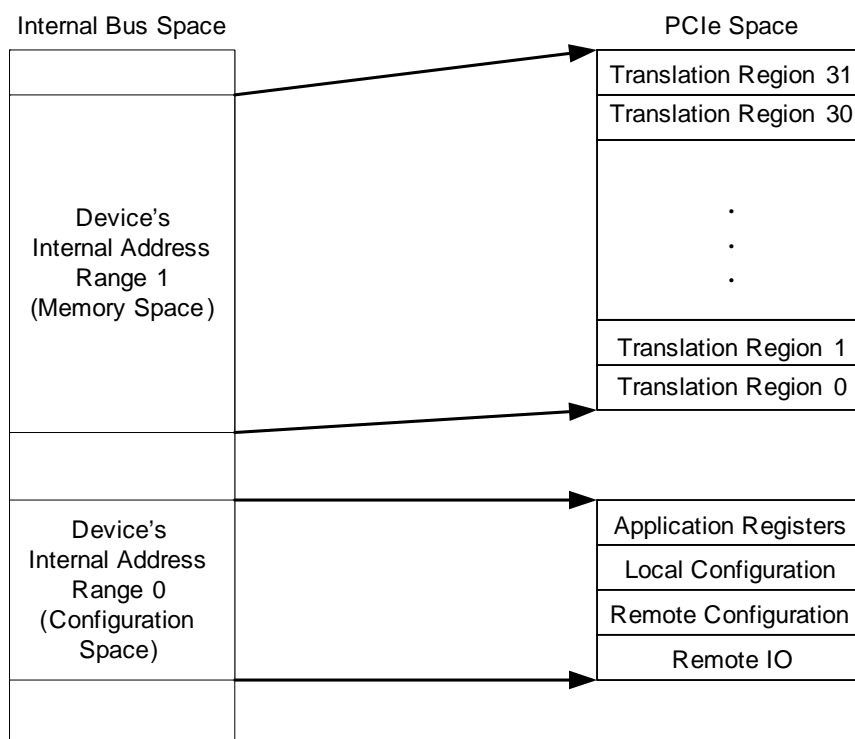


The memory range that the PCIESS occupies in the devices internal address range is divided into 32 equally-sized translation regions (Regions 0 to 31). These regions can be programmed to be of 1, 2, 4, or 8 MB in size via the OB_SIZE register. Each such region can be remapped to a PCIe address range of same size as the size of translation region itself. The address translation logic identifies and extracts the 5 bits (32 regions) of the device's internal address and determines which of the 32 regions it belongs to. The bit address positions of these 5 bits depend on the range size. Once the region is

identified, the address translation logic then generates PCIe base address, from the values provided within the corresponding configuration registers for that region, i.e., the registers OB_OFFSET_INDEXn [n=0-31] and OB_OFFSETn_HI [n=0-31]. If 32-bit addressing is used, OB_OFFSETn_HI will always be programmed with zero.

Once the PCIe base address has been identified, the offset that is to be added to this base address is derived from the lower bit fields of the internal bus address and the bit fields that make up this offset correspond to the size of the regions.

Application software is required to identify the desired internal bus memory that is to be accessible by the PCIe module and initialize the corresponding registers prior to enabling PCIe transactions.

To accomplish address translation, each region uses three registers content values:

1. **Outbound Size Register (OB_SIZE)** — Application software initializes this register with the size value. This field is used to identify the size of all equally spaced regions and identify one of the 32 regions based on the size value. OB_SIZE=0,1,2 and 3 correspond to region sizes of 1MB, 2MB, 4MB and 8MB and the corresponding indexed regions for these sizes are bits[24:20], bits [25:21], bits[26:22] and bits[27:23] respectively. So, index identification is done based on the size of regions and the values of these 5 bit fields identify one of the 32 Regions that are used for mapping. This affects the meaningful bit fields used within OB_OFFSET_INDEXn [n=0-31] register. The index value will be extracted from the device internal bus address and will be matched to one of the 32 regions. The lower bit fields corresponding to the size of the region are masked and not used in the mapping.

2. **Outbound Translation Region n Offset Low and Index Register (OB_OFFSET_INDEXn [n=0-31])** — Application software initializes this register with the 32-bit PCIe address. Not all bits in this field are used in all cases. The number of bits used changes depending upon the size of each address translation region. If each region is 1MB, 2MB, 4MB or 8MB, then this field will be used to create bits[31:20], bits[31:21], bits[31:22] or bits[31:23] of the translated address respectively.

3. **Outbound Address Offset High Register (OB_OFFSETn_HI [n=0-31])** — Application software initializes this register and this is a 32-bit field that represents bits [63:32] of translated PCIe space address if using 64-bit addressing. This register is required to be programmed with a Zero value if using 32-bit addressing.

The following example demonstrates the mapping of a given internal bus address to a PCIe address.

**Example 2-1    Outbound Address Translation**

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

For a given device internal bus address of 0x9D3A1234, what would be the corresponding PCIe address that would be used on an outgoing TLP header (assume a 2MB region partition)?

For this example, further assume the following:
- 64-bit addressing is to be used
- Translation Region 9 high offset OB_OFFSET9_HI = 0x33445566
- Translation Region 9 low offset OB_OFFSET_INDEX9 = 0x56E00001 (Region 9 is enabled)

For an internal address 0x9D3A1234, the translated address is calculated based as outlined below:

1. Extract the 5 bits index and offset from given internal bus address of 0x9D3A1234. Because the region partition is 2MB (OB_SIZE = 1), bits [25:21] of address 0x9D3A1234 is extracted as 01001b, which is 9.

2. Match it to one of the region index fields. In this example, it matches translation region 9 as internal address bits [25:21] are equal to value of index 9. So the translation registers contents for region 9 will be used.

3. Use programmed offsets 0x33445566 (from OB_OFFSET9_HI) for 64-bit addressing bits[63:32] and 0x56E (from OB_OFFSET_INDEX9) for bits[31:21]. The translated PCIe base address = 0x33445566_56E00000.

4. Map the offset (bits[20:0]) of the internal bus address directly to PCIe address. Bits[20:0] of 0x9D3A1234 is 0x1A1234.

5. Compute translated address = translated base address + offset = 0x33445566_56E00000+0x1A1234 = 0x33445566_56FA1234.

**End of Example 2-1**

- - - - - - - - - - - - - - - - - - - - - - - - - - - - -

In Example 2-1, 2MB size region will allow a total of 2MB*32=64MB unique internal bus address location to be mapped onto a corresponding PCIe address, values 0x0000_0000 to 0x03FF_FFFF. Any internal bus address outside this 64MB space will not be uniquely mapped and the internal bus address will be truncated to fall in within the 64MB region, which happens to be for this example. In other words, bits [31:26] of the given example internal bus address is masked and does not contribute to the PCIe translated address generation (e.g. internal bus address 0x9D3A_1234 will have the same PCIe translated address as 0x913A_1234 or 0x953A_1234, etc.).

> **Note—**The outbound translation is used only for memory read/write transactions and not for PCIe configuration or IO read/write transactions. Outbound translation may not be necessary in all applications.

### 2.7.1.1 Transactions Violating Address Translation Boundaries

If a transaction is large enough that it goes past the address translation region, unspecified behavior may occur. The address translation works only at the time a command is issued. So, a memory write, for example, will not automatically go to the next translation region if it starts in the previous one and is bigger than the remaining size in the starting translation region.

## 2.7.2 Inbound Address Translation

Inbound address translation is used to remap accepted incoming accesses from other PCIe devices to locations within the device's memory map.

The PCIESS is aware of two internal bus address spaces. The first address space (Address Space 0) is dedicated for local application registers, local configuration accesses, remote configuration accesses and remote IO accesses (RC only). The second one (Address Space 1) is dedicated for data transfer. Details on this are covered in Section 2.8.

Address Space 0 occupies a contiguous 16K bytes of location, where 4KB of 16KB is the configuration space. Address Space 1 is used for data transfer and is large in size and not necessarily contiguous. To perform a mapping of a PCIe addresses that would land within Address Space 1, 4 dedicated regions (Region 0~3) are to be used by the inbound address translator to achieve the desired address mapping. Note that outbound translation has 32 regions while inbound translation has 4 regions.

Any accepted incoming address should match one of the BARs (Base Address Registers). If the PCIESS is configured as EP, BAR0~BAR5 will be mapped to one of the two address spaces (Address Space 0/1). If the PCIESS is configured as RC, only BAR0 and BAR1 will be used to be mapped to the address spaces. Please see Section 2.7.3 for details of BAR usage.

In 32-bit addressing, BAR0 is dedicated to Address Space 0 and BAR1~BAR5 are dedicated to Address Space 1. Regions 0~3 should be associated only with BAR1~BAR5.

In 64-bit addressing, a pair of adjacent BARs concatenated is required to hold the 64-bit address. This means BAR0 and BAR1 will hold 64-bit address with BAR0 holding the lower 32-bit address to match while BAR1 holding the higher 32-bit address to match. And BARs[0-1] will be dedicated to Address Space 0. The same holds for Address Space 1 association. BARs[2-3] and BARs[4-5] will hold the accepted 64-bit address that will be associated to Address Space 1 where mapping takes place via the use of Regions[0-3].

Address translation for Address Space 0 does not exist because the internal location is unique and is contiguous. All is needed is that the PCIe address within the received TLP address matches BAR0 for 32-bit addressing, and BAR0, BAR1 for 64-bit addressing.

Please note that in 64-bit addressing, BAR0 and BAR1 are concatenated and dedicated to Address Space 0. There is no other BAR available in RC mode to map packets with 64-bit addressing to any other internal memory regions with inbound address translation.

However, address translation for Address Space 1 requires the use of one of the four regions (Regions[0-3]) to map accepted TLPs to internal memory. Four memory mapped registers that are region specific are used by the inbound address translator.

1. **Inbound Translation Bar Match Register (IB_BARn [n=0-3])** — This field indicates which BAR the inbound transaction must be targeted to for the translation rule specified to activate.

2. **Inbound Translation Start Address High Register (IB_STARTn_HI [n=0-3])** — This field indicates the starting address bits [63:32] as seen in PCIe address. Typically, this field will match the BAR value and is used as the reference for this address translation region in 64-bit addressing. This register is required to be programmed with a Zero value if using 32-bit addressing.

3. **Inbound Translation Start Address Low Register (IB_STARTn_LO [n=0-3])** — This field indicates the starting address bits [31:0] as seen in PCIe address. Typically, this field will match the BAR value and is used as the reference for this address translation region in both of 32-bit and 64-bit addressing.

4. **Inbound Translation Address Offset Register (IB_OFFSETn [n=0-3])** — This field indicates the internal bus address that will be the starting point of the mapped or translated PCIE address region.

The procedure used by the inbound address translator to perform the mapping between the PCIe address and internal bus address as follows:

1. Extract the offset: PCIe address - (IB_STARTn_HI : IB_STARTn_LO)
2. Compute internal address: add IB_OFFSETn to the offset extracted on step 1.

**Example 2-2     Inbound Address Translation**

For a given 64-bit PCIe address of 0x12345678_ABC50000, which qualifies for acceptance, what would be the corresponding internal bus address (assume that Region 1 registers are programmed to match BAR2, BAR3 programmed addresses)?

For this example, further assume that application software has programmed the set of registers corresponding to Region 1 as follows:

- IB_BAR1 = 2. This assignment associates Region 1 with BAR2, BAR3 for 64-bit addressing. The programmed value 2 here associate the match between Region 1 (IB_BAR1) and configuration register BAR2.
- IB_START1_HI = 0x12345678
- IB_START1_LO = 0xABC00000
- IB_OFFSET1 = 0x33400000

The internal bus address is computed by extracting the offset from the PCIe address (address within the TLP header) and add the resultant to the start address of the internal address.

- Extract the offset from the PCIe address:

  Extracted offset = PCIe address - (IB_STARTn_HI : IB_STARTn_LO)

  = 0x12345678_ABC50000 - 0x12345678_ABC00000

  = 0x00050000

- Compute internal device address:

  Internal address = IB_OFFSET + extracted offset

  = 0x33400000 + 0x00050000

  = 0x33450000

**End of Example 2-2**

In this example, PCIe address 0x12345678_ABC50000 is translated or mapped to internal bus address 0x33450000.

### 2.7.2.1  Mapping Multiple Non-Contiguous Memory Ranges To One Region

A single inbound address translation region can be used to map accesses to multiple non-contiguous internal address ranges, within Address Space 1, by ensuring that the start addresses of these regions are programmed in ascending order in the inbound start address registers.

**Example 2-3     Mapping Multiple Non-Contiguous Memory Ranges to One Region**

In this example, two BARs (BAR1 and BAR2) are remapped to four separate locations that are non-contiguous using 32-bit addressing. The first two regions (Region 0 and Region 1) remap accesses landing in BAR1 space and the following two regions (Region 2 and Region 3) remap accesses landing in BAR2 space. Each BAR is treated as a 32-bit BAR and therefore, the higher 32-bits of start address are zero.

Programmed value of Base Address Registers in configuration space:
- BAR1: 0x11110000
- BAR2: 0x22220000

**Table 2-6      Mapping Multiple Non-Contiguous Memory Ranges to One Region**

| IB Region | IB_BAR | IB_START_ADDR | IB_OFFSET | TLP Address | Translated Address |
|-----------|--------|---------------|-----------|-------------|--------------------|
| 0 | 1 | 0x11110000 | 0x33330000 | 0x11110080 | [0x11110080 - 0x11110000] + 0x33330000 = 0x33330080 |
| 1 | 1 | 0x11118000 | 0x44440000 | 0x11119000 | [0x11119000 - 0x11118000] + 0x44440000 = 0x44441000 |
| 2 | 2 | 0x22220000 | 0x55550000 | 0x22220400 | [0x22220400 - 0x22220000] + 0x55550000 = 0x55550400 |
| 3 | 2 | 0x22220800 | 0x66660000 | 0x22231000 | [0x22231000 - 0x22220800] + 0x66660000 = 0x66670800 |
| **End of Table 2-6** | | | | | |

**End of Example 2-3**

In case there are more than one matching BAR, then the highest start address that is less than the PCIe TLP address is the one that is considered for translation.

### 2.7.2.2  BAR0 Exception for In-Bound Address Translation

The memory space covered by BAR0 in inbound direction is completely dedicated to accessing the application registers, Address Space 0, in both RC and EP modes. It implies that the BAR0 cannot be remapped to any other location but to application registers. Any remote inbound access matching the BAR0 region will automatically be routed to these registers. It allows RC devices to control EP devices in absence of dedicated software running on EP. For RC and EP, this mapping of BAR0 to registers allows the message signaled interrupts to work. There is no support to disable BAR0 accesses from reaching the application registers.

### 2.7.2.3  Using BAR1 Value As Start Address

The inbound address translation for BAR1 has additional capability of using the value of BAR1 register (from PCIe configuration space) as the start address for inbound address translation. This feature can be activated by leaving the start address of the corresponding inbound translation region (one of the four regions associated to BAR1) programmed with zero. When an incoming read/write access matches BAR1 and the inbound region's BAR match (IB_BARn) is set to 1 with start address programmed to zero, the BAR1 value is used to compute the translated address.

**Note**—If this feature is used, only one inbound translation window is available and it will be relative to BAR1 programmed value. No other BARs or inbound translation windows can be used if BAR1 is designated as the reference for inbound address translation.

### 2.7.3  Use of Base Address Registers (BARs)

Base address registers (BARs) are treated differently depending upon the type — Type 0 or Type 1 — of the configuration space supported by the device. Typically, a RC module uses Type 1 and EP configuration uses Type 0 configuration space.

The primary difference between Type 0 and Type 1 configuration space is in regarding TLP filtering. When a TLP is to be routed by its address, the address range in the BAR will decide whether the TLP is rejected or accepted. In the case of an EP, if the address is in the range configured in the BAR, the EP will accept the TLP and pass it to the internal bus side. In the case of RC, if the address is in the range configured in the BAR or is outside of the range defined by the three Base/Limit register sets (non-prefetchable memory, prefetchable memory, and I/O), and then that TLP is accepted.

These three Base/Limit registers contain the address ranges setup for all of the downstream devices.

- Memory Limit and Base Register (MEMSPACE)
- Prefetchable Memory Limit and Base Register (PREFETCH_MEM). Prefetchable Memory Base Upper 32 bits Register (PREFETCH_BASE) and Prefetchable Limit Upper 32 bits Register (PREFETCH_LIMIT) are used as well for 64-bit addressing.
- IO Base and Limit Upper 16 bits Register (IOSPACE)

Therefore, any TLP coming into the RC with the downstream address within the range means that it is intended for a device somewhere downstream of RC, and therefore it has been misrouted and will be rejected by the RC filter. TLPs with an address outside the range of the Base/Limit registers are given to the RC application.

For an RC port, the use of BAR register is applicable only when the RC port itself has a memory that can be targeted from the PCIe link. The BAR range in the RC is normally must be outside of the three Base/Limit regions. In a normal system setup, the system memory would be connected on the host/CPU bus. This host/CPU is the initiator/completer for TLPs. To access a downstream device, the host/CPU would initiate a request that is forwarded by the RC port to either a switch or an EP device. Completions (if any), for this request are given to the host/CPU application. If a downstream EP wants to access the system memory, it would initiate a request with an address range outside of the RC port Base/Limit registers. That TLP will be forwarded to the host/CPU interface. If the BAR range in RC is inside of the three Base/Limit regions, the TLP with the address targeted to the RC BAR range will still be accepted by RC.

### 2.7.3.1 BAR Mask Registers

The BARs are accessible in both EP and RC modes by the device itself via internal bus. The BARs in EP devices are usually programmed by RC during the PCIe configuration process. Because the PCIESS is a Dual Mode (DM) core, prior to the time when PCIe configuration starts, the PCIESS EP device has the opportunity to modify the behavior of BAR registers prior to RC starts the enumeration process and assignment of PCIe Base Address. Using the BAR Mask registers, which are overlaid on the BAR registers, the software on the EP device can configure the amount of address space that the EP will request from the RC during configuration for each of its BARs. The software can also modify the BAR types and can enable/disable the BARs. That is, the read-only bit fields of BARs are user-programmable and is required to be programmed prior to the enumeration process begins.

Note that the BAR Mask registers are accessible, for configuration purposes of the BARs, in both EP and RC mode and only when DBI_CS2 (bit 5 in Command Status Register (CMD_STATUS)) is enabled. The software needs to clear CMD_STATUS [DBI_CS2] after initial configuration prior to RC starts enumeration. The software

needs to always read back the written value to DBI_CS2 after modification to ensure that the write has completed because the read will not happen until the write completion. And the read back should be done in both cases of enabling and disabling the DBI_CS2 bit.

Please also note that the BAR Mask registers are writable but not readable. The read back value will be the BAR registers values instead of BAR Mask registers.

In order to verify if the BAR Mask registers have been set correctly. we can write test pattern into BAR registers and see if the bits have been masked correctly.

For example, if BAR0 is 32-bit BAR, enable DBI_CS2 bit, write BAR0 Mask register = 0x00003FFF, disable DBI_CS2 bit, write BAR0 = 0xFFFFFFF0, then BAR0 should be read as 0xFFFFC000, as the lower bits of BAR0 have been masked.

Another example, if BAR1 is 32-bit BAR, and BAR1 Mask register = 0x0007FFFF, write BAR1 = 0xFFFFFFF0, then BAR1 should be read as 0xFFF80000. Please see the Registers chapter for details of BAR Mask registers and BAR registers.

It is important to not attempt modification of BAR Mask registers from serial link side as unpredictable behavior may occur and the system would become unusable.

The BAR0 Mask register in both RC and EP modes is fixed because BAR0 is completely dedicated to accessing the application registers. Only bits [31:20] in BAR0 are configurable. Please refer to section 2.7.2.2 for details.

### 2.7.3.2 Example BAR Setup

Figure 2-4 shows an example configuration of the six BARs and their corresponding BAR Mask Registers (EP mode). The example configuration includes:

- One 64-bit memory BAR (non-prefetchable)
- One 32-bit memory BAR (non-prefetchable)
- One 32-bit I/O BAR

**Figure 2-4      Example Base Address Register Configuration**

## 2.8 PCIe Address Spaces

From the PCIESS perspective, the internal bus addressable resources are categorized into two address spaces. That is, the PCIESS has two address spaces. The first (Address Space 0) is dedicated for local application registers, local configuration accesses, remote configuration accesses and remote IO accesses (RC only). The second (Address Space 1) is dedicated for data transfer.

### 2.8.1 Address Space 0

Address Space 0 occupies a contiguous 16 KB of memory partitioned into 4 sections/regions with equal sizes (each 4 KB long). Figure 2-5 illustrates the relationship of the various address regions within the Address Space 0.

**Figure 2-5    Mapping of the PCIESS Address Space 0**
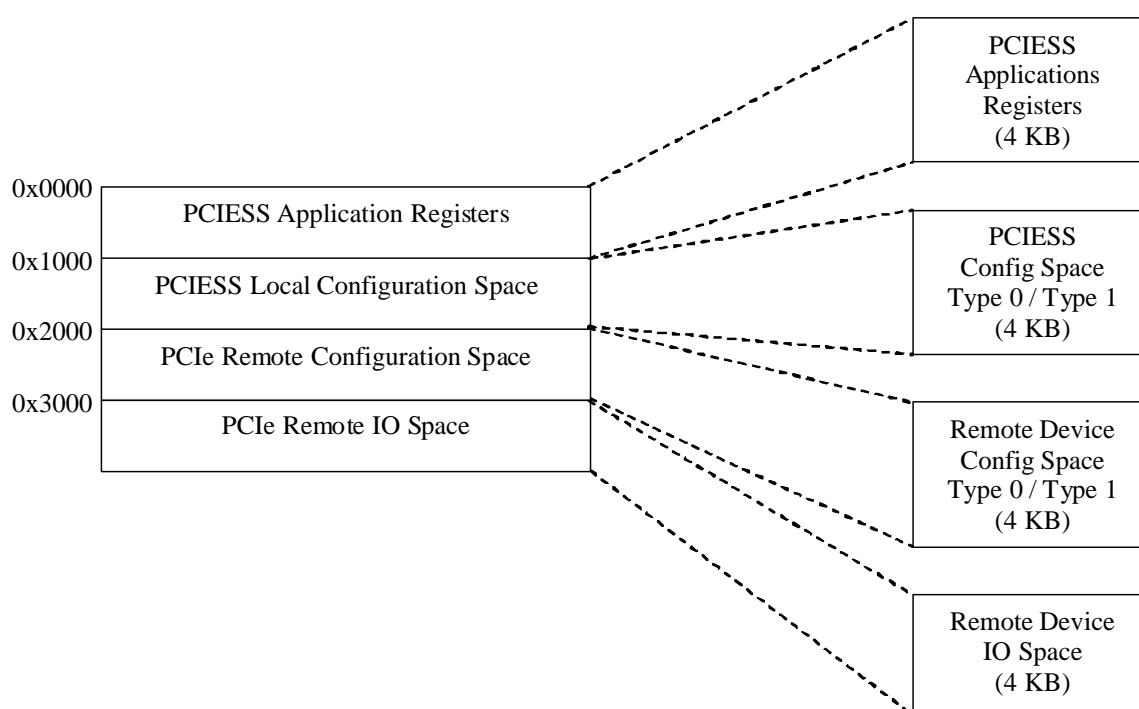


Each of the four regions is meant for accessing a set of registers.

1. **PCIESS Application Registers** — These registers are used to configure and monitor various settings within the PCIESS. These registers are specific to the application needs and are not related to the PCIe configuration registers. Note that all PCIESS application registers should be accessed in 32-bit mode even if the PCIESS data bus is wider than 32-bits.

   The Application registers are accessible in multiple ways depending on the point of origin of such accesses. When accessed from the device internal bus slave, these registers are accessed via the 4KB space in Address Space 0. When accessed from the PCIe serial link side, the application registers are mapped to BAR0 of an EP as well as RC. In addition, a RC can access these registers in a PCIe EP via the upper 1 KB space of the PCIe configuration space. The offsets of the registers do not change irrespective of what the mode of accessing these registers is.

2. **PCIe Local Configuration Registers** — These registers are used to read the settings of configuration registers of the local PCIe device. Prior to PCI Express configuration is complete, these registers can also be written to. Depending upon whether the PCIESS is configured as RC or EP, the layout of these registers is either Config Space Type 0 or Type 1. Note that all accesses on PCIe local configuration registers must be made in 32-bit mode even if the PCIESS data bus is wider than 32-bits.

3. **PCIe Remote Configuration Registers** — PCIe remote configuration registers are accessed by first setting up the bus number, device number and function number of the remote PCIe device in one of the application register and then accessing the PCI remote configuration registers as if it were the PCIe configuration space of a single PCIe function. The layout of the remote configuration registers varies based upon whether the device is an end point or a PCIe switch.

4. **PCIe IO Access Window** — A 4KB region is dedicated for remote IO accesses when the PCIESS is in RC mode. Any access made on this space becomes an IO access. The actual address in the TLP gets its base address from the IO_BASE Register value and an offset that is directly derived from the address of the internal bus access in this 4KB space.

None of the four address ranges described above support burst transactions. So, only single 32-bit transactions should be issued to these addresses. These addresses should also be configured as non-cacheable address space.

### 2.8.1.1 Remote Configuration and I/O Requests (Caution)

Because the remote configuration and IO transaction windows are directly mapped to internal bus space, the software must take caution not to access these spaces when there is no operational PCIe link. No response may be generated for such transactions. It is recommended that checks be built into software to avoid remote accesses in the absence of an operational link.

### 2.8.1.2 Organization of Configuration Registers

As showed in Table 2-7, the registers for various PCIe capabilities are linked to each other via address offsets specified in the registers themselves.

**Table 2-7    Layout of PCIe Configuration Registers**

| Offset From Start of Configuration Space | Register Block |
|---|---|
| 0x000 | PCI-Compatible Header (Type 0/1) |
| 0x040 | Power Management Capabilities Registers |
| 0x050 | MSI Capabilities Registers |
| 0x070 | PCI Express Capabilities Registers |
| 0x100 | PCI Express Extended Capabilities Registers |
| 0x700 | Port Logic Registers |
| **End of Table 2-7** | |

### 2.8.2 Address Space 1

The second address space is used for data transfer. The BAR values setup in PCIe local configuration registers define where within the memory map of the CPU on root complex side the end points are located. All locations other than what is setup in BAR registers of the end points are on the root complex side.

The Address Space 1 is mapped to multiple devices in RC mode. Each remote device could be allocated a portion of this memory space and any transaction that is targeted to this address space gets converted to a PCIe transaction targeted to the appropriate remote PCIe device.

## 2.9  DMA Support

The PCIESS does not have DMA capabilities built into it. It has internal bus interface slave and master ports connected to the chip-level interconnect. If the PCIESS is accessing the external PCIe device, a DMA engine can make burst-data read/writes on the slave port of the PCIESS. And the master port on the PCIESS can initiate reads/writes to memory on behalf of a remote PCI Express device without need of DMA. Both of the master and slave ports can be used, no matter the PCIESS is in RC or EP mode.

### 2.9.1  DMA Support in RC Mode

When operating in root complex mode, a DMA controller internal to the device (outside of the PCIESS) can perform DMA to and from any remote device located on the PCI Express fabric. The memory address of such devices is available to the software via the PCI Express bus enumeration procedure. In addition, the PCIe subsystem has a provision to perform memory address translation on outbound requests. Thus, the software is able to map different memory regions in its memory map to correspond to different addresses (and different access types) on the PCI Express side.

There are bandwidth implications of using an external DMA. If the PCIe core has been programmed to establish a link in the PCIe 2.5 Gbps rate, then the DMA controller that drives the PCIESS slave port must be able to write/read data at about 85% of 2 Gbps bandwidth per PCIe link. For a PCIe link speed of 5.0 Gbps, the DMA controller must be able to provide bandwidth of about 85% of 4 Gbps per PCIe link.

In addition, the master port on PCIe port can issue read/write accesses that have been initiated by a remote PCI Express device. The interconnect fabric should provide sufficient capacity to serve 85% of 2 Gbps (4 Gbps in Gen2 mode) per PCIe link in each direction.

### 2.9.2  DMA Support in EP Mode

When operating as a PCIe end point, the device will be located in PCIe memory map at location programmed in the Base Address Registers by the PCIe root complex device. Any PCIe transactions destined for the device from the upstream ports will get transferred to the master port on the PCIe subsystem. Similarly, any transactions originating from the software will be sent over to PCIe link.

In end point mode, the PCIESS provides address translation functionality. It is possible to map IO, config and memory accesses originating on PCI Express side to memory accesses with different address on the internal bus side. These address ranges are configurable through application registers.

The approximate data rate for each of these transactions will be about 85% of 2 Gbps (4 Gbps in Gen2 mode) in each direction on each PCIe lane.

### 2.9.3  EDMA Transfer Examples

#### 2.9.3.1  Memory Write Transfer

The following example details how to perform an EDMA transfer from the device source (for example, L2 or EMIF) to PCIe memory. For the EDMA configuration, please see the EDMA user's guide.

**Example 2-4     Memory Write**

```
/* Configure PCIESS module and prepare transfer parameters*/
/* Initialize the PCIESS module */
/* Define payload size and buffer size */
pcie_max_payload = 128; /* Assume pcie maximum payload size is 128 bytes */
buff_size= 2048; /* Assume the transfer buffer size 2048 bytes */
/* Define PCIe data starting address for EDMA transfer */
pcieAddr = <PCIe data address defined in device-specific data manual>
...


/* Setup EDMA module and enable the DMA region */
/* setup EDMA PARAM */
/* The OPT register contains following bit fields:
ITCCHEN = 0x0; TCCHEN = 0x0; ITCINTEN = 0x0; TCINTEN = 0x1;
TCC = 0x0; TCCMODE = 0x0; FWID = 0x0; STATIC = 0x0; SYNCDIM = 0x1; DAM
= 0x0; SAM = 0x0 */
paramSetup_pcie.option      = CSL_EDMA3_OPT_MAKE(0,0,0,1,0,0,0,0,1,0,0);
paramSetup_pcie.aCntbCnt    = CSL_EDMA3_CNT_MAKE(pcie_max_payload,
buff_size/pcie_max_payload);
paramSetup_pcie.srcDstBidx  =
CSL_EDMA3_BIDX_MAKE(pcie_max_payload,pcie_max_payload);
paramSetup_pcie.srcDstCidx  = CSL_EDMA3_CIDX_MAKE(0,0);
paramSetup_pcie.cCnt        = 1;
paramSetup_pcie.linkBcntrld =
CSL_EDMA3_LINKBCNTRLD_MAKE(CSL_EDMA3_LINK_NULL,0);
paramSetup_pcie.srcAddr     = (Uint32)srcAddr;
paramSetup_pcie.dstAddr     = (Uint32)pcieAddr;


/* Setup EDMA Channel */
/* Enable the EDMA Channel */
/* Wait for a transmit completion */
```

**End of Example 2-4**

### 2.9.3.2 Memory Read Transfer

The following example details how to perform an EDMA transfer from PCIe memory to the device destination (for example, L2 or EMIF). For the EDMA configuration, please see the EDMA user's guide.

**Example 2-5     Memory Read**

- - - - - - - - - - - - - - - - - - - - - - - -

```
/* Configure PCIESS module and prepare transfer parameters*/
/* Initialize the PCIESS module */
/* Define payload size and buffer size */
pcie_max_payload = 128; /* Assume pcie maximum payload size is 128 bytes */
buff_size= 2048; /* Assume the transfer buffer size 2048 bytes */
/* Define PCIe data starting address for EDMA transfer */
pcieAddr = <PCIe data starting address defined in device-specific data manual>
...


/* Setup EDMA module and enable the DMA region */
/* Setup EDMA PARAM */
/* The OPT register contains following bit fields:
ITCCHEN = 0x0; TCCHEN = 0x0; ITCINTEN = 0x0; TCINTEN = 0x1;
TCC = 0x0; TCCMODE = 0x0; FWID = 0x0; STATIC = 0x0; SYNCDIM = 0x1; DAM
= 0x0; SAM = 0x0 */
paramSetup_pcie.option     = CSL_EDMA3_OPT_MAKE(0,0,0,1,0,0,0,0,1,0,0);
paramSetup_pcie.aCntbCnt   = CSL_EDMA3_CNT_MAKE(pcie_max_payload,
buff_size/pcie_max_payload);
paramSetup_pcie.srcDstBidx =
CSL_EDMA3_BIDX_MAKE(pcie_max_payload,pcie_max_payload);
paramSetup_pcie.srcDstCidx = CSL_EDMA3_CIDX_MAKE(0,0);
paramSetup_pcie.cCnt       = 1;
paramSetup_pcie.linkBcntrld =
CSL_EDMA3_LINKBCNTRLD_MAKE(CSL_EDMA3_LINK_NULL,0);
paramSetup_pcie.srcAddr    = (Uint32)pcieAddr;
paramSetup_pcie.dstAddr    = (Uint32)dstAddr;


/* Setup EDMA Channel */
/* Enable the EDMA Channel */
/* Wait for a transmit completion */
```

**End of Example 2-5**

- - - - - - - - - - - - - - - - - - - - - - - -

## 2.10 PCIe Transactions

### 2.10.1 Transaction Requirements

There are some requirements that must be adhered to while issuing transactions to the PCIESS internal bus interface.

#### 2.10.1.1 Bus Mastering

An end point that is capable of becoming a bus master and initiate transactions in upstream direction must have its Bus Master Enable bit set in configuration space registers (STATUS_COMMAND[BUS_MASTER]). If transactions are initiated (while assuming the role of a RC or EP) before enabling bus mastering capability, then transactions will not start until the bus master enable is set. There is no timeout or error response generated when transactions does not go out because of bus master bit not being set.

#### 2.10.1.2 Address Alignment Requirements

The limit on transaction size on the internal bus interface is 128 bytes. So, the transactions must be 128 bytes or smaller on the internal bus interface. If a transaction received in the inbound direction crosses a 128-byte-aligned address, then the PCIESS master interface can split such transaction into two transactions at the 128 byte boundary.

If the starting address is not aligned to an 8-byte boundary, then the maximum transaction size is reduced to 120 bytes. Unspecified behavior will occur if misaligned transactions in outbound direction are not limited to 120 byte.

#### 2.10.1.3 Burst Type Requirements

The PCIESS does not support *fixed* or *wrap* burst types. Transactions that required any burst type other than incremental burst type will result in unspecified behavior, possibly bus lock up.

#### 2.10.1.4 Read Interleaving

Read interleaving refers to the process of returning split read responses from multiple transactions. This implies that read data is not guaranteed to be sent in sequential order (data for one transaction to be sent completely before the next). PCIe core is guaranteed to not interleave read responses if the outbound read command/transaction size does not exceed the max transaction size configured in the PCIe core. Currently this is configured as 128 bytes.

#### 2.10.1.5 Byte Strobe Requirements

For any type of write transactions, the byte enables can have only a single unbroken string of 1s. In other words, in a transaction, if a byte's write strobe is set, then all following bytes must have write strobe set until the last byte with write enabled. *Holes* or *0s* in between the byte enables are not allowed.

Because the internal bus width is greater that 32-bit, the TLP size will not be 1 (PCIe counts in 32-bit units) and therefore, it is through the FBE/LBE (First/Last Byte Enable) that the actual data transfer size is controlled.

### 2.10.2 Support for Endian Modes

The PCIe core supports only little endian mode (at the pins) but the PCIe subsystem (PCIESS) supports both little endian and big endian modes.

#### 2.10.2.1 Endian Modes for Inbound Transactions

Inbound write and read requests from PCI Express link can be as small as a byte or the maximum configured PCI Express payload size. The PCIESS is configured in big endian or little endian mode through device boot configuration module (see the device-specific data manual for details). No endian conversion is required if the system/application operates in little endian mode.

If the system is configured in big endian mode, then little endian to big endian conversion is required as shown in table below. There are multiple operating modes that modify how the endian conversion behaves. The byte swapping modes are programmed via register write to the PCIESS endian configuration register (ENDIAN[1:0]). See Registers chapter for details. The mode field is valid only when device is in big endian mode. The byte enables are swapped with the data based on the swapping sequence.

**Table 2-8    Big Endian Byte Swap for Inbound Transactions**

| Endian Mode | Swizzle Mode | | Byte Lanes | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7 | | | | | | | 0 |
| Little Endian | No swizzle | NA | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Big Endian | Swap on 8-byte | D (0x3) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Big Endian | Swap on 4-byte | C (0x2) | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 |
| Big Endian | Swap on 2-byte | B (0x1) | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 |
| Big Endian | Swap on 1-byte (default) | A (0x0) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **End of Table 2-8** | | | | | | | | | | |

#### 2.10.2.2 Endian Modes for Outbound Transactions

The PCIESS uses a single slave interface to access both the MMR register space (32-bit access) and memory (32-bit or 64-bit access). The MMR and non-prefetchable memory accesses are 32-bit accesses that are performed on 32-bit values using a 64-bit data bus, swapping will always be on a 32-bit boundary shown as mode C in Table 2-9.

The endian conversion for the 64-bit memory accesses is implemented using the swapping sequence tabulated below (Table 2-9). By default, the swapping is on a byte boundary shown as Mode A (default mode) in the table. The byte swapping modes are programmed via register write to the PCIESS endian configuration register (ENDIAN[1:0]). This mode setting is valid only when device is in big endian mode.

**Table 2-9    Big Endian Byte Swap for Outbound Transactions**

| Endian Mode | Swizzle Mode | | Byte Lanes | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 7 | | | | | | | 0 |
| Little Endian | No swizzle | NA | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Big Endian | Swap on 8-byte | D (0x3) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Big Endian | Swap on 4-byte | C (0x2) | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| Big Endian | Swap on 2-byte | B (0x1) | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 |
| Big Endian | Swap on 1-byte (default) | A (0x0) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **End of Table 2-9** | | | | | | | | | | |

## 2.11 Operations

### 2.11.1 PCIe as Root Complex

#### 2.11.1.1 Initialization Sequence

The initialization sequence is as follows:

1. Provide PLL reference clock to SerDes inside the PCIESS. See Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

2. Turn on the power domain and clock domain of the PCIe module. Or set the PCIESSEN to 1 in the device level register DEVSTAT to enable the PCIe module. Please see the device-specific data manual and PSC user's guide for details.

3. Set PCIESSMODE[1:0]to 0x10 in the device level register DEVSTAT to operate the PCIESS in RC mode. Please see Section 1.4.1 and device-specific data manual for details.

4. Program PLL settings and enable PLL from PCIe SerDes configuration register (PCIE_SERDES_CFGPLL). Please see Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

5. Wait until PLL is locked by sampling LOCK bit in the PCIe SerDes Status register (PCIE_SERDES_STS[LOCK]=1). Please see Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

6. Disable link training by de-asserting the LTSSM_EN bit in the PCIESS Command Status Register (CMD_STATUS[LTSSM_EN]=0). Upon reset, the LTSSM_EN is de-asserted automatically by hardware.

7. Program the configuration registers in the PCIESS to desired values.

8. Initiate link training can be initiated by asserting LTSSM_EN bit in the CMD_STATUS register (CMD_STATUS[LTSSM_EN]=1).

9. Insure link training completion and success by observing LTSSM_STATE field in DEBUG0 register change to 0x11.

10. In conjunction with the system software, start bus enumeration and setup configuration space on downstream ports.

11. Continue software handshake and initialization on the remote devices. This includes setting up DMA protocols, interrupt procedures, etc.

12. With the completion of software initialization, DMA accesses can be started on various end points.

#### 2.11.1.2 Configuration Accesses

Configuration accesses are made by RC port to individual function in each downstream EP device to program the PCIe specific operating parameters. In particular, the configuration accesses are used to allocate memory ranges for each downstream device, configure those memory ranges as IO or Memory type, enable bus master capability on the device if necessary and also build a software database of PCIe attributes and capabilities of each downstream device. Each downstream device can be configured through the configuration access region of the PCIESS. The PCIESS converts memory reads and writes on the configuration region of internal bus interface into configuration access on the serial link.

### 2.11.1.3 Memory Accesses

There are two types of memory accesses — inbound and outbound memory accesses. The outbound memory accesses that are initiated by the DMA on the PCIESS slave port and the inbound memory accesses that are initiated by the PCIESS master port targeted to internal memory regions within the internal bus interconnect.

The outbound PCIe memory read and write accesses are made through the PCIESS slave port through the device memory range that is dedicated to data transfers. The read and write transactions on this region are directly mapped to PCI Express space by the PCIESS in conjunction with the outbound address translation mechanism. The completions to the bus transactions are generated by the PCIESS when it receives completions from remote devices. In case of errors or timeouts, an error response is provided. For reads, the error responses span as many phases as there would be data phases if the error had not occurred.

The inbound memory accesses received by the PCIESS on the serial link are initiated by remote PCIe end points that are capable of bus mastership. Such accesses are converted to bus transactions on the PCIESS master port and once a response/completion is received, the PCIESS sends data/response back to the PCIe bus master over the serial links. Typically, the inbound accesses will be targeted to memory space resident on the bus side of the RC port. The locations to which inbound memory accesses map are determined by software. The software must inform the remote devices about what protocol is to be followed so that the remote device will access the relevant memory regions to read or write data/control information. Not all end points have the capability to initiate inbound accesses.

An inbound access cannot cross a 4-KB boundary as per PCIe specifications. In addition, any access that spans a 128-byte boundary in the device memory map can get split into two transactions at the 128-byte boundary.

### 2.11.1.4 I/O Accesses

The PCIESS supports outbound IO accesses in RC mode. These accesses are initiated through a 4-KB memory space and a programmable register. All accesses made to the 4-KB space become IO accesses and the IO base register determines the target address for such accesses. The IO accesses cannot be for more than 32-bits of data aligned at a 4-byte boundary. See the memory map section for location of the IO access address range. Inbound IO accesses are not supported in the PCIESS RC mode.

## 2.11.2 PCIe as End Point

### 2.11.2.1 Initialization Sequence

Upon de-assertion of reset, the PCIESS is configured as end point by chip-level setting of PCIESS inputs. Before a root complex is allowed to access the configuration space of the end point, the following initialization sequence should be followed:

1. Provide PLL reference clock to SerDes inside the PCIESS. See Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

2. Turn on the power domain and clock domain of the PCIe module. Or set the PCIESSEN to 1 in the device level register DEVSTAT to enable the PCIe module. Please see the device-specific data manual and PSC user's guide for details.

3. Set PCIESSMODE[1:0]to 0x0 in the device level register DEVSTAT to operate the PCIESS in EP mode. Please see Section 1.4.1 and device-specific data manual for details.

4. Program PLL settings and enable PLL from PCIe SerDes configuration register (PCIE_SERDES_CFGPLL). Please see Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

5. Wait until PLL is locked by sampling LOCK bit in the PCIe SerDes Status register (PCIE_SERDES_STS[LOCK]=1). Please see Section 2.3 "SerDes Configuration for KeyStone I Devices" on page 2-5 for details.

6. Disable link training by de-asserting the LTSSM_EN bit in the PCIESS Command Status Register (CMD_STATUS[LTSSM_EN]=0). Upon reset, the LTSSM_EN is de-asserted automatically by hardware.

7. Program the configuration registers in the PCIESS to desired values.

8. Initiate link training can be initiated by asserting LTSSM_EN bit in the CMD_STATUS register (CMD_STATUS[LTSSM_EN]=1).

9. Insure link training completion and success by observing LTSSM_STATE field in DEBUG0 register change to 0x11.

10. If further configuration register initialization is required, the application request retry bit (APP_RETRY_EN) should be set in CMD_STATUS register. This will lead to incoming accesses to be responded with the retry response. This feature allows slow devices extra time before the root port assumes the devices to be inactive. Once programming is complete, de-assert the APP_RETRY_EN bit to allow transactions from the root complex

11. Once configuration setup is complete, DMA transactions can begin.

12. Inbound PCIe transactions will arrive at the master port of the PCIESS end point. These will be responded to by the target slave devices and the PCIESS will relay the response back to the PCIe device that initiated the transaction.

13. Outbound PCIe transactions will be targeted to the slave port of the PCIESS end point. These transactions will be serviced only if the PCIESS end point has been given bus master capability by the root complex.

### 2.11.2.2 Configuration Accesses

As an end point, the PCIESS can be only a target of configuration accesses from upstream. Until the link is established and APP_RETRY_EN is disabled, the PCIESS will not respond to configuration accesses. When enabled, the PCIESS will respond to configuration accesses automatically and these accesses do not get relayed to the master interface on the device interconnect side.

End point is not capable of accessing configuration space of devices other than its own. The system software can initialize the read-only fields in the PCIESS configuration space via the slave port before the link training has been initiated. Once the configuration is complete by the PCIe root complex, the system software should not modify the PCIESS configuration parameters. There is no explicit prevention mechanism in hardware to disallow such accesses though.

### 2.11.2.3 Memory Accesses

There are two types of memory accesses — inbound and outbound memory accesses.

An inbound access is typically initiated by a RC port or by another EP that is reaching the PCIESS via a PCI Express switch that supports peer-to-peer access. In either case, the incoming PCIe transaction results in an access on the PCIESS master port. The response to such accesses is relayed back to the originating PCIe device.

In an outbound access, a DMA module or a host CPU initiates a read/write access on the PCIESS slave port. This request is converted into a PCIe memory read/write transaction over the PCIe link. Once the PCIESS receives completion from the remote device, it generates a completion on the internal bus slave port. The software/hardware that initiates the request on the slave port must perform it in a memory region that has been determined previously through software protocols. For example, the software may get information about applicable memory regions from the software that is running on the root complex device.

#### 2.11.2.4 I/O Accesses

The PCIESS in EP mode does not support inbound or outbound IO accesses.

### 2.11.3 Accessing Read-Only Registers in Configuration Space

Some of the register fields provided in the configuration space can be written to prior to bus enumeration via the slave interface of the PCIESS. Note that the hardware does not prevent modification of read-only fields after bus enumeration but it is strongly advised that read-only fields not be written once the bus enumeration is complete.

In addition, the BAR Mask registers can also be programmed by first enabling the DBI_CS2 bit in CMD_STATUS register and then performing writes on the BAR registers. The BAR Mask registers are overlaid on BAR registers. For the BAR Mask registers to be writable, the respective BAR must first be enabled.

### 2.11.4 Accessing EP Application Registers from PCIe RC

The application registers are also mapped at 2K and above address in the configuration space. The RC software can access these registers over the PCIe link provided the registers are programmed to some default values by the boot code in the PCIESS host device that enables link training and TLP exchange.

## 2.12  PCIe Loopback

The PCIe specifications provides loopback support in two ways:
- Through PIPE interface (Link Layer)
- Through PHY loopback capability (PHY Layer)

The PIPE interface loopback requires for two PCIe devices/components to attached to each other in a loopback master and a loopback slave configuration. A loopback master is the component requesting loopback. A loopback slave is the component looping back the data. Note that, regardless with the PCIESS role it assumes (RC or EP), it can assume a role of a loopback master or a loopback slave.

### 2.12.1  PIPE Loopback

The procedure depends upon whether the device is operating in RC or EP mode. In either case, the PCIESS can be loopback master or loopback slave as outlined in the PCIe specifications. Note that this loopback mode cannot be used for looping back transactions. These modes are to be used with PCIe test equipment only for symbol level loopback.

#### 2.12.1.1  Loopback Master

The loopback path when PCIESS is loopback master is:
PCIESS→PIPE (TX)→PCIe Link→Loopback→PCIe Link→PIPE (RX)→PCIESS

The loopback entry procedure when the PCIESS is a loopback master:

**RC Mode**
1. Set Loopback Enable bit (LPBK_EN) in the Port Link Control Register (PL_LINK_CTRL).
2. The link retraining sequence must be initiated by setting Retrain Link bit (RETRAIN_LINK) in the Link Status and Control Register (LINK_STAT_CTRL).

**EP Mode**
1. Set Loopback Enable bit (LPBK_EN) in the Port Link Control Register (PL_LINK_CTRL).
2. Force the LTSSM to be in recovery state via the Link State field (LINK_STATE) in Port Force Link Register (PL_FORCE_LINK).
3. Set Force Link bit (FORCE_LINK) in Port Force Link Register (PL_FORCE_LINK).

Once this is done, devices at the ends of a PCIe link enter the PCIe LTSSM loopback state. The initiator of loopback state is the loopback master and the other device is the loopback slave. Note that it is not possible to send TLPs in this mode and return them via the loopback state of the other device.

#### 2.12.1.2  Loopback Slave

The loopback path when the PCIESS is loopback slave is:
Remote device→PCIe Link→PIPE (RX)→Loopback→PIPE (TX)→PCIe Link→Remote device

If the PCIESS is a loopback slave, then the incoming serial data is routed back to the originating device from the PIPE interface as per PCIe loopback requirements. Typically, PCIe test equipment will be used as loopback master and it will transition the PCIESS into the loopback slave state following which the inbound transactions will be looped back to the test equipment. There is no programming required on the PCIESS to enter loopback in slave mode. PHY support is not required to use this loopback mode.

### 2.12.2 PHY Loopback

The PHY loopback is accomplished by switching the PHY to loopback where the transmitted data is looped back to the receive path at the PHY level. This mode can be used to perform TLP loopback even if there is no link partner. It is, however, possible to set up only when the PCIESS is used in RC mode. This limitation is because of the fact that link training cannot occur between two upstream ports; at least one port must be a downstream port.

The loopback path in PHY loopback is:
Slave interface (TX)→PCIESS→PHY (TX)→PHY (RX)→PCIESS→Master Interface (RX)

The procedure is similar to the one for PIPE (Link Layer) interface but the PHY programming is used instead. This mode is entered by configuring the SerDes configuration registers. Both the Transmit and Receive paths must be set in loopback mode to enable PHY loopback mode.

PHY loopback configuration for KeyStone I devices:
- Please follow the steps (up to Step 7) described in Section 2.11.1.1 "Initialization Sequence" for RC mode.
- Before executing Step 8, we need to configure the SerDes configuration registers to set Transmit and Receive paths to be in loopback mode.
    - Enable TX loopback and RX loopback in SerDes Configuration Lane 0 Register (SERDES_CFG0[TX_LOOPBACK]=0x2 and SERDES_CFG0[RX_LOOPBACK]=0x3).
    - Disable loss of signal detection in SerDes Configuration Lane 0 Register (SERDES_CFG0[RX_LOS]=0).
- Do the same configuration for SerDes Configuration Lane 1 Register if testing in two lanes mode.
- Then initialize link training (Step 8 in Section 2.11.1.1).
- Before checking link training status (Step 9 in Section 2.11.1.1), we need to skip Detect state in LTSSM and force link to begin with POLL_ACTIVE state.
    - Set 0x2 (POLL_ACTIVE state) in LNK_STATE field of Port Force Link Register (PL_FORCE_LINK[LINK_STATE]=0x2).
    - Set 0x1 in FORCE_LINK field of PL_FORCE_LINK register to force the link to the state specified by LINK_STATE field (PL_FORCE_LINK[FORCE_LINK]=0x1).
- Then insure link training completion (Step 9 in Section 2.11.1.1)
- With the initialization above, the PCIESS should be configured in PHY loopback mode.

Note that there are several other requirements for this to work correctly:

1. The PHY should be configured to operate in loopback mode before any transaction is sent out. Recommended approach is to set loopback before link training. Otherwise, any transactions that were not looped back will cause sequence numbers to increment on transmitter but not on receiver and all following transactions will be dropped because of sequence number mismatch.

2. The BAR0 and BAR1 values (if programmed) should not match the address for the transaction that is issued on slave interface. Otherwise, it will not get to the master port but to internal registers.

3. The memory base/limit registers should not be configured such that the address of the transaction lies in the range specified by the memory base/limit registers. Otherwise, the transaction will be discarded as a misrouted packet.

4. It is not possible to use configuration type transactions in this mode as RC cannot be a target of configuration transactions. Such transactions will be invalid and loopback will not work.

## 2.13 Reset Considerations

The PCIESS supports the conventional reset mechanism that is specified within the PCI Express specification. Both hardware and software reset are supported.

No support for the functional level reset is needed because the PCIESS supports a single function.

### 2.13.1 Hardware Reset Considerations

#### 2.13.1.1 Power-On Reset

The power-on reset is used as cold reset of the PCIESS. The entire device is reset when power-on reset is asserted. The device level power-on reset acts as the power-on reset for the PCIESS. Note that the PCIESS will require power-on reset in case the device software transitions it to certain power-down states that mandate power-on reset to exit.

#### 2.13.1.2 System Reset

The system reset is a warm reset, which includes the hard reset and soft reset. The hard reset resets the entire PCIESS. Soft reset resets the entire PCIESS except the sticky bits in MMRs and power management logic inside the PCIe. Please see the device-specific data manual and PLL user guide for details.

### 2.13.2 Software Reset Considerations

The PCIESS module contains a software reset (INIT_RST) bit in the Reset Command Register (RSTCMD) that is used to issue a hot reset. In general, this reset is software controlled procedure and can be issued only by a root complex in a PCIe network as the propagation is downstream only. As a result of a reset, the PCIESS module register values go to their reset state except the sticky bits. The default states of the registers are shown in Chapter 3 "Registers" on page 3-1.

> **Note—**After a reset, the software must wait at least 100 ms before attempting any PCIe transaction on the device that has been reset. If the downstream device does not respond to transaction packets, it must not give up until 1 second plus an additional 50% (0.5 second) time has elapsed.

### 2.13.3 Power Domain and Module State Transitions Considerations

Turning off the PCIe power domain and module state and turning them back on will reset the PCIe module to initial state. However, the user must ensure that all of the outbound and inbound traffic on PCIe link is stopped from both the local device and the remote device. If a system hang occurs from improperly turning off the PCIe module, the user may need to apply device power-on reset to recover the system.

To turn on and off the power domain and module state, the user must follow the transition sequence and check the transition status bit field (GOSTAT) as mentioned in PSC user guide. There is no additional delay requirements from when the PCIe power domain is turned on until it is ready for use.

After turning the PCIe module back on, the link training must be restarted following any internal configuration that software may need to perform.

## 2.14  Interrupt Support

The PCIESS provides a total of fourteen interrupts to the device interrupt controller

Both message signaled interrupt (MSI) and legacy interrupt are handled by the PCIESS. When operating as an EP, the PCIESS is capable of generating MSI or legacy interrupt, depending on the role it is assuming. Note that one PCIe component can not generate both types of interrupts. It is either one or the other. The interrupt type an EP generates is configured during configuration time. When operating as RC, the PCIESS is capable of handling both MSI and legacy interrupts. This is because when operating as RC it should be able to service both PCIe end points as well as legacy end points.

### 2.14.1  Interrupt Allocation

Multiple events from the PCIESS are mapped to CPU subsystem in the device. The PCIESS provides a total of fourteen interrupts to the device interrupt controller. Some of the events are meaningful based on the role the PCIESS assumes (RC or EP). The interrupts and their underlying events are listed below.

**Table 2-10        PCIESS Interrupt Events**

| Interrupt Event Number | Interrupt Description |
|---|---|
| 0 | PCIe express legacy interrupt mode - INTA (RC mode only) |
| 1 | PCIe express legacy interrupt mode - INTB (RC mode only) |
| 2 | PCIe express legacy interrupt mode - INTC (RC mode only) |
| 3 | PCIe express legacy interrupt mode - INTD (RC mode only) |
| 4 | MSI interrupts 0, 8, 16, 24 (EP/RC modes) |
| 5 | MSI interrupts 1, 9, 17, 25 (EP/RC modes) |
| 6 | MSI interrupts 2, 10, 18, 26 (EP/RC modes) |
| 7 | MSI interrupts 3, 11, 19, 27 (EP/RC modes) |
| 8 | MSI Interrupts 4, 12, 20, 28 (EP/RC modes) |
| 9 | MSI Interrupts 5, 13, 21, 29 (EP/RC modes) |
| 10 | MSI Interrupts 6, 14, 22, 30 (EP/RC modes) |
| 11 | MSI Interrupts 7, 15, 23, 31 (EP/RC modes) |
| 12 | Error Interrupts<br>[0] System error (OR of fatal, nonfatal, correctable errors) (RC mode only)<br>[1] PCIe fatal error (RC mode only)<br>[2] PCIe non-fatal error (RC mode only)<br>[3] PCIe correctable error (RC mode only)<br>[4] AXI Error due to fatal condition in AXI bridge (EP/RC modes)<br>[5] PCIe advanced error (RC mode only) |
| 13 | Power management and reset event interrupts<br>[0] Power management turn-off message interrupt (EP mode only)<br>[1] Power management ack message interrupt (RC mode only)<br>[2] Power management event interrupt (RC mode only)<br>[3] Link request reset interrupt (hot reset or link down) (RC mode only) |
| **End of Table 2-10** | |

### 2.14.2  Interrupt Generation in EP Mode

When the PCIESS is operating as an EP, either the legacy interrupts or the MSI interrupts can be triggered to the upstream ports (eventually leading to an interrupt in RC device). As per PCIe specifications, each PCIe function may generate only one of the legacy or MSI interrupt types as decided during configuration period.

#### 2.14.2.1  Legacy Interrupt Generation in EP Mode

The end point can trigger generation of a PCI legacy interrupt at the root complex via an in-band Assert_INTx / Deassert_INTx message. The actual interrupt that is generated on RC port is based on the configuration of the EP that generates the interrupt and it could be one of INTA, INTB, INTC, or INTD. Please see the interrupt-related registers in configuration space registers for details.

To generate an interrupt, following steps are required:
1. Legacy interrupt generation should be enabled via LEGACY_X_IRQ_ENABLE_SET register (X=A/B/C/D)
2. Write 0x1 to EP_IRQ_SET register to enable the legacy interrupt
3. An assert INTA/B/C/D message is automatically sent.
4. Write 0x1 to EP_IRQ_CLR register to disable the legacy interrupt by sending a deassert INT A/B/C/D message.

Once an assert message has been generated, it cannot be generated again until a deassert message is generated. Thus, only one interrupt can be pending at a time. The pending status can be checked in EP_IRQ_STATUS register.

There is no hardware input port provided that will allow generation of legacy interrupts on the EP port.

> **Note—**The interrupt messaging mechanism makes it unfeasible to guarantee a time of delivery of the interrupt unlike in conventional designs where the interrupt line is often electrically connected to the final destination.

#### 2.14.2.2  MSI Interrupt Generation in EP Mode

MSI interrupts are generated by a PCIe 32-bit memory write transaction to a pre-determined address with pre-determined data. The PCIe system software configures the address and the data that is to be used in the memory write transaction at the time of initialization of the EP device. The MSI scheme supports multiple interrupts and each device can request up to 32 interrupt vectors even though the allotted interrupts may be less than the requested number.

To generate MSI interrupts, following steps need to be taken:
1. Ensure that the MSI support has been enabled in the device (Set MSI_EN bit in MSI Capabilities Register (MSI_CAP). Legacy interrupt must be disabled)
2. Read the value of the MSI Address Register in the local PCIe configuration space (Read the value of MSI Lower 32 Bits Register (MSI_LOW32) for 32-bit addressing or MSI Upper 32 Bits Register (MSI_UP32) and MSI_LOW32 together for 64-bit addressing (64BIT_EN bit is enabled in MSI_CAP register))
3. Read the value of the MSI Data Register in the local PCIe configuration space (Read the value of MSI Data Register (MSI_DATA))

4. Determine the number of MSI vectors allocated (and the number requested) to the device

5. Depending upon the number of MSI interrupts allocated, issue a memory write transaction with the address the same as MSI Address Register and the data the same as the MSI Data Register. In the data, the LSBs can be modified to reflect the appropriate MSI event that needs to be notified to root complex

6. The memory write transaction can also be optionally routed through the outbound address translation interface if the destination PCIe address is not directly accessible.

The memory write transactions to generate MSI interrupts in RC are actually targeted at MSI_IRQ register. The MSI interrupt is generated as a result of the one of 32 events that is triggered by a write of MSI vector value to MSI_IRQ register in RC. Before the end point devices can issue MSI interrupts, the MSI address and data registers must be configured by system software to make sure the MSI_IRQ registers could be accessed correctly with proper MSI vector value.

If there is no system software supported, the user application needs to make sure EP could issue memory write transaction to MSI_IRQ register in RC with proper MSI vector value to generate MSI interrupt in RC.

For more details about how MSI interrupts are expected to behave, please see the PCIe standard specifications.

As per the PCIe specification, an EP device can generate MSI interrupts only to RC. However, the PCIESS has a provision to allow generation of MSI interrupts from an EP to another EP. To generate an interrupt to another EP, instead of doing the memory write to a register in RC memory space, an EP can target this memory write to an analogous register in another EP device that integrates a PCIESS. This memory write would be to the BAR0 memory space where all registers are located.

**Note—**There is no hardware input port to allow generation of MSI interrupts on the EP port.

### 2.14.3 Interrupt Generation in RC Mode

In accordance with PCI Express base specifications, root complex ports only receive interrupts. There is no mechanism to generate interrupts from RC port to EP mode as per PCIe specification. However, the PCIESS does support generation of interrupts from RC to EP. The behavior is similar to generation and reception of MSI interrupts in RC mode except for the fact that this functionality is enabled in EP mode as well.

The RC device can perform a memory write into the MSI_IRQ register over the PCIe link to generate one of 32 EP interrupts. Note that the PCIESS will follow PCIe MSI rules and will not necessarily accumulate multiple writes to the same MSI vector. Only one of such writes is guaranteed to be processed and subsequent writes on the same vector arriving before the interrupt status is cleared may be lost.

### 2.14.4 Interrupt Reception in EP Mode

The PCIe specification does not have provision for end points to receive legacy interrupts. As a result, only events other than these are used to map to interrupts. The MSI interrupts are not supported on EP devices as per PCIe specification but the PCIESS does support these interrupts. The MSI interrupt is generated as a result of the one of 32 events that is triggered by a write of MSI vector value to MSI_IRQ register in EP.

These interrupts, delivered via register writes over the serial link, could also be coming from another end point that performs a write to the appropriate interrupt registers in the EP's BAR0 space. It is up to software designers to implement a way to determine the actual source of the interrupt.

Among the 32 MSI interrupt events, every four MSI interrupts share the same interrupt event number in the PCIESS, which could generate interrupt event to CPU. See Table 2-10 "PCIESS Interrupt Events" for details. The MSI interrupts can be enabled by setting corresponding bits in MSIn_IRQ_EN_SET registers. They can be disabled by setting corresponding bits in MSIn_IRQ_EN_CLR registers.

When EP receives the MSI interrupt, the corresponding status bit will be set in MSIn_IRQ_STATUS register. Before the same MSI interrupt event to be generated again, the MSI interrupt status needs to be cleared by writing one to the corresponding bit in MSIn_IRQ_STATUS register and writing the interrupt event number in IRQ_EOI register to indicate the end of the interrupt event. See Table 2-10 "PCIESS Interrupt Events" for the interrupt event number of each MSI interrupt.

#### 2.14.4.1 Hot Reset Request Interrupt

When the link is down, the upstream port may request reset of the end point. This request is terminated as an interrupt to the end point host software. The PCIESS automatically disables LTSSM by de-asserting the LTSSM_EN bit in the CMD_STATUS register and suspends LTSSM in the *detect quiet* state. All outstanding transactions are errored out on the slave port and further transactions are not generated on the master port. Once the transactions are completely stopped through the internal bus disconnect protocol, the software should issue a local reset to the PCIESS. The re-initialization process may then be started.

### 2.14.5 Interrupt Reception in RC Mode

#### 2.14.5.1 Legacy Interrupts Reception in RC Mode

Any of the four legacy interrupts may be generated by the PCIESS. Each interrupt can originate from multiple end point devices. The software should service these by probing the interrupt registers in each downstream device's configuration space. When all devices have been serviced, the last device serviced will send an interrupt deassert message, which will clear the interrupt.

The interrupt request signal at the PCIESS boundary is a pulse signal that is triggered each time an assert interrupt message is received. The interrupt pending signal is a level signal that is high as long as the interrupt has not be serviced and the interrupt status not cleared through a register write.

Note that the traditional EOI procedure, although implemented, may not operate as expected if the deassert message arrives after the EOI has been issued. This cannot be corrected but only worked around by doing a read on the interrupt register downstream before issuing an EOI. If the EOI register is written to before the deassert message has reached the interrupt logic, the interrupt pending status will not have cleared and the interrupt will retrigger.

In addition, the software drivers for downstream devices must ensure that the data transaction that is expected to complete before interrupt is triggered in RC device's CPU has completed. Because PCIe write transactions are posted, it is not necessary that a write from EP to system memory in RC has completed before a write to MSI interrupt generation register has completed.

The reason is that the PCIe RC or switch could reorder the memory write transactions just posted ahead of previously posted memory write transactions or message transactions. Similarly, message transactions just posted may be ordered ahead of previously posted memory write or message transactions due to the relax ordering feature in PCIe. The relaxed ordering could improve the performance of the post transactions while it could potentially cause issues if the reordering is not permitted in some circumstances.

In the RC mode, the user could disable the relaxed ordering feature by clearing the RELAXED field to 0 in DEV_STAT_CTRL register. The RC could also clear RELAXED field to 0 in TLPCFG register to disable the reordering request for all outgoing TLPs.

### 2.14.5.2  MSI Interrupts Reception in RC Mode

A total of 32 MSI interrupts can be generated from one or more downstream devices. These MSI interrupts represent the MSI interrupts that have been allocated to various downstream devices during PCIe configuration/enumeration procedure. The MSI interrupt is generated as a result of the one of 32 events that is triggered by a write of MSI vector value to MSI_IRQ register in RC.

Among the 32 MSI interrupt events, every four MSI interrupts share the same interrupt event number in the PCIESS, which could generate interrupt event to CPU. See Table 2-10 "PCIESS Interrupt Events" for details. The MSI interrupts can be enabled by setting corresponding bits in MSIn_IRQ_EN_SET registers. They can be disabled by setting corresponding bits in MSIn_IRQ_EN_CLR registers.

When RC receives the MSI interrupt, the corresponding status bit will be set in MSIn_IRQ_STATUS register. Before the same MSI interrupt event to be generated again, the MSI interrupt status needs to be cleared by writing one to the corresponding bit in MSIn_IRQ_STATUS register and writing the interrupt event number in IRQ_EOI register to indicate the end of the interrupt event. See Table 2-10 "PCIESS Interrupt Events" for the interrupt event number of each MSI interrupt.

Each end point can use either MSI interrupts or legacy interrupts, but not both at the same time. MSI interrupts have the same race condition hazard as the legacy interrupts. Hence, software drivers should take precaution.

### 2.14.5.3  Advanced Error Reporting Interrupt

If enabled by software at the time of enumeration, the PCIESS will generate this interrupt to indicate occurrence of errors of various levels of severity. The software can choose not to enable advanced error reporting. But if it is enabled, it must process the interrupt as per PCIe specifications. The error message reporting enable bit must be set in the Root Error Command Register (RC_ERR_CMD) to generate this interrupt. This interrupt can originate either via an MSI message or via an INTx internally in the root complex.

## 2.14.6  Interrupt Reception in RC and EP Mode

These interrupts are common to both RC and EP modes of operation.

### 2.14.6.1  Link down Interrupt

If the PHY link is disconnected, this interrupt will be generated. The expected course of action is to reset the entire PCIESS subsystem and restart. All application states must also be initialized so that the operations can resume following the reset and renegotiation of the link.

### 2.14.6.2  Transaction Error Interrupts

If there is a timeout on PCIe or an abort, this interrupt will be issued. These errors should be handled based upon the severity of the error. For more details, please see the PCIe standard specifications.

### 2.14.6.3  Power Management Event Interrupt

This interrupt is generated to let the software know of power management events.

## 2.15  Power Management

PCI Express has multiple power management protocols. Some of these are invoked by the hardware, such as Advanced State Power Management (ASPM), while others are activated at higher levels via software.

### 2.15.1  Device Power Management

The PCI Express protocol is compatible with all PCI power management functionality. The power states specified are D0, D1, D2, D3Hot, and D3cold. All functions must support D0 and D3 states.

**Figure 2-6      PCI Express Power Management State Transitions**
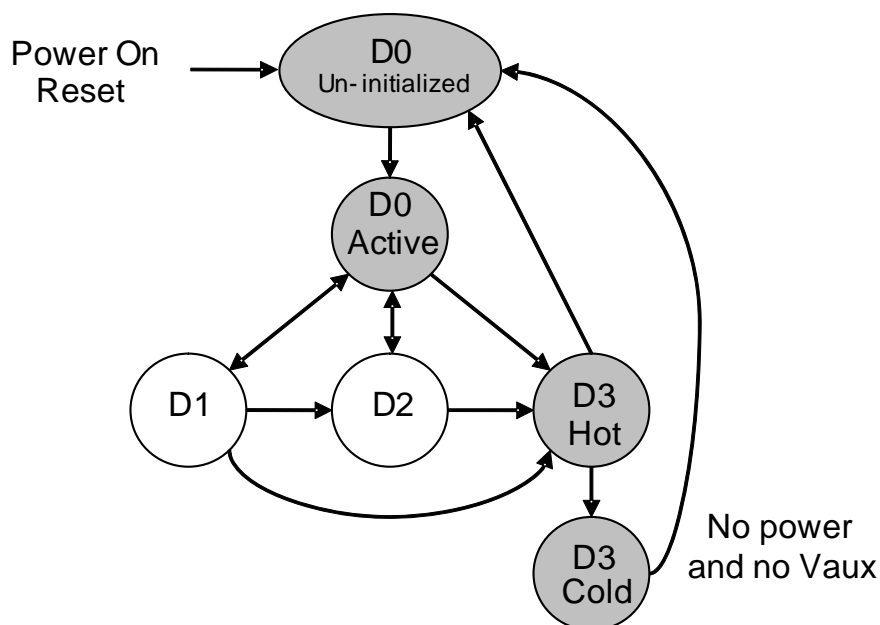


#### 2.15.1.1  D0 State

Upon completion of a reset such as a power-on or hot reset, the function is considered to be in the D0 uninitialized state. Once the function is enumerated, configured, and one or more of the memory spaces are enabled, and IO space enable or bus master enable bits are set, it is considered to be in D0 active state. The D0 active state is the full-operation state of a PCI Express function.

#### 2.15.1.2  D1 State

Support for the D1 state is optional and is primarily driven by software. It is considered to be a light sleep state that provides some power savings compared to D0 state while still allowing transition to D0 state. While in the D1 state, a function must not initiate any request TLPs on the link except for a power management event (PME) message. Configuration and message requests are the only TLPs accepted by a function in the D1 state. All other received requests must be handled as unsupported requests, and all received completions may optionally be handled as unexpected completions. A function's software driver participates in the process of transitioning the function from D0 to D1. It contributes to the process by saving any functional state (if necessary), and

otherwise preparing the function for the transition to D1. As part of this quiescence process the function's software driver must ensure that any mid-transaction TLPs (i.e., requests with outstanding completions), are terminated prior to handing control to the system configuration software that would then complete the transition to D1.

### 2.15.1.3  D2 State

Support for the D2 state is optional in PCI Express and is primarily driven by software. It provides significant power savings while still retaining capability to transition back to a previous condition. In this state, the PCI function can only initiate power management events and it can only respond to configuration accesses.

### 2.15.1.4  D3hot and D3cold State

All PCI Express functions are required to support D3 state. In the D3hot state, power is still present while in D3cold there is no power. Devices in the D3hot state may be transitioned back to the D0 state while the devices in the D3cold state need re-initialization to be brought back to the D0 state. Note that all PCIe devices support D3cold state as it is nothing but a power off state.

Functions in D3hot respond to configuration space accesses as long as power and clock are supplied so that they can be returned to D0 by software. When programmed to D0, the function may return to the D0 Initialized or D0 Un-initialized state without PCIe reset being asserted. There is an option of either performing an internal reset or not performing an internal reset. If not performing an internal reset, upon completion of the D3hot to D0 Initialized state, no additional operating system intervention is required beyond writing the power state bits. If the internal reset is performed, devices return to the D0 Un-initialized and a full re-initialization is performed on the device. The full re-initialization sequence returns the device to D0 Initialized, where normal operation may resume.

## 2.15.2  Link State Power Management

There are multiple states for the physical layer - L0, L0s, L1, L2, and L3 states that provide incrementally higher power savings as the states transition from L0 toward L3. The PCIESS supports the L0, L0s, and L1 states. The L3 state is a power-off state and is supported by default. The PCIESS does not support L2 power down states.

Some of the link power states are autonomously managed by hardware. This scheme is referred to as active state power management (ASPM). Figure 2-7 illustrates the transitions between states L0, L0s, and L1 that are managed by ASPM. The transitions between gray states are allowed in ASPM. These transitions can be enabled through registers in the PCI Express configuration space to allow either just L0s transition or both L0s and L1 transitions.

**Figure 2-7     ASPM Link State Transitions**
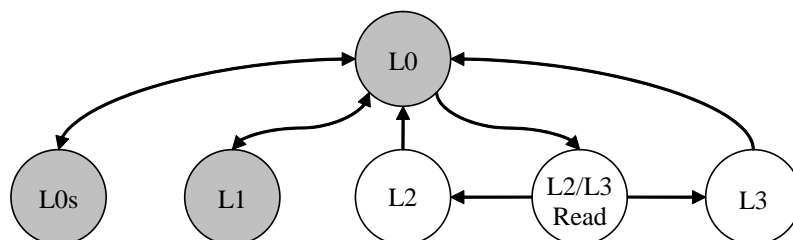
The L1 power state can also be entered via software control. When the device power state is D1, D2, or D3hot, then the link state must transition to L1 state. shows such state transitions.

**Figure 2-8      Software driven Link Power State Transition**
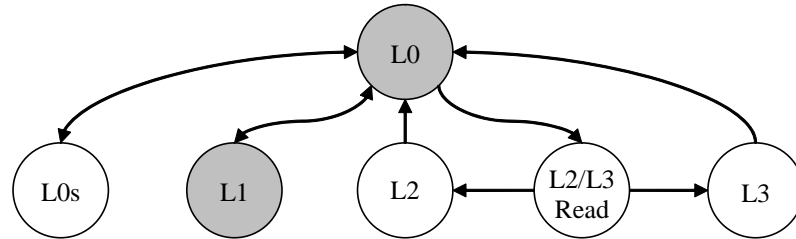


### 2.15.2.1  L0s State

L0s is a lower power state enabled by active state power management (ASPM). Each device can control its L0s transition on its transmitter. Receiver side is controlled by the remote device.

### 2.15.2.2  L1 State

The L1 state is reached either through ASPM timer mechanism or via the software initiated transition to a D state other than the D0 state. In the L1 state, the link is in an idle state and both receiver and transmitter in devices on both ends of a link are able to conserve energy.

### 2.15.2.3  L2/L3 Ready State

This state is a transitional state reached from L1, from which the link must either transition to the L2 or to the L3 state. The L3 state is a full power off state. The L2 state is also a power off state except that the wake signal can be used by end point devices to request power and clock from the system.

### 2.15.2.4  L2 State

The L2 state is appropriate when a device needs to monitor an external event while in deep power down mode. In the L2 state, auxiliary power is supplied and a minimal amount of current is drawn from the power source. Almost all of the logic is without power. To recover, the wake signal is used.

### 2.15.2.5  L3 State

In this state, device is supplied no power from the PCIe fabric. There is no mechanism to communicate in the L3 state. To recover, the system must re-establish power and reference clock followed by a fundamental reset.

### 2.15.3 Relationship Between Device and Link Power States

The device D-states are correlated to the link power states. For each D state, there are specific states that the PCIe link or interconnect can transition to. Table 2-11 below shows the permissible state combinations.

**Table 2-11       Device and Link Power States Combinations**

| Downstream Device State | Permissible Upstream Device State | Permissible Link State |
|---|---|---|
| D0 | D0 | L0 (required), L0s (required), L1 ASPM (optional) |
| D1 | D0 - D1 | L1 |
| D2 | D0 - D2 | L1 |
| D3hot | D0 - D3hot | L1, L2/L3 Ready |
| D3cold | D0 - D3cold | L2aux, L3 |
| **End of Table 2-11** | | |

## 2.16 Error Handling

### 2.16.1 Error Reporting

PCI Express provides three mechanisms for establishing the error reporting policy. These mechanisms are controlled and reported through configuration registers mapped into three distinct regions of configuration space. The various error reporting features are enabled as follows:

- PCI-compatible Registers — this error reporting mechanism provides backward compatibility with existing PCI compatible software and is enabled via the Status and Command Register (STATUS_COMMAND).
- PCI Express Capability Registers — this mechanism is available only to software that has knowledge of PCI Express. This required error reporting is enabled via the PCI Express Device Status and Control Register (DEV_STAT_CTRL) mapped within PCI-compatible configuration space.
- PCI Express Advanced Error Reporting Registers — this mechanism involves registers mapped into the extended configuration address space. PCI Express compatible software enables error reporting for individual errors via the Error Mask Registers.

PCI Express includes two methods of reporting errors:

- Error message transactions — used to report errors to the host

  The conventional PCI reports errors via the PERR# and SERR# signals. But PCI Express eliminates theses error-related signals and error messages have been defined to replace these signals by acting as virtual wires. Furthermore, these messages provide additional information that could not be conveyed directly via the PERR# and SERR# signals. This includes identification of the device that detected the error and an indication of the severity of each error.

- Completion status — used by the completer to report errors to the requester

  PCI Express defines a completion status field within the completion header that enables the transaction completer to report errors back to the requester.

### 2.16.2 Error Detection and Handling

This section defines the required support for detecting and handling PCI Express errors.

#### 2.16.2.1 PCI-Compatible Error Handling

Each PCI Express must map required PCI Express error support to the PCI-related error registers. This involves enabling error reporting and setting status bits that can be read by PCI-compliant software. The PCI Express errors tracked by the PC I compatible registers are:

- Transaction Poisoning
- Completer Abort (CA) detected by a completer
- Unrecognizable Request (UR) detected by a completer

The PCI mechanism for reporting errors is the assertion of PERR# (data parity errors) and SERR# (unrecoverable errors). The PCI Express mechanisms for reporting these events are via the split transaction mechanism (transaction completions) and virtual SERR# signaling via error messages.

The Status and Command Register (STATUS_COMMAND) has error-related bits to understand the features available from the PCI-compatible point of view. While the Status and Command Register bits have the PCI name, some of the field definitions have been modified to reflect the related PCI Express error conditions and reporting mechanisms. Please see the Chapter 3 Registers for the details of this register.

The following bits are set to enable baseline error reporting under control of PCI-compatible software.

- SERR# Enable (bit 8)
- Parity Error Response (bit 6)

The following bits are set under certain error circumstances:

- Detected Parity Error (bit 31)
- Signalled System Error (bit 30)
- Received Master Abort (bit 29)
- Received Target Abort (bit 28)
- Signalled Target Abort (bit 27)
- Master Data Parity Error (bit 24)

### 2.16.2.2  PCI Express Baseline Error Handling

The Device Status and Control Register (DEV_STAT_CTRL) permits software to enable generation of error messages for four error-related events and to check status information to determine which type of error has been detected:

- Correctable Errors
- Non-Fatal Errors
- Fatal Errors
- Unsupported Request Errors

#### 2.16.2.2.1  Enabling Error Reporting

Setting the corresponding bit in the Device Status and Control Register enables the generation of the corresponding error message, which reports errors associated with each classification. Unsupported Request errors are specified as Non-Fatal errors and are reported via a Non-Fatal error message, but only when the Unsupported Request Enable bit is set.

- Unsupported Request Reporting Enable (bit 3)
- Fatal Error Reporting Enable (bit 2)
- Non-Fatal Error Reporting Enable (bit 1)
- Correctable Error Reporting Enable (bit 0)

#### 2.16.2.2.2  Error Status

An error status bit is set any time an error associated with its classification is detected. These bits are set irrespective of the setting of the Error Reporting Enable bits within the Device Status and Control Register. Because Unsupported Request errors are by default considered Non-Fatal errors, when these errors occur both the Non-Fatal error status bit and the Unsupported Request status bit will be set. Note that these bits are cleared by software when writing a one (1) to the bit field.

- Unsupported Request Detected (bit 19)
- Fatal Error Detected (bit 18)

- Non-Fatal Error Detected (bit 17)
- Correctable Error Detected (bit 16)

### 2.16.2.2.3 Link Errors

The physical link connecting two devices may fail causing a variety of errors. Because the link has incurred errors, the error cannot be reported to the host via the failed link. Therefore, link errors must be reported via the upstream port of switches or by the Root Port itself. Also the related fields in the PCI Express Link Status and Control Register (LINK_STAT_CTRL) are valid only in RC (not applicable within EP device). This permits system software to access link-related error registers on the port that is closest to the host.

The Link Status and Control Register includes a RETRAIN_LINK bit (bit 5) that when set forces the RC to retrain the link. If transaction (upon completion of the retraining) can once again traverse the link without errors, the problem will have been solved.

Software can monitor the LINK_TRAINING bit (bit 27) in the Link Status and Control Register to determine when retraining has completed.

### 2.16.2.2.4 Root's Response to Error Message

When a message is received by the RC the action that it takes when reporting the error message to the host system is determined in part by the Root Control and Capabilities Register (ROOT_CTRL_CAP) settings. There are three bit fields that specify whether an error should be reported as a fatal System Error (SERR set enables generation of a system error):

- SERR on Fatal Error Enable (bit 2)
- SERR on Non-Fatal Error Enable (bit 1)
- SERR on Correctable Error Enable (bit 0)

The PME Interrupt Enable bit (bit 3) allows software to enable and disable interrupt generation upon the RC detecting a PME Message transaction.

## 2.16.2.3 PCI Express Advanced Error Reporting

Advanced Error Reporting requires implementation of the Advanced Error Reporting registers. These registers provide several additional error reporting features.

### 2.16.2.3.1 ECRC Generation and Checking

End-to-End CRC (ECRC) generation and checking can be enabled by PCI Express Advanced Capabilities and Control Register (PCIE_ACCR):

- ECRC Check Enable (bit 8)
- ECRC Check Capable (bit 7)
- ECRC Generation Enable (bit 6)
- ECRC Generation Capable (bit 5)

In some cases, multiple uncorrectable errors may be detected prior to software reading and clearing the register. The First Error Pointer field (bit[4:0]) identifies the bit position within the PCI Express Uncorrectable Error Status Register (PCIE_UNCERR) corresponding to the error that occurred first.

> **Note**—When connecting KeyStone PCIe device to another external PCIe device, it is worth checking if the external PCIe device supports the ECRC feature. The ECRC should be disabled by clearing bits ECRC_CHK_EN and ECRC_GEN_EN in the PCIE_ACCR register if the external PCIe device does not support it, especially if the KeyStone PCIe device is configured as RC.

#### 2.16.2.3.2  Advanced Correctable Error Handling

Advanced error reporting provides the ability to pinpoint specific correctable errors. These errors can selectively cause the generation of a Correctable Error Message being sent to the host system:

- Advanced Correctable Error Status

  When a correctable error occurs the corresponding bit within the PCI Express Correctable Error Status Register (PCIE_CERR) is set. These bits are automatically set by hardware and are cleared by software when writing a 1 to the bit position. These bits are set whether or not the error is reported via an error message.

  - Replay Timer Timeout Status (bit 12)
  - REPLAY_NUM Rollover Status (bit 8)
  - Bad DLLP Status (bit 7)
  - Bad TLP Status (bit 6)
  - Receiver Error Status (bit 0)

- Advanced Correctable Error Reporting

  Whether a particular correctable error is reported to the host is specified by the PCI Express Correctable Error Mask Register (PCIE_CERR_MASK). The default state of the mask bits are cleared (0), thereby causing a Correctable Error message to be delivered when any of the correctable errors are detected. Software may choose to set one or more bits to prevent a Correctable Error Message from being sent when the selected error is detected.

  - Replay Timer Timeout Mask (bit 12)
  - REPLAY_NUM Rollover Mask (bit 8)
  - Bad DLLP Mask (bit 7)
  - Bad TLP Mask (bit 6)
  - Receiver Error Mask (bit 0)

### 2.16.2.3.3 Advanced Uncorrectable Error Handling

Advanced error reporting provides the ability to pinpoint which uncorrectable error has occurred. Furthermore software can specify the severity of each error and select which errors will result in an error message being sent to the host system (Root Complex).

- **Advanced Uncorrectable Error Status:** When an uncorrectable error occurs the corresponding bit within the PCI Express Uncorrectable Error Status Register (PCIE_UNCERR) is set. These bits are automatically set by hardware and are cleared by software when writing a 1 to the bit position. These bits are set whether or not the error is reported via an error message.
    - Unsupported Request Error Status (bit 20)
    - ECRC Error Status (bit 19)
    - Malformed TLP Status (bit 18)
    - Receiver Overflow Status (bit 17)
    - Unexpected Completion Status (bit 16)
    - Completer Abort Status (bit 15)
    - Completion Timeout Status (bit 14)
    - Flow Control Protocol Error Status (bit 13)
    - Poisoned TLP Status (bit 12)
    - Data Link Protocol Error Status (bit 4)

- **Advanced Uncorrectable Error Reporting:** Software can mask out specific errors so that they never cause an error message to be generated. The default condition in the PCI Express Uncorrectable Error Mask Register (PCIE_UNCERR_MASK) is to generate error messages for each type of error (all bits are cleared).
    - Unsupported Request Error Mask (bit 20)
    - ECRC Error Mask (bit 19)
    - Malformed TLP Mask (bit 18)
    - Receiver Overflow Mask (bit 17)
    - Unexpected Completion Mask (bit 16)
    - Completer Abort Mask (bit 15)
    - Completion Timeout Mask (bit 14)
    - Flow Control Protocol Error Mask (bit 13)
    - Poisoned TLP Mask (bit 12)
    - Data Link Protocol Error Mask (bit 4)

- **Selecting the Severity of Each Uncorrectable Error:** Advanced error handling permits software to select the severity of each error within the PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY). This gives software the opportunity to treat errors according to the severity associated with a given application. For example, Poisoned TLP data associated with audio data being sent to a speaker, while not correctable has no serious side effects relative

to the integrity of the system. However, if real-time status information is being retrieved that will help make critical decisions, any errors in this data can be very serious. The values in the individual bit fields represent the default severity levels for each type of error (0 = Non-Fatal,1 = Fatal).

– Unsupported Request Error Severity (bit 20)

– ECRC Error Severity (bit 19)

– Malformed TLP Severity (bit 18)

– Receiver Overflow Severity (bit 17)

– Unexpected Completion Severity (bit 16)

– Completer Abort Severity (bit 15)

– Completion Timeout Severity (bit 14)

– Flow Control Protocol Error Severity (bit 13)

– Poisoned TLP Severity (bit 12)

– Data Link Protocol Error Severity (bit 4)

#### 2.16.2.3.4 Error Logging

A four double-word portion (HDR_LOG0~HDR_LOG3) of the Advanced Error Registers block is reserved for storing the header of the transaction that has incurred a failure. Only a select group of transaction layer errors result in the transaction header being logged. Table 2-12 lists the transactions that are logged.

**Table 2-12    Transaction Layer Error That are Logged**

| Name of Error | Default Classification |
| --- | --- |
| Poisoned TLP Received | Uncorrectable - Non Fatal |
| ECRC Check Failed | Uncorrectable - Non Fatal |
| Unsupported Request | Uncorrectable - Non Fatal |
| Completion Abort | Uncorrectable - Non Fatal |
| Unexpected Completion | Uncorrectable - Non Fatal |
| Malformed TLP | Uncorrectable - Fatal |
| **End of Table 2-12** | |

#### 2.16.2.3.5 Root Complex Error Tracking and Reporting

The Root Complex is the target of all error messages issued by devices within the PCI Express fabric. Errors received by the RC result in status registers being updated and the error being conditionally reported to the appropriate software handler or handlers.

- **Root Error Status Registers:** When the Root Complex receives an error message, it sets status bits within the Root Error Status Register (ROOT_ERR_ST). This register indicates the types of errors received and also indicates when multiple errors of the same type have been received.

  – Fatal Error Messages Received (bit 6)

  – Non-Fatal Error Messages Received (bit 5)

  – First Uncorrectable Fatal Received (bit 4)

  – Multiple Uncorrectable Error Received (bit 3)

  – Uncorrectable Error Received (bit 2)

  – Multiple Correctable Error Received (bit 1)

  – Correctable Error Received (bit 0)

- **Root Error Command Register:** The Root Error Status Register sets status bits that determines whether a Correctable, Fatal, or Non-Fatal error has occurred. In conjunction with these status bits the Root Complex can also generate separate interrupts that call handlers for each of the error categories. The Root Error Command register (ROOT_ERR_CMD) enables interrupt generation for all three categories as follows:
  - Fatal Error Reporting Enable (bit 2)
  - Non-Fatal Error Reporting Enable (bit 1)
  - Correctable Error Reporting Enable (bit 0)
- **Error Source ID Register:** Software error handlers may need to read and clear error status registers within the device that detected and reported the error. The error messages contain the ID of the device reporting the error. The Error Source Identification Register (ERR_SRC_ID) captures the error message ID associated with the first Fatal/Non-Fatal and correctable Error message received by the Root Complex.

## 2.17 Emulation Considerations

The emulation suspend interface is not supported on PCIESS.

# Registers

This chapter describes the PCI Express registers. The offset is relative to the associated base address of the module. See the device-specific data manual for the memory address of these registers.

## 3.1 Application Registers

The application registers are accessible in multiple ways depending on the point of origin of such accesses. When accessed from the internal bus slave, these registers are accessed via the 4KB space in Address Space 0. When accessed from the PCIe serial link side, the application registers are mapped to BAR0 of an EP as well as a RC. In addition, a RC can access these registers in a PCIESS EP via the upper 1KB space of the PCIe configuration space. The offsets of the registers do not change irrespective of what the mode of accessing these registers is.

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.1.1 Register Summary

**Table 3-1    PCI Express Application Registers (Part 1 of 3)**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | PID | Peripheral Version and ID register | Section 3.1.2 |
| 4h | CMD_STATUS | Command Status Register | Section 3.1.3 |
| 8h | CFG_SETUP | Configuration Transaction Setup Register | Section 3.1.4 |
| Ch | IOBASE | IO TLP Base Register | Section 3.1.5 |
| 10h | TLPCFG | TLP Attribute Configuration Register | Section 3.1.6 |
| 14h | RSTCMD | Reset Command Register | Section 3.1.7 |
| 20h | PMCMD | Power Management Command Register | Section 3.1.8 |
| 24h | PMCFG | Power Management Configuration Register | Section 3.1.9 |
| 28h | ACT_STATUS | Activity Status Register | Section 3.1.10 |
| 30h | OB_SIZE | Outbound Size Register | Section 3.1.11 |
| 34h | DIAG_CTRL | Diagnostic Control Register | Section 3.1.12 |
| 38h | ENDIAN | Endian Mode Register | Section 3.1.13 |
| 3Ch | PRIORITY | Transaction Priority Register | Section 3.1.14 |
| 50h | IRQ_EOI | End of Interrupt Register | Section 3.1.15 |
| 54h | MSI_IRQ | MSI Interrupt IRQ Register | Section 3.1.16 |
| 64h | EP_IRQ_SET | Endpoint Interrupt Request Set Register | Section 3.1.17 |
| 68h | EP_IRQ_CLR | Endpoint Interrupt Request Clear Register | Section 3.1.18 |
| 6Ch | EP_IRQ_STATUS | Endpoint Interrupt Status Register | Section 3.1.19 |
| 70h | GPR0 | General Purpose 0 Register | Section 3.1.20 |
| 74h | GPR1 | General Purpose 1 Register | Section 3.1.21 |
| 78h | GPR2 | General Purpose 2 Register | Section 3.1.22 |
| 7Ch | GPR3 | General Purpose 3 Register | Section 3.1.23 |
| 100h | MSI0_IRQ_STATUS_RAW | MSI 0 Raw Interrupt Status Register | Section 3.1.24 |
| 104h | MSI0_IRQ_STATUS | MSI 0 Interrupt Enabled Status Register | Section 3.1.25 |
| 108h | MSI0_IRQ_ENABLE_SET | MSI 0 Interrupt Enable Set Register | Section 3.1.26 |
| 10Ch | MSI0_IRQ_ENABLE_CLR | MSI 0 Interrupt Enable Clear Register | Section 3.1.27 |
| 110h | MSI1_IRQ_STATUS_RAW | MSI 1 Raw Interrupt Status Register | Section 3.1.28 |
| 114h | MSI1_IRQ_STATUS | MSI 1 Interrupt Enabled Status Register | Section 3.1.29 |
| 118h | MSI1_IRQ_ENABLE_SET | MSI 1 Interrupt Enable Set Register | Section 3.1.30 |
| 11Ch | MSI1_IRQ_ENABLE_CLR | MSI 1 Interrupt Enable Clear Register | Section 3.1.31 |
| 120h | MSI2_IRQ_STATUS_RAW | MSI 2 Raw Interrupt Status Register | Section 3.1.32 |
| 124h | MSI2_IRQ_STATUS | MSI 2 Interrupt Enabled Status Register | Section 3.1.33 |

**Table 3-1      PCI Express Application Registers (Part 2 of 3)**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 128h | MSI2_IRQ_ENABLE_SET | MSI 2 Interrupt Enable Set Register | Section 3.1.34 |
| 12Ch | MSI2_IRQ_ENABLE_CLR | MSI 2 Interrupt Enable Clear Register | Section 3.1.35 |
| 130h | MSI3_IRQ_STATUS_RAW | MSI 3 Raw Interrupt Status Register | Section 3.1.36 |
| 134h | MSI3_IRQ_STATUS | MSI 3 Interrupt Enabled Status Register | Section 3.1.37 |
| 138h | MSI3_IRQ_ENABLE_SET | MSI 3 Interrupt Enable Set Register | Section 3.1.38 |
| 13Ch | MSI3_IRQ_ENABLE_CLR | MSI 3 Interrupt Enable Clear Register | Section 3.1.39 |
| 140h | MSI4_IRQ_STATUS_RAW | MSI 4 Raw Interrupt Status Register | Section 3.1.40 |
| 144h | MSI4_IRQ_STATUS | MSI 4 Interrupt Enabled Status Register | Section 3.1.41 |
| 148h | MSI4_IRQ_ENABLE_SET | MSI 4 Interrupt Enable Set Register | Section 3.1.42 |
| 14Ch | MSI4_IRQ_ENABLE_CLR | MSI 4 Interrupt Enable Clear Register | Section 3.1.43 |
| 150h | MSI5_IRQ_STATUS_RAW | MSI 5 Raw Interrupt Status Register | Section 3.1.44 |
| 154h | MSI5_IRQ_STATUS | MSI 5 Interrupt Enabled Status Register | Section 3.1.45 |
| 158h | MSI5_IRQ_ENABLE_SET | MSI 5 Interrupt Enable Set Register | Section 3.1.46 |
| 15Ch | MSI5_IRQ_ENABLE_CLR | MSI 5 Interrupt Enable Clear Register | Section 3.1.47 |
| 160h | MSI6_IRQ_STATUS_RAW | MSI 6 Raw Interrupt Status Register | Section 3.1.48 |
| 164h | MSI6_IRQ_STATUS | MSI 6 Interrupt Enabled Status Register | Section 3.1.49 |
| 168h | MSI6_IRQ_ENABLE_SET | MSI 6 Interrupt Enable Set Register | Section 3.1.50 |
| 16Ch | MSI6_IRQ_ENABLE_CLR | MSI 6 Interrupt Enable Clear Register | Section 3.1.51 |
| 170h | MSI7_IRQ_STATUS_RAW | MSI 7 Raw Interrupt Status Register | Section 3.1.52 |
| 174h | MSI7_IRQ_STATUS | MSI 7 Interrupt Enabled Status Register | Section 3.1.53 |
| 178h | MSI7_IRQ_ENABLE_SET | MSI 7 Interrupt Enable Set Register | Section 3.1.54 |
| 17Ch | MSI7_IRQ_ENABLE_CLR | MSI 7 Interrupt Enable Clear Register | Section 3.1.55 |
| 180h | LEGACY_A_IRQ_STATUS_RAW | Raw Interrupt Status Register | Section 3.1.56 |
| 184h | LEGACY_A_IRQ_STATUS | Interrupt Enabled Status Register | Section 3.1.57 |
| 188h | LEGACY_A_IRQ_ENABLE_SET | Interrupt Enable Set Register | Section 3.1.58 |
| 18Ch | LEGACY_A_IRQ_ENABLE_CLR | Interrupt Enable Clear Register | Section 3.1.59 |
| 190h | LEGACY_B_IRQ_STATUS_RAW | Raw Interrupt Status Register | Section 3.1.60 |
| 194h | LEGACY_B_IRQ_STATUS | Interrupt Enabled Status Register | Section 3.1.61 |
| 198h | LEGACY_B_IRQ_ENABLE_SET | Interrupt Enable Set Register | Section 3.1.62 |
| 19Ch | LEGACY_B_IRQ_ENABLE_CLR | Interrupt Enable Clear Register | Section 3.1.63 |
| 1A0h | LEGACY_C_IRQ_STATUS_RAW | Raw Interrupt Status Register | Section 3.1.64 |
| 1A4h | LEGACY_C_IRQ_STATUS | Interrupt Enabled Status Register | Section 3.1.65 |
| 1A8h | LEGACY_C_IRQ_ENABLE_SET | Interrupt Enable Set Register | Section 3.1.66 |
| 1ACh | LEGACY_C_IRQ_ENABLE_CLR | Interrupt Enable Clear Register | Section 3.1.67 |
| 1B0h | LEGACY_D_IRQ_STATUS_RAW | Raw Interrupt Status Register | Section 3.1.68 |
| 1B4h | LEGACY_D_IRQ_STATUS | Interrupt Enabled Status Register | Section 3.1.69 |
| 1B8h | LEGACY_D_IRQ_ENABLE_SET | Interrupt Enable Set Register | Section 3.1.70 |
| 1BCh | LEGACY_D_IRQ_ENABLE_CLR | Interrupt Enable Clear Register | Section 3.1.71 |
| 1C0h | ERR_IRQ_STATUS_RAW | Raw ERR Interrupt Status Register | Section 3.1.72 |
| 1C4h | ERR_IRQ_STATUS | ERR Interrupt Enabled Status Register | Section 3.1.73 |
| 1C8h | ERR_IRQ_ENABLE_SET | ERR Interrupt Enable Set Register | Section 3.1.74 |
| 1CCh | ERR_IRQ_ENABLE_CLR | ERR Interrupt Enable Clear Register | Section 3.1.75 |
| 1D0h | PMRST_IRQ_STATUS_RAW | Power Management and Reset Interrupt Status Register | Section 3.1.76 |
| 1D4h | PMRST_IRQ_STATUS | Power Management and Reset Interrupt Enabled Status Register | Section 3.1.77 |

**Table 3-1          PCI Express Application Registers (Part 3 of 3)**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 1D8h | PMRST_ENABLE_SET | Power Management and Reset Interrupt Enable Set Register | Section 3.1.78 |
| 1DCh | PMRST_ENABLE_CLR | Power Management and Reset Interrupt Enable Clear Register | Section 3.1.79 |
| (200+N*8)h | OB_OFFSET_INDEXn | Outbound Translation Region N Offset Low and Index Register (0x200 + N*8) | Section 3.1.80 |
| (204+N*8)h | OB_OFFSETn_HI | Outbound Translation Region N Offset High Register (0x200 + N*8 + 0x4) | Section 3.1.81 |
| 300h | IB_BAR0 | Inbound Translation Bar Match 0 Register | Section 3.1.82 |
| 304h | IB_START0_LO | Inbound Translation 0 Start Address Low Register | Section 3.1.83 |
| 308h | IB_START0_HI | Inbound Translation 0 Start Address High Register | Section 3.1.84 |
| 30Ch | IB_OFFSET0 | Inbound Translation 0 Address Offset Register | Section 3.1.85 |
| 310h | IB_BAR1 | Inbound Translation Bar Match 1 Register | Section 3.1.86 |
| 314h | IB_START1_LO | Inbound Translation 1 Start Address Low Register | Section 3.1.87 |
| 318h | IB_START1_HI | Inbound Translation 1 Start Address High Register | Section 3.1.88 |
| 31Ch | IB_OFFSET1 | Inbound Translation 1 Address Offset Register | Section 3.1.89 |
| 320h | IB_BAR2 | Inbound Translation Bar Match 2 Register | Section 3.1.90 |
| 324h | IB_START2_LO | Inbound Translation 2 Start Address Low Register | Section 3.1.91 |
| 328h | IB_START2_HI | Inbound Translation 2 Start Address High Register | Section 3.1.92 |
| 32Ch | IB_OFFSET2 | Inbound Translation 2 Address Offset Register | Section 3.1.93 |
| 330h | IB_BAR3 | Inbound Translation Bar Match 3 Register | Section 3.1.94 |
| 334h | IB_START3_LO | Inbound Translation 3 Start Address Low Register | Section 3.1.95 |
| 338h | IB_START3_HI | Inbound Translation 3 Start Address High Register | Section 3.1.96 |
| 33Ch | IB_OFFSET3 | Inbound Translation 3 Address Offset Register | Section 3.1.97 |
| 380h | PCS_CFG0[2] | PCS Configuration Lane 0 | Section 3.1.98 |
| 384h | PCS_CFG1[2] | PCS Configuration Lane 1 | Section 3.1.99 |
| 388h | PCS_STATUS[2] | PCS Status Register | Section 3.1.100 |
| 390h | SERDES_CFG0[2] | SerDes Configuration Lane 0 | Section 3.1.101 |
| 394h | SERDES_CFG1[2] | SerDes Configuration Lane 1 | Section 3.1.102 |
| **End of Table 3-1** | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses.

2. These SerDes related registers are only available in KeyStone I devices.

### 3.1.2  Peripheral Version and ID Register (PID)

The Peripheral Version and ID Register (PID) is shown in Figure 3-1 and described in Table 3-2. The offset is 0h.

**Figure 3-1  Peripheral Version and ID Register (PID)**

| 31 | 30 | 29 | 28 | 27 | | 16 |
|---|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | | |
| R-1 | | R-0 | | R-xxx | | |

| 15 | | 11 | 10 | 8 | 7 | 6 | 5 | | 0 |
|---|---|---|---|---|---|---|---|---|---|
| RTL | | | MAJOR | | CUSTOM | | MINOR | | |
| R-2 | | | R-1 | | R-0 | | R-0 | | |
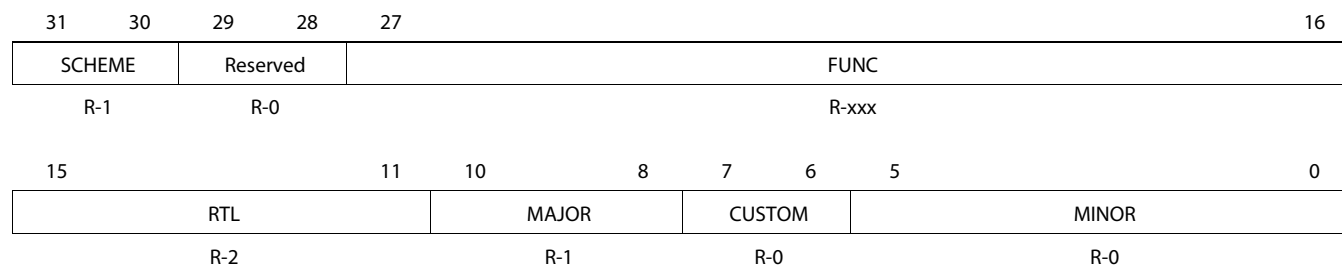
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-2  Peripheral Version and ID Register (PID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-30 | SCHEME | PID Register format scheme. |
| 29-28 | Reserved | Reads return 0 and writes have no effect. |
| 27-16 | FUNC | Function code of the peripheral.<br>E30h for KeyStone I devices<br>E32h for KeyStone II devices |
| 15-11 | RTL | RTL version number. |
| 10-8 | MAJOR | Major revision code. |
| 7-6 | CUSTOM | Custom code. |
| 5-0 | MINOR | Minor revision code. |
| **End of Table 3-2** | | |

### 3.1.3 Command Status Register (CMD_STATUS)

The Command Status Register (CMD_STATUS) is shown in and Figure 3-2 described in Table 3-3. The offset is 4h.

**Figure 3-2     Command Status Register (CMD_STATUS)**

| 31            6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | DBI_CS2 | APP_RETRY_EN | POSTED_WR_EN | IB_XLT_EN | OB_XLT_EN | LTSSM_EN |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-3     Command Status Register (CMD_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | DBI_CS2 | Set to enable writing to BAR mask registers that are overlaid on BAR registers.<br>0 = Disable writing to BAR mask registers.<br>1 = Enable writing to BAR mask registers. |
| 4 | APP_RETRY_EN | Application request retry enable. This feature can be used if initialization can take longer than PCIe stipulated time frame.<br>0 = Disable all incoming PCIe transactions to be returned with a retry response.<br>1 = Enable all incoming PCIe transactions to be returned with a retry response. |
| 3 | POSTED_WR_EN | Posted write enable. Default is 0 with all internal bus master writes defaulting to non-posted. This bit is not applicable for KeyStone device.<br>0 = Disable the internal bus master to use posted write commands.<br>1 = Enable the internal bus master to use posted write commands. |
| 2 | IB_XLT_EN | Inbound address translation enable.<br>0 = Disable translation of inbound memory/IO read/write requests into memory read/write requests.<br>1 = Enable translation of inbound memory/IO read/write requests into memory read/write requests. |
| 1 | OB_XLT_EN | Outbound address translation enable.<br>0 = Disable translation of outbound memory read/write requests into memory/IO/configuration read/write requests.<br>1 = Enable translation of outbound memory read/write requests into memory/IO/configuration read/write requests. |
| 0 | LTSSM_EN | Link training enable.<br>0 = Disable LTSSM in PCI Express core.<br>1 = Enable LTSSM in PCI Express core and link negotiation with link partner will begin. |
| **End of Table 3-3** | | |

### 3.1.4 Configuration Transaction Setup Register (CFG_SETUP)

The Configuration Transaction Setup Register (CFG_SETUP) is shown in Figure 3-3 and described in Table 3-4. The offset is 8h.

**Figure 3-3      Configuration Transaction Setup Register (CFG_SETUP)**

| 31 | 25 | 24 | 23 | 16 |
|----|----|----|----|----|
| Reserved | | CFG_TYPE | CFG_BUS | |
| R-0 | | R/W-0 | R/W-0 | |

| 15 | 13 | 12 | 8 | 7 | 3 | 2 | 0 |
|----|----|----|----|----|----|----|----|
| Reserved | | CFG_DEVICE | | Reserved | | CFG_FUNC | |
| R-0 | | R/W-0 | | R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-4      Configuration Transaction Setup Register (CFG_SETUP) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-25 | Reserved | Reads return 0 and writes have no effect. |
| 24 | CFG_TYPE | Configuration type for outbound configuration accesses.<br> 0 = Type 0 access.<br> 1 = Type 1 access. |
| 23-16 | CFG_BUS | PCIe bus number for outbound configuration accesses (value = 0-FFh). |
| 15-13 | Reserved | Reads return 0 and writes have no effect. |
| 12-8 | CFG_DEVICE | PCIe device number for outbound configuration accesses (value = 0-1Fh). |
| 7-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | CFG_FUNC | PCIe function number for outbound configuration accesses (value = 0-7h). |
| **End of Table 3-4** | | |

### 3.1.5 IO TLP Base Register (IOBASE)

The IO TLP Base Register (IOBASE) is shown in Figure 3-4 and described in Table 3-5. The offset is Ch.

**Figure 3-4      IO TLP Base Register (IOBASE)**

| 31 | 12 | 11 | 0 |
|----|----|----|----|
| IOBASE | | Reserved | |
| R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-5      IO TLP Base Register (IOBASE) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-12 | IOBASE | Bits [31:12] of outgoing IO TLP (value = 0-FFFFFh). RC mode only. |
| 11-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-5** | | |

### 3.1.6  TLP Attribute Configuration Register (TLPCFG)

The TLP Attribute Configuration Register (TLPCFG) is shown in Figure 3-5 and described in Table 3-6. The offset is 10h.

**Figure 3-5     TLP Attribute Configuration Register (TLPCFG)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | RELAXED | NO_SNOOP |
| R-0 | | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-6     TLP Attribute Configuration Register (TLPCFG) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reads return 0 and writes have no effect. |
| 1 | RELAXED | Enable relaxed ordering for all outgoing TLPs.<br>0 = Disable relaxed ordering for all outgoing TLPs.<br>1 = Enable relaxed ordering for all outgoing TLPs. |
| 0 | NO_SNOOP | Enable No Snoop attribute on all outgoing TLPs.<br>0 = Disable no snoop attribute on all outgoing TLPs.<br>1 = Enable no snoop attribute on all outgoing TLPs. |
| **End of Table 3-6** | | |

### 3.1.7  Reset Command Register (RSTCMD)

The Reset Command Register (RSTCMD) is shown in Figure 3-6 and described in Table 3-7. The offset is 14h

**Figure 3-6     Reset Command Register (RSTCMD)**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | FLUSH_N |
| R-0 | | R-1 |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_RST |
| R-0 | | W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-7     Reset Command Register (RSTCMD) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-17 | Reserved | Reads return 0 and writes have no effect. |
| 16 | FLUSH_N | Bridge flush status. Used to ensure no pending transactions prior to issuing warm reset.<br>0 = No transaction is pending.<br>1 = There are transactions pending. |
| 15-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INIT_RST | Write 1 to initiate a downstream hot reset sequence on downstream.<br>0 = No effect.<br>1 = Initiate a downstream hot rest sequence on downstream. |
| **End of Table 3-7** | | |

### 3.1.8 Power Management Command Register (PMCMD)

The Power Management Command Register (PMCMD) is shown in Figure 3-7 and described in Table 3-8. The offset is 20h.

**Figure 3-7    Power Management Command Register (PMCMD)**

| 31 | | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | | PM_XMT_TURNOFF | PM_XMT_PME |
| R-0 | | | W1S-0 | W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-8    Power Management Command Register (PMCMD) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reads return 0 and writes have no effect. |
| 1 | PM_XMT_TURNOFF | Write 1 to transmit a PM_TURNOFF message. Reads 0. Applicable in RC mode only.<br>0 = No effect<br>1 = Transmit a PM_TURNOFF message |
| 0 | PM_XMT_PME | Write 1 to transmit a PM_PME message. Reads 0. Applicable to EP mode only.<br>0 = No effect<br>1 = Transmit a PM_PME message |
| **End of Table 3-8** | | |

### 3.1.9 Power Management Configuration Register (PMCFG)

The Power Management Configuration Register (PMCFG) is shown in Figure 3-8and described in Table 3-9. The offset is 24h.

**Figure 3-8    Power Management Configuration Register (PMCFG)**

| 31 | | 1 | 0 |
|---|---|---|---|
| Reserved | | | ENTR_L23 |
| R-0 | | | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-9    Power Management Configuration Register (PMCFG) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 0 | ENTR_L23 | Write 1 to enable entry to L2/L3 ready state. Read to check L2/L3 entry readiness. Applicable to RC and EP.<br>0 = Disable entry to L2/L3 ready state.<br>1 = Enable entry to L2/L3 ready state. |
| **End of Table 3-9** | | |

### 3.1.10  Activity Status Register (ACT_STATUS)

The Activity Status Register (ACT_STATUS) is shown in Figure 3-9and described in Table 3-10. The offset is 28h.

**Figure 3-9        Activity Status Register (ACT_STATUS)**

| 31 | | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | | OB_NOT_EMPTY | IB_NOT_EMPTY |
| R-0 | | | R-0 | R-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-10        Activity Status Register (ACT_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reads return 0 and writes have no effect. |
| 1 | OB_NOT_EMPTY | 0 = Outbound buffers are empty<br>1 = Outbound buffers are not empty |
| 0 | IB_NOT_EMPTY | 0 = Inbound buffers are empty<br>1 = Inbound buffers are not empty |
| **End of Table 3-10** | | |

### 3.1.11  Outbound Size Register (OB_SIZE)

The Outbound Size Register (OB_SIZE) is shown in Figure 3-10 and described in Table 3-11. The offset is 30h.

**Figure 3-10        Outbound Size Register (OB_SIZE)**

| 31 | | 3 | 2 | 0 |
|---|---|---|---|---|
| Reserved | | | OB_SIZE | |
| R-0 | | | R/W-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-11        Outbound Size Register (OB_SIZE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | OB_SIZE | Set each outbound translation window size. Applicable to RC and EP.<br>0 = 1MB<br>1h = 2MB<br>2h = 4MB<br>3h = 8MB<br>4-7h = Reserved |
| **End of Table 3-11** | | |

### 3.1.12 Diagnostic Control Register (DIAG_CTRL)

The Diagnostic Control Register (DIAG_CTRL) is shown in Figure 3-11 and described in Table 3-12. The offset is 34h.

**Figure 3-11      Diagnostic Control Register (DIAG_CTRL)**

| 31 | | 2 | 1 | 0 |
|---|---|---|---|---|
| | Reserved | | INV_ECRC | INV_LCRC |
| | R-0 | | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-12      Diagnostic Control Register (DIAG_CTRL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reads return 0 and writes have no effect. |
| 1 | INV_ECRC | Write 1 to force inversion of LSB of ECRC for the next one packet. It is self cleared when the ECRC error has been injected on one TLP. |
| 0 | INV_LCRC | Write 1 to force inversion of LSB of LCRC for the next one packet. It is self cleared when the LCRC error has been injected on one TLP. |
| **End of Table 3-12** | | |

### 3.1.13 Endian Mode Register (ENDIAN)

The Endian Mode Register (ENDIAN) is shown in Figure 3-12 and described in Table 3-13. The offset is 38h.

**Figure 3-12      Endian Mode Register (ENDIAN)**

| 31 | | 2 | 1 | 0 |
|---|---|---|---|---|
| | Reserved | | | ENDIAN_MODE |
| | R-0 | | | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-13      Endian Mode Register (ENDIAN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reads return 0 and writes have no effect. |
| 1-0 | ENDIAN_MODE | Endian mode.<br> 0 = Swap on 1-byte<br> 1h = Swap on 2-byte<br> 2h = Swap on 4-byte<br> 3h = Swap on 8-byte |
| **End of Table 3-13** | | |

### 3.1.14 Transaction Priority Register (PRIORITY)

The Transaction Priority Register is shown in Figure 3-13 and described in Table 3-14. The offset is 3Ch.

**Figure 3-13    CBA Transaction Priority Register (PRIORITY)**

| 31 | | | | | | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | MST_PRIV |
| R-0 | | | | | | | R/W-0 |

| 15 | 12 | 11 | 8 | 7 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | MST_PRIVID | | Reserved | | MST_PRIORITY | |
| R-0 | | R-0 | | R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-14    CBA Transaction Priority Register Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-17 | Reserved | Reads return 0 and writes have no effect. |
| 16 | MST_PRIV | Master transaction mode. PCIe transaction needs to be in supervisor mode to access the device registers.<br>    0 = PCIe transaction is in user mode.<br>    1 = PCIe transaction is in supervisor mode. |
| 15-12 | Reserved | Reads return 0 and writes have no effect. |
| 11-8 | MST_PRIVID | Master PRIVID value on master transactions. |
| 7-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | MST_PRIORITY | Priority level for each inbound transaction on the internal bus master port. 0 is the highest priority level and 7h is the lowest priority level. |
| **End of Table 3-14** | | |

### 3.1.15 End of Interrupt Register (IRQ_EOI)

The End of Interrupt Register (IRQ_EOI) is shown in Figure 3-14 and described in Table 3-15. The offset is 50h.

**Figure 3-14    End of Interrupt Register (IRQ_EOI)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | EOI | |
| R-0 | | W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-15    End of Interrupt Register (IRQ_EOI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | EOI | EOI for interrupts. Write the interrupt event number to indicate end-of-interrupt for the interrupt events. Write 0 to mark EOI for INTA, 1 for INTB and so on. Please see the Table 2-10 ''PCIESS Interrupt Events'' for the interrupt event number. |
| **End of Table 3-15** | | |

### 3.1.16 MSI Interrupt IRQ Register (MSI_IRQ)

The MSI Interrupt IRQ Register (MSI_IRQ) is shown in Figure 3-15 and described in Table 3-16. The offset is 54h.

**Figure 3-15     MSI Interrupt IRQ Register (MSI_IRQ)**

| 31 | 0 |
|---|---|
| MSI_IRQ | |
| W-0 | |
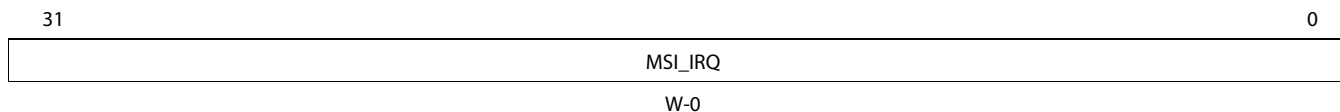
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-16     MSI Interrupt IRQ Register (MSI_IRQ) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | MSI_IRQ | This register is written to by the remote device. Writes initiated by an EP over PCIe link that target BAR0 of the RC land to this register if the offset matches. EP should write MSI vector value to this register to generate corresponding MSI Interrupt, such as writing 0x0 to generate MSI_0 interrupt (with vector 0), writing 0x1F to generate MSI_7 interrupt (with vector 31). |
| **End of Table 3-16** | | |

### 3.1.17 Endpoint Interrupt Request Set Register (EP_IRQ_SET)

The Endpoint Interrupt Request Set Register (EP_IRQ_SET) is shown in Figure 3-16 and described in Table 3-17. The offset is 64h.

**Figure 3-16     Endpoint Interrupt Request Set Register (EP_IRQ_SET)**
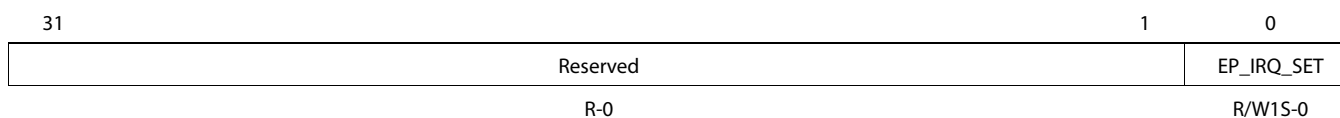
| 31 | 1 | 0 |
|---|---|---|
| Reserved | | EP_IRQ_SET |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-17     Endpoint Interrupt Request Set Register (EP_IRQ_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | EP_IRQ_SET | Write 1 to generate assert interrupt message. If MSI is disabled, legacy interrupt assert message will be generated. On read, a 1 indicates currently asserted interrupt. |
| **End of Table 3-17** | | |

### 3.1.18 Endpoint Interrupt Request Clear Register (EP_IRQ_CLR)

The Endpoint Interrupt Request Clear Register (EP_IRQ_CLR) is shown in Figure 3-17 and described in Table 3-18. The offset is 68h.

**Figure 3-17    Endpoint Interrupt Request Set Register (EP_IRQ_CLR)**

| 31 | 0 |
|---|---|
| Reserved | EP_IRQ_CLR |
| R-0 | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-18    Endpoint Interrupt Request Clear Register (EP_IRQ_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | EP_IRQ_CLR | Write 1 to generate deassert interrupt message. If MSI is disabled, legacy interrupt deassert message will be generated. On read, a 1 indicates currently asserted interrupt. |
| **End of Table 3-18** | | |

### 3.1.19 Endpoint Interrupt Status Register (EP_IRQ_STATUS)

The Endpoint Interrupt Status Register (EP_IRQ_STATUS) is shown in Figure 3-18 and described in Table 3-19. The offset is 6Ch.

**Figure 3-18    Endpoint Interrupt Status Register (EP_IRQ_STATUS)**

| 31 | 0 |
|---|---|
| Reserved | EP_IRQ_STATUS |
| R-0 | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-19    Endpoint Interrupt Status Register (EP_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | EP_IRQ_STATUS | Indicates whether interrupt for function 0 is asserted or not.<br>0 = Interrupt for function 0 is not asserted.<br>1 = Interrupt for function 0 is asserted. |
| **End of Table 3-19** | | |

### 3.1.20  General Purpose 0 Register (GPR0)

The General Purpose 0 Register (GPR0) is shown in Figure 3-19 and described in Table 3-20. The offset is 70h.

**Figure 3-19    General Purpose 0 Register (GPR0)**

| 31 | 0 |
|---|---|
| GENERIC0 | |

R/W-0
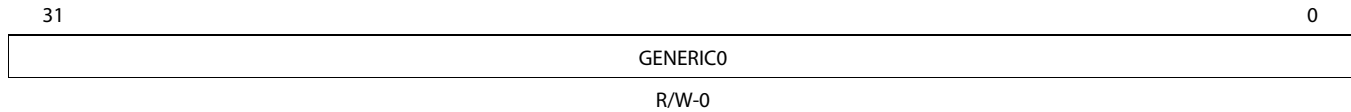
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-20    General Purpose 0 Register (GPR0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | GENERIC0 | Generic info field 0 (value = 0-FFFFFFFFh). |
| **End of Table 3-20** | | |

### 3.1.21  General Purpose 1 Register (GPR1)

The General Purpose 1 Register (GPR1) is shown in Figure 3-20 and described in Table 3-21. The offset is 74h.

**Figure 3-20    General Purpose 1 Register (GPR1)**

| 31 | 0 |
|---|---|
| GENERIC1 | |

R/W-0
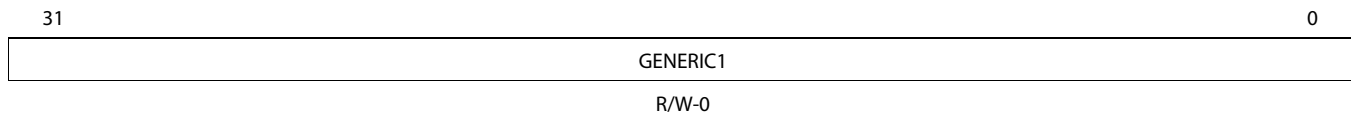
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-21    General Purpose 1 Register (GPR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | GENERIC1 | Generic info field 1 (value = 0-FFFFFFFFh). |
| **End of Table 3-21** | | |

### 3.1.22  General Purpose 2 Register (GPR2)

The General Purpose 2 Register (GPR2) is shown in Figure 3-21 and described in Table 3-22. The offset is 78h.

**Figure 3-21    General Purpose 2 Register (GPR2)**

| 31 | 0 |
|---|---|

| GENERIC2 |
|---|

R/W-0
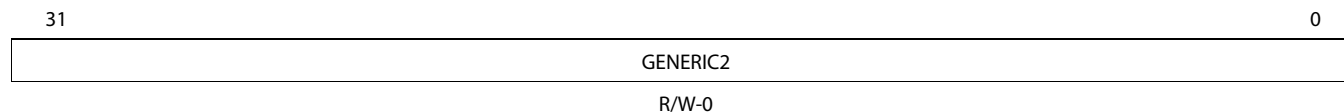
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-22    General Purpose 2 Register (GPR2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | GENERIC2 | Generic info field 2 (value = 0-FFFFFFFFh). |
| **End of Table 3-22** | | |

### 3.1.23  General Purpose 3 Register (GPR3)

The General Purpose 3 Register (GPR3) is shown in Figure 3-22 and described in Table 3-23. The offset is 7Ch.

**Figure 3-22    General Purpose 3 Register (GPR3)**

| 31 | 0 |
|---|---|

| GENERIC3 |
|---|

R/W-0
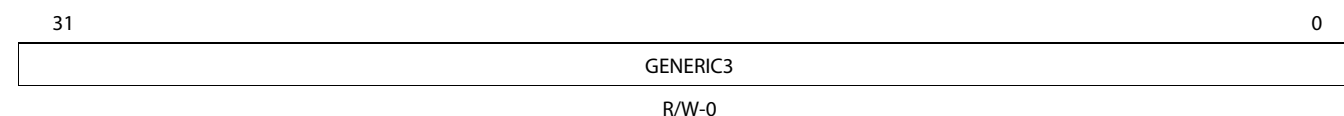
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-23    General Purpose 3 Register (GPR3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | GENERIC3 | Generic info field 3 (value = 0-FFFFFFFFh). |
| **End of Table 3-23** | | |

### 3.1.24  MSI 0 Interrupt Raw Status Register (MSI0_IRQ_STATUS_RAW)

The MSI 0 Interrupt Raw Status Register (MSI0_RAW_STATUS) is shown in Figure 3-23 and described in Table 3-24. The offset is 100h.

**Figure 3-23    MSI 0 Interrupt Raw Status Register (MSI0_IRQ_STATUS_RAW)**
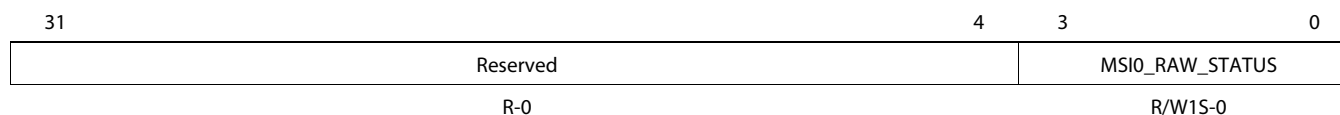
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI0_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-24    MSI 0 Interrupt Raw Status Register (MSI0_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI0_RAW_STATUS | Each bit indicates raw status of MSI vectors (24, 16, 8, 0) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 24 status<br>bit 2 = MSI vector 16 status<br>bit 1 = MSI vector 8 status<br>bit 0 = MSI vector 0 status |
| **End of Table 3-24** | | |

### 3.1.25  MSI 0 Interrupt Enabled Status Register (MSI0_IRQ_STATUS)

The MSI 0 Interrupt Enabled Status Register (MSI0_IRQ_STATUS) is shown in Figure 3-24 and described in Table 3-25. The offset is 104h.

**Figure 3-24    MSI 0 Interrupt Enabled Status Register (MSI0_IRQ_STATUS)**
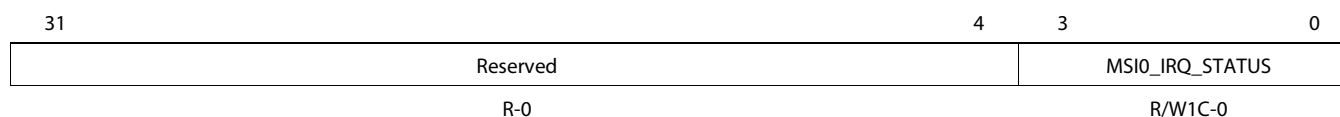
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI0_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-25    MSI 0 Interrupt Enabled Status Register (MSI0_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI0_IRQ_STATUS | Each bit indicates status of MSI vector (24, 16, 8, 0) associated with the bit. Each of the bits can be written with 1 to clear the respective interrupt status bit.<br>bit 3 = MSI vector 24 status<br>bit 2 = MSI vector 16 status<br>bit 1 = MSI vector 8 status<br>bit 0 = MSI vector 0 status |
| **End of Table 3-25** | | |

### 3.1.26 MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_SET)

The MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_SET) is shown in Figure 3-25 and described in Table 3-26. The offset is 108h.

**Figure 3-25    MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI0_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-26    MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI0_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (24, 16, 8, 0) associated with the bit.<br>  bit 3 = MSI vector 24 enable<br>  bit 2 = MSI vector 16 enable<br>  bit 1 = MSI vector 8 enable<br>  bit 0 = MSI vector 0 enable |
| **End of Table 3-264** | | |

### 3.1.27 MSI 0 Interrupt Enable Clear Register (MSI0_IRQ_ENABLE_CLR)

The MSI 0 Interrupt Enable Clear Register (MSI0_IRQ_ENABLE_CLR) is shown in Figure 3-26 and described in Table 3-27. The offset is 10Ch.

**Figure 3-26    MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI0_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-27    MSI 0 Interrupt Enable Set Register (MSI0_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI0_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (24, 16, 8, 0) associated with the bit.<br>  bit 3 = MSI vector 24 disable<br>  bit 2 = MSI vector 16 disable<br>  bit 1 = MSI vector 8 disable<br>  bit 0 = MSI vector 0 disable |
| **End of Table 3-27** | | |

### 3.1.28 MSI 1 Interrupt Raw Status Register (MSI1_IRQ_STATUS_RAW)

The MSI 1 Interrupt Raw Status Register (MSI1_RAW_STATUS) is shown in Figure 3-27 and described in Table 3-28. The offset is 110h.

**Figure 3-27    MSI 1 Interrupt Raw Status Register (MSI1_IRQ_STATUS_RAW)**
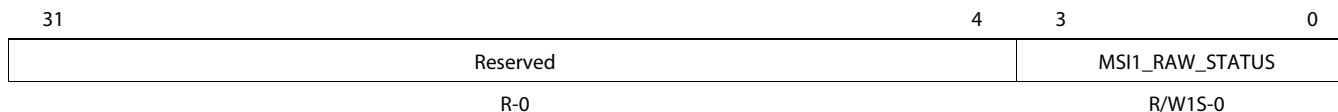
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI1_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-28    MSI 1 Interrupt Raw Status Register (MSI1_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI1_RAW_STATUS | Each bit indicates raw status of MSI vectors (25, 17, 9, 1) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 25 status<br>bit 2 = MSI vector 17 status<br>bit 1 = MSI vector 9 status<br>bit 0 = MSI vector 1 status |
| **End of Table 3-28** | | |

### 3.1.29 MSI 1 Interrupt Enabled Status Register (MSI1_IRQ_STATUS)

The MSI 1 Interrupt Enabled Status Register (MSI1_IRQ_STATUS) is shown in Figure 3-28 and described in Table 3-29. The offset is 114h.

**Figure 3-28    MSI 1 Interrupt Enabled Status Register (MSI1_IRQ_STATUS)**
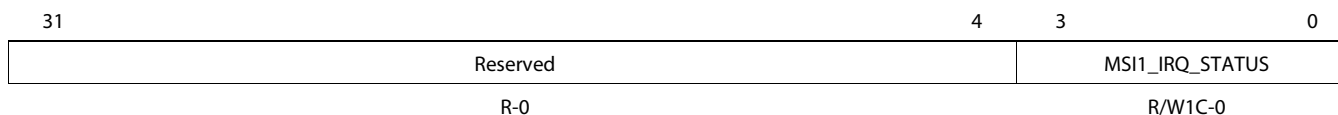
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI1_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-29    MSI 1 Interrupt Enabled Status Register (MSI1_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI1_IRQ_STATUS | Each bit indicates status of MSI vector (25, 17, 9, 1) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 25 status<br>bit 2 = MSI vector 17 status<br>bit 1 = MSI vector 9 status<br>bit 0 = MSI vector 1 status |
| **End of Table 3-29** | | |

### 3.1.30 MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_SET)

The MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_SET) is shown in Figure 3-29 and described in Table 3-30. The offset is 118h.

**Figure 3-29      MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_SET)**

| 31 | | 4 | 3 | | 0 |
|---|---|---|---|---|---|
| Reserved | | | MSI1_IRQ_EN_SET | | |
| R-0 | | | R/W1S-0 | | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-30      MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI1_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (25, 17, 9, 1) associated with the bit.<br>  bit 3 = MSI vector 25 enable<br>  bit 2 = MSI vector 17 enable<br>  bit 1 = MSI vector 9 enable<br>  bit 0 = MSI vector 1 enable |
| **End of Table 3-304** | | |

### 3.1.31 MSI 1 Interrupt Enable Clear Register (MSI1_IRQ_ENABLE_CLR)

The MSI 1 Interrupt Enable Clear Register (MSI1_IRQ_ENABLE_CLR) is shown in Figure 3-30 and described in Table 3-31. The offset is 11Ch.

**Figure 3-30      MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_CLR)**

| 31 | | 4 | 3 | | 0 |
|---|---|---|---|---|---|
| Reserved | | | MSI1_IRQ_EN_CLR | | |
| R-0 | | | R/W1C-0 | | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-31      MSI 1 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI1_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (25, 17, 9, 1) associated with the bit.<br>  bit 3 = MSI vector 25 disable<br>  bit 2 = MSI vector 17 disable<br>  bit 1 = MSI vector 9 disable<br>  bit 0 = MSI vector 1 disable |
| **End of Table 3-31** | | |

### 3.1.32 MSI 2 Interrupt Raw Status Register (MSI2_IRQ_STATUS_RAW)

The MSI 2 Interrupt Raw Status Register (MSI2_RAW_STATUS) is shown in Figure 3-31 and described in Table 3-32. The offset is 120h.

**Figure 3-31      MSI 2 Interrupt Raw Status Register (MSI2_IRQ_STATUS_RAW)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI2_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-32      MSI 2 Interrupt Raw Status Register (MSI2_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI2_RAW_STATUS | Each bit indicates raw status of MSI vectors (26, 18, 10, 2) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 26 status<br>bit 2 = MSI vector 18 status<br>bit 1 = MSI vector 10 status<br>bit 0 = MSI vector 2 status |
| **End of Table 3-32** | | |

### 3.1.33 MSI 2 Interrupt Enabled Status Register (MSI2_IRQ_STATUS)

The MSI 2 Interrupt Enabled Status Register (MSI2_IRQ_STATUS) is shown in Figure 3-32 and described in Table 3-33. The offset is 124h.

**Figure 3-32      MSI 2 Interrupt Enabled Status Register (MSI2_IRQ_STATUS)**
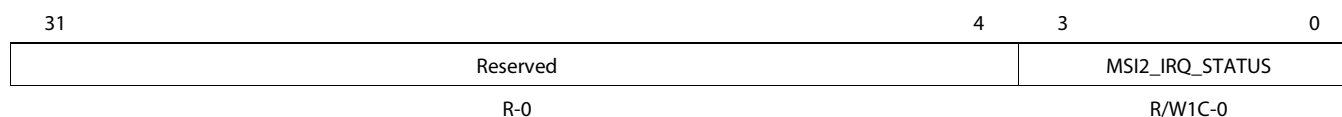
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI2_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-33      MSI 2 Interrupt Enabled Status Register (MSI2_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI2_IRQ_STATUS | Each bit indicates status of MSI vector (26, 18, 10, 2) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 26 status<br>bit 2 = MSI vector 18 status<br>bit 1 = MSI vector 10 status<br>bit 0 = MSI vector 2 status |
| **End of Table 3-33** | | |

### 3.1.34 MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_SET)

The MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_SET) is shown in Figure 3-33 and described in Table 3-34. The offset is 128h.

**Figure 3-33     MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI2_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-34     MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI2_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (26, 18, 10, 2) associated with the bit. <br> bit 3 = MSI vector 26 enable <br> bit 2 = MSI vector 18 enable <br> bit 1 = MSI vector 10 enable <br> bit 0 = MSI vector 2 enable |
| **End of Table 3-344** | | |

### 3.1.35 MSI 2 Interrupt Enable Clear Register (MSI2_IRQ_ENABLE_CLR)

The MSI 2 Interrupt Enable Clear Register (MSI2_IRQ_ENABLE_CLR) is shown in Figure 3-34 and described in Table 3-35. The offset is 12Ch.

**Figure 3-34     MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI2_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-35     MSI 2 Interrupt Enable Set Register (MSI2_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI2_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (26, 18, 10, 2) associated with the bit. <br> bit 3 = MSI vector 26 disable <br> bit 2 = MSI vector 18 disable <br> bit 1 = MSI vector 10 disable <br> bit 0 = MSI vector 2 disable |
| **End of Table 3-35** | | |

### 3.1.36 MSI 3 Interrupt Raw Status Register (MSI3_IRQ_STATUS_RAW)

The MSI 3 Interrupt Raw Status Register (MSI3_RAW_STATUS) is shown in Figure 3-35 and described in Table 3-36. The offset is 130h.

**Figure 3-35     MSI 3 Interrupt Raw Status Register (MSI3_IRQ_STATUS_RAW)**
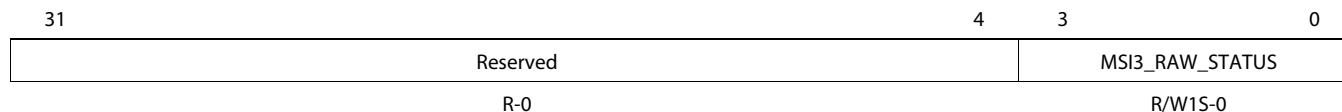
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI3_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-36     MSI 3 Interrupt Raw Status Register (MSI3_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI3_RAW_STATUS | Each bit indicates raw status of MSI vectors (27, 19, 11, 3) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 27 status<br>bit 2 = MSI vector 19 status<br>bit 1 = MSI vector 11 status<br>bit 0 = MSI vector 3 status |
| **End of Table 3-36** | | |

### 3.1.37 MSI 3 Interrupt Enabled Status Register (MSI3_IRQ_STATUS)

The MSI 3 Interrupt Enabled Status Register (MSI3_IRQ_STATUS) is shown in Figure 3-36 and described in Table 3-37. The offset is 134h.

**Figure 3-36     MSI 3 Interrupt Enabled Status Register (MSI3_IRQ_STATUS)**
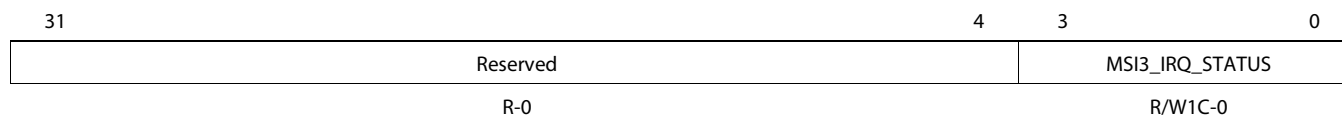
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI3_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-37     MSI 3 Interrupt Enabled Status Register (MSI3_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI3_IRQ_STATUS | Each bit indicates status of MSI vector (27, 19, 11, 3) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 27 status<br>bit 2 = MSI vector 19 status<br>bit 1 = MSI vector 11 status<br>bit 0 = MSI vector 3 status |
| **End of Table 3-37** | | |

### 3.1.38 MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_SET)

The MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_SET) is shown in Figure 3-37 and described in Table 3-38. The offset is 138h.

**Figure 3-37     MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI3_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-38     MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI3_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (27, 19, 11, 3) associated with the bit.<br>bit 3 = MSI vector 27 enable<br>bit 2 = MSI vector 19 enable<br>bit 1 = MSI vector 11 enable<br>bit 0 = MSI vector 3 enable |
| **End of Table 3-384** | | |

### 3.1.39 MSI 3 Interrupt Enable Clear Register (MSI3_IRQ_ENABLE_CLR)

The MSI 3 Interrupt Enable Clear Register (MSI3_IRQ_ENABLE_CLR) is shown in Figure 3-38 and described in Table 3-39. The offset is 13Ch.

**Figure 3-38     MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI3_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-39     MSI 3 Interrupt Enable Set Register (MSI3_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI3_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (27, 19, 11, 3) associated with the bit.<br>bit 3 = MSI vector 27 disable<br>bit 2 = MSI vector 19 disable<br>bit 1 = MSI vector 11 disable<br>bit 0 = MSI vector 3 disable |
| **End of Table 3-39** | | |

### 3.1.40 MSI 4 Interrupt Raw Status Register (MSI4_IRQ_STATUS_RAW)

The MSI 4 Interrupt Raw Status Register (MSI4_RAW_STATUS) is shown in Figure 3-39 and described in Table 3-40. The offset is 140h.

**Figure 3-39      MSI 4 Interrupt Raw Status Register (MSI4_IRQ_STATUS_RAW)**
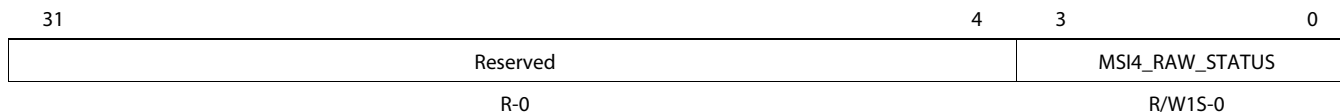
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI4_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-40      MSI 4 Interrupt Raw Status Register (MSI4_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI4_RAW_STATUS | Each bit indicates raw status of MSI vectors (28, 20, 12, 4) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 28 status<br>bit 2 = MSI vector 20 status<br>bit 1 = MSI vector 12 status<br>bit 0 = MSI vector 4 status |
| **End of Table 3-40** | | |

### 3.1.41 MSI 4 Interrupt Enabled Status Register (MSI4_IRQ_STATUS)

The MSI 4 Interrupt Enabled Status Register (MSI4_IRQ_STATUS) is shown in Figure 3-40 and described in Table 3-41. The offset is 144h.

**Figure 3-40      MSI 4 Interrupt Enabled Status Register (MSI4_IRQ_STATUS)**
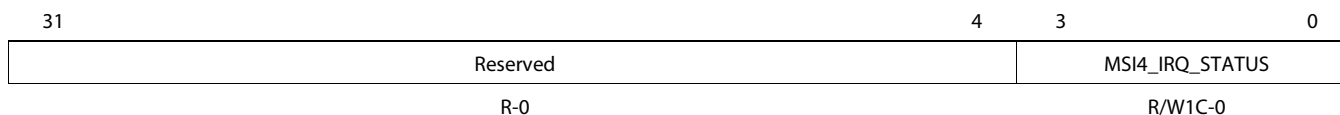
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI4_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-41      MSI 4 Interrupt Enabled Status Register (MSI4_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI4_IRQ_STATUS | Each bit indicates status of MSI vector (28, 20, 12, 4) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 28 status<br>bit 2 = MSI vector 20 status<br>bit 1 = MSI vector 12 status<br>bit 0 = MSI vector 4 status |
| **End of Table 3-41** | | |

### 3.1.42 MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_SET)

The MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_SET) is shown in Figure 3-41 and described in Table 3-42. The offset is 148h.

**Figure 3-41       MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI4_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-42       MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI4_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (28, 20, 12, 4) associated with the bit.<br>bit 3 = MSI vector 28 enable<br>bit 2 = MSI vector 20 enable<br>bit 1 = MSI vector 12 enable<br>bit 0 = MSI vector 4 enable |
| **End of Table 3-424** | | |

### 3.1.43 MSI 4 Interrupt Enable Clear Register (MSI4_IRQ_ENABLE_CLR)

The MSI 4 Interrupt Enable Clear Register (MSI4_IRQ_ENABLE_CLR) is shown in Figure 3-42 and described in Table 3-43. The offset is 14Ch.

**Figure 3-42       MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI4_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-43       MSI 4 Interrupt Enable Set Register (MSI4_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI4_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (28, 20, 12, 4) associated with the bit.<br>bit 3 = MSI vector 28 disable<br>bit 2 = MSI vector 20 disable<br>bit 1 = MSI vector 12 disable<br>bit 0 = MSI vector 4 disable |
| **End of Table 3-43** | | |

### 3.1.44  MSI 5 Interrupt Raw Status Register (MSI5_IRQ_STATUS_RAW)

The MSI 5 Interrupt Raw Status Register (MSI5_RAW_STATUS) is shown in Figure 3-43 and described in Table 3-44. The offset is 150h.

**Figure 3-43     MSI 5 Interrupt Raw Status Register (MSI5_IRQ_STATUS_RAW)**
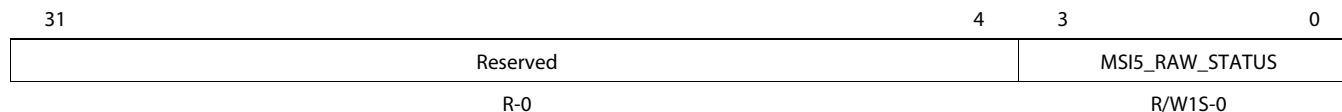
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI5_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-44     MSI 5 Interrupt Raw Status Register (MSI5_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI5_RAW_STATUS | Each bit indicates raw status of MSI vectors (29, 21, 13, 5) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 29 status<br>bit 2 = MSI vector 21 status<br>bit 1 = MSI vector 13 status<br>bit 0 = MSI vector 5 status |
| **End of Table 3-44** | | |

### 3.1.45  MSI 5 Interrupt Enabled Status Register (MSI5_IRQ_STATUS)

The MSI 5 Interrupt Enabled Status Register (MSI5_IRQ_STATUS) is shown in Figure 3-44 and described in Table 3-45. The offset is 154h.

**Figure 3-44     MSI 5 Interrupt Enabled Status Register (MSI5_IRQ_STATUS)**
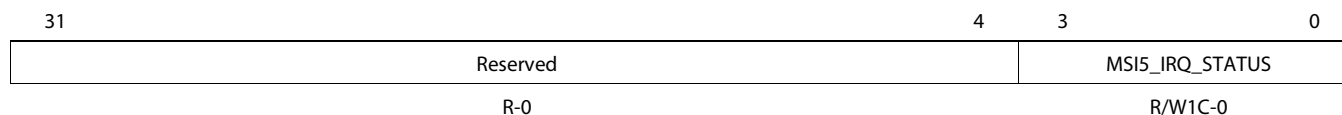
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI5_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-45     MSI 5 Interrupt Enabled Status Register (MSI5_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI5_IRQ_STATUS | Each bit indicates status of MSI vector (29, 21, 13, 5) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 29 status<br>bit 2 = MSI vector 21 status<br>bit 1 = MSI vector 13 status<br>bit 0 = MSI vector 5 status |
| **End of Table 3-45** | | |

### 3.1.46  MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_SET)

The MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_SET) is shown in Figure 3-45 and described in Table 3-46. The offset is 158h.

**Figure 3-45     MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI5_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-46     MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI5_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (29, 21, 13, 5) associated with the bit.<br>  bit 3 = MSI vector 29 enable<br>  bit 2 = MSI vector 21 enable<br>  bit 1 = MSI vector 13 enable<br>  bit 0 = MSI vector 5 enable |
| **End of Table 3-464** | | |

### 3.1.47  MSI 5 Interrupt Enable Clear Register (MSI5_IRQ_ENABLE_CLR)

The MSI 5 Interrupt Enable Clear Register (MSI5_IRQ_ENABLE_CLR) is shown in Figure 3-46 and described in Table 3-47. The offset is 15Ch.

**Figure 3-46     MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI5_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-47     MSI 5 Interrupt Enable Set Register (MSI5_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI5_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (29, 21, 13, 5) associated with the bit.<br>  bit 3 = MSI vector 29 disable<br>  bit 2 = MSI vector 21 disable<br>  bit 1 = MSI vector 13 disable<br>  bit 0 = MSI vector 5 disable |
| **End of Table 3-47** | | |

### 3.1.48 MSI 6 Interrupt Raw Status Register (MSI6_IRQ_STATUS_RAW)

The MSI 6 Interrupt Raw Status Register (MSI6_RAW_STATUS) is shown in Figure 3-47 and described in Table 3-48. The offset is 160h.

**Figure 3-47     MSI 6 Interrupt Raw Status Register (MSI6_IRQ_STATUS_RAW)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI6_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-48     MSI 6 Interrupt Raw Status Register (MSI6_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI6_RAW_STATUS | Each bit indicates raw status of MSI vectors (30, 22, 14, 6) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 30 status<br>bit 2 = MSI vector 22 status<br>bit 1 = MSI vector 14 status<br>bit 0 = MSI vector 6 status |
| **End of Table 3-48** | | |

### 3.1.49 MSI 6 Interrupt Enabled Status Register (MSI6_IRQ_STATUS)

The MSI 6 Interrupt Enabled Status Register (MSI6_IRQ_STATUS) is shown in Figure 3-48 and described in Table 3-49. The offset is 164h.

**Figure 3-48     MSI 6 Interrupt Enabled Status Register (MSI6_IRQ_STATUS)**
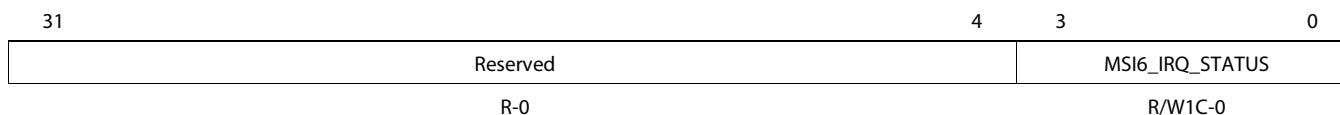
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI6_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-49     MSI 6 Interrupt Enabled Status Register (MSI6_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI6_IRQ_STATUS | Each bit indicates status of MSI vector (30, 22, 14, 6) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 30 status<br>bit 2 = MSI vector 22 status<br>bit 1 = MSI vector 14 status<br>bit 0 = MSI vector 6 status |
| **End of Table 3-49** | | |

### 3.1.50 MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_SET)

The MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_SET) is shown in Figure 3-49 and described in Table 3-50. The offset is 168h.

**Figure 3-49    MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_SET)**

| 31 | | 4 | 3 | 0 |
|---|---|---|---|---|
| | Reserved | | MSI6_IRQ_EN_SET | |
| | R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-50    MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI6_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (30, 22, 14, 6) associated with the bit. <br> bit 3 = MSI vector 30 enable <br> bit 2 = MSI vector 22 enable <br> bit 1 = MSI vector 14 enable <br> bit 0 = MSI vector 6enable |
| **End of Table 3-504** | | |

### 3.1.51 MSI 6 Interrupt Enable Clear Register (MSI6_IRQ_ENABLE_CLR)

The MSI 6 Interrupt Enable Clear Register (MSI6_IRQ_ENABLE_CLR) is shown in Figure 3-50 and described in Table 3-51. The offset is 16Ch.

**Figure 3-50    MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_CLR)**

| 31 | | 4 | 3 | 0 |
|---|---|---|---|---|
| | Reserved | | MSI6_IRQ_EN_CLR | |
| | R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-51    MSI 6 Interrupt Enable Set Register (MSI6_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI6_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (30, 22, 14, 6) associated with the bit. <br> bit 3 = MSI vector 30 disable <br> bit 2 = MSI vector 22 disable <br> bit 1 = MSI vector 14 disable <br> bit 0 = MSI vector 6 disable |
| **End of Table 3-51** | | |

### 3.1.52 MSI 7 Interrupt Raw Status Register (MSI7_IRQ_STATUS_RAW)

The MSI 7 Interrupt Raw Status Register (MSI7_RAW_STATUS) is shown in Figure 3-51 and described in Table 3-52. The offset is 170h.

**Figure 3-51    MSI 7 Interrupt Raw Status Register (MSI7_IRQ_STATUS_RAW)**
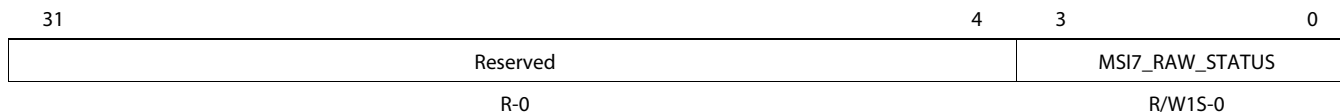
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI7_RAW_STATUS | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-52    MSI 7 Interrupt Raw Status Register (MSI7_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI7_RAW_STATUS | Each bit indicates raw status of MSI vectors (31, 23, 15, 7) associated with the bit. Typically, writes to this register are only done for debug purposes.<br>bit 3 = MSI vector 31 status<br>bit 2 = MSI vector 23 status<br>bit 1 = MSI vector 15 status<br>bit 0 = MSI vector 7status |
| **End of Table 3-52** | | |

### 3.1.53 MSI 7 Interrupt Enabled Status Register (MSI7_IRQ_STATUS)

The MSI 7 Interrupt Enabled Status Register (MSI7_IRQ_STATUS) is shown in Figure 3-52 and described in Table 3-53. The offset is 174h.

**Figure 3-52    MSI 7 Interrupt Enabled Status Register (MSI7_IRQ_STATUS)**
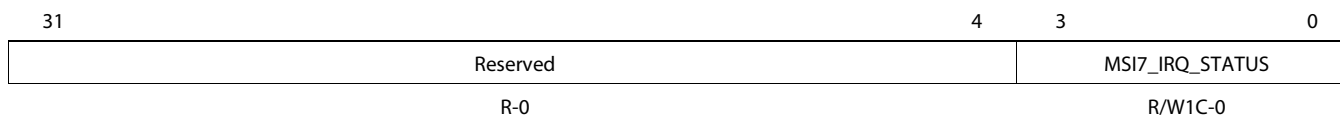
| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI7_IRQ_STATUS | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-53    MSI 7 Interrupt Enabled Status Register (MSI7_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI7_IRQ_STATUS | Each bit indicates status of MSI vector (31, 23, 15, 7) associated with the bit. Each of the bits can be written with one to clear the respective interrupt status bit.<br>bit 3 = MSI vector 31 status<br>bit 2 = MSI vector 23 status<br>bit 1 = MSI vector 15 status<br>bit 0 = MSI vector 7 status |
| **End of Table 3-53** | | |

### 3.1.54 MSI 7 Interrupt Enable Set Register (MSI7_IRQ_ENABLE_SET)

The MSI 7 Interrupt Enable Set Register (MSI7_IRQ_ENABLE_SET) is shown in Figure 3-53 and described in Table 3-54. The offset is 178h.

**Figure 3-53      MSI 7 Interrupt Enable Set Register (MSI7_IRQ_ENABLE_SET)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI7_IRQ_EN_SET | |
| R-0 | | R/W1S-0 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-54      MSI 7 Interrupt Enable Set Register (MSI7_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI7_IRQ_EN_SET | Each bit, when written to, enables the MSI interrupt (31, 23, 15, 7) associated with the bit.<br>  bit 3 = MSI vector 31 enable<br>  bit 2 = MSI vector 23 enable<br>  bit 1 = MSI vector 15 enable<br>  bit 0 = MSI vector 7 enable |
| **End of Table 3-544** | | |

### 3.1.55  MSI7 Interrupt Enable Clear Register (MSI7_IRQ_ENABLE_CLR)

The MSI 7 Interrupt Enable Clear Register (MSI7_IRQ_ENABLE_CLR) is shown in Figure 3-54 and described in Table 3-55. The offset is 17Ch.

**Figure 3-54      MSI 7 Interrupt Enable Set Register (MSI1_IRQ_ENABLE_CLR)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | MSI7_IRQ_EN_CLR | |
| R-0 | | R/W1C-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-55      MSI 7 Interrupt Enable Set Register (MSI7_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3-0 | MSI7_IRQ_EN_CLR | Each bit, when written to, disables the MSI interrupt (31, 23, 15, 7) associated with the bit.<br>  bit 3 = MSI vector 31 disable<br>  bit 2 = MSI vector 23 disable<br>  bit 1 = MSI vector 15 disable<br>  bit 0 = MSI vector 7 disable |
| **End of Table 3-55** | | |

### 3.1.56 LEGACY A Raw Interrupt Status Register (LEGACY_A_IRQ_STATUS_RAW)

The Legacy A Raw Interrupt Status Register (LEGACY_A_IRQ_STATUS_RAW) is shown in Figure 3-55 and described in Table 3-56. The offset is 180h.

**Figure 3-55    Legacy A Raw Interrupt Status Register (LEGACY_A_IRQ_STATUS_RAW)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | Reserved | | INTA |
| | R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-56    Legacy A Raw Interrupt Status Register (LEGACY_A_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTA | Legacy Interrupt A raw status. RC mode only.<br> 0 = interrupt is not active<br> 1 = interrupt is active |
| **End of Table 3-56** | | |

### 3.1.57 LEGACY A Interrupt Enabled Status Register (LEGACY_A_IRQ_STATUS)

The Legacy A Interrupt Enabled Status Register (LEGACY_A_IRQ_STATUS) is shown in Figure 3-56 and described in Table 3-57. The offset is 184h.

**Figure 3-56    Legacy A Interrupt Enabled Status Register (LEGACY_A_IRQ_STATUS)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | Reserved | | INTA |
| | R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-57    Legacy A Interrupt Enabled Status Register (LEGACY_A_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTA | Legacy Interrupt A status. Set when interrupt is active. Write one to clear the interrupt event. RC mode only. |
| **End of Table 3-57** | | |

### 3.1.58 LEGACY A Interrupt Enabled Set Register (LEGACY_A_IRQ_ENABLE_SET)

The Legacy A Interrupt Enabled Set Register (LEGACY_A_IRQ_ENABLE_SET) is shown in Figure 3-57 and described in Table 3-58. The offset is 188h.

**Figure 3-57      Legacy A Interrupt Enabled Set Register (LEGACY_A_IRQ_ENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTA |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-58      Legacy A Interrupt Enabled Set Register (LEGACY_A_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTA | Legacy Interrupt A enable. Set to enable the interrupt. <br><br>On read: <br> 0 = interrupt is disabled <br> 1 = interrupt is enabled |
| **End of Table 3-58** | | |

### 3.1.59 LEGACY A Interrupt Enabled Clear Register (LEGACY_A_IRQ_ENABLE_CLR)

The Legacy A Interrupt Enabled Clear Register (LEGACY_A_IRQ_ENABLE_CLR) is shown in Figure 3-58 and described in Table 3-59. The offset is 18Ch.

**Figure 3-58      Legacy A Interrupt Enabled Clear Register (LEGACY_A_IRQ_ENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTA |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-59      Legacy A Interrupt Enabled Clear Register (LEGACY_A_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTA | Legacy Interrupt A disable. Set to disable the interrupt. <br><br>On read: <br> 0 = interrupt is disabled <br> 1 = interrupt is enabled |
| **End of Table 3-59** | | |

### 3.1.60 LEGACY B Raw Interrupt Status Register (LEGACY_B_IRQ_STATUS_RAW)

The Legacy B Raw Interrupt Status Register (LEGACY_B_IRQ_STATUS_RAW) is shown in Figure 3-59 and described in Table 3-60. The offset is 190h.

**Figure 3-59    Legacy B Raw Interrupt Status Register (LEGACY_B_IRQ_STATUS_RAW)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTB |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-60    Legacy B Raw Interrupt Status Register (LEGACY_B_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTB | Legacy Interrupt B raw status. RC mode only.<br>  0 = interrupt is not active<br>  1 = interrupt is active |
| **End of Table 3-60** | | |

### 3.1.61 LEGACY B Interrupt Enabled Status Register (LEGACY_B_IRQ_STATUS)

The Legacy B Interrupt Enabled Status Register (LEGACY_B_IRQ_STATUS) is shown in Figure 3-60 and described in Table 3-61. The offset is 194h.

**Figure 3-60    Legacy B Interrupt Enabled Status Register (LEGACY_B_IRQ_STATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTB |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-61    Legacy B Interrupt Enabled Status Register (LEGACY_B_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTB | Legacy Interrupt B status. Set when interrupt is active. Write one to clear the interrupt event. RC mode only. |
| **End of Table 3-61** | | |

### 3.1.62  LEGACY B Interrupt Enabled Set Register (LEGACY_B_IRQ_ENABLE_SET)

The Legacy B Interrupt Enabled Set Register (LEGACY_B_IRQ_ENABLE_SET) is shown in Figure 3-61 and described in Table 3-62. The offset is 198h.

**Figure 3-61**     **Legacy B Interrupt Enabled Set Register (LEGACY_B_IRQ_ENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTB |
| R-0 | | R/W1S-0 |

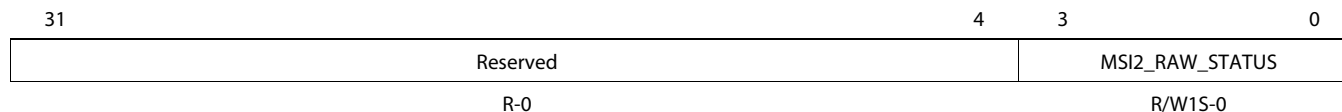Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 3-62**     **Legacy B Interrupt Enabled Set Register (LEGACY_B_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTB | Legacy Interrupt B enable. Set to enable the interrupt. <br><br> On read: <br>  0 = interrupt is disabled <br>  1 = interrupt is enabled |
| **End of Table 3-62** | | |

### 3.1.63  LEGACY B Interrupt Enabled Clear Register (LEGACY_B_IRQ_ENABLE_CLR)

The Legacy B Interrupt Enabled Clear Register (LEGACY_B_IRQ_ENABLE_CLR) is shown in Figure 3-62 and described in Table 3-63. The offset is 19Ch.

**Figure 3-62**     **Legacy B Interrupt Enabled Clear Register (LEGACY_B_IRQ_ENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTB |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-63**     **Legacy B Interrupt Enabled Clear Register (LEGACY_B_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTB | Legacy Interrupt B disable. Set to disable the interrupt. <br><br> On read: <br>  0 = interrupt is disabled <br>  1 = interrupt is enabled |
| **End of Table 3-63** | | |

### 3.1.64 LEGACY C Raw Interrupt Status Register (LEGACY_C_IRQ_STATUS_RAW)

The Legacy C Raw Interrupt Status Register (LEGACY_C_IRQ_STATUS_RAW) is shown in Figure 3-63 and described in Table 3-64. The offset is 1A0h.

**Figure 3-63      Legacy C Raw Interrupt Status Register (LEGACY_C_IRQ_STATUS_RAW)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | Reserved | | INTC |
| | R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-64      Legacy C Raw Interrupt Status Register (LEGACY_C_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTC | Legacy Interrupt C raw status. RC mode only.<br> 0 = interrupt is not active<br> 1 = interrupt is active |
| **End of Table 3-64** | | |

### 3.1.65 LEGACY C Interrupt Enabled Status Register (LEGACY_C_IRQ_STATUS)

The Legacy C Interrupt Enabled Status Register (LEGACY_C_IRQ_STATUS) is shown in Figure 3-64 and described in Table 3-65. The offset is 1A4h.

**Figure 3-64      Legacy C Interrupt Enabled Status Register (LEGACY_C_IRQ_STATUS)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | Reserved | | INTC |
| | R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-65      Legacy C Interrupt Enabled Status Register (LEGACY_C_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTC | Legacy Interrupt C status. Set when interrupt is active. Write one to clear the interrupt event. RC mode only. |
| **End of Table 3-65** | | |

### 3.1.66 LEGACY C Interrupt Enabled Set Register (LEGACY_C_IRQ_ENABLE_SET)

The Legacy C Interrupt Enabled Set Register (LEGACY_C_IRQ_ENABLE_SET) is shown in Figure 3-65 and described in Table 3-66. The offset is 1A8h.

**Figure 3-65      Legacy C Interrupt Enabled Set Register (LEGACY_C_IRQ_ENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTC |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-66      Legacy C Interrupt Enabled Set Register (LEGACY_C_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTC | Legacy Interrupt C enable. Set to enable the interrupt.<br><br>On read:<br> 0 = interrupt is disabled<br> 1 = interrupt is enabled |
| **End of Table 3-66** | | |

### 3.1.67 LEGACY C Interrupt Enabled Clear Register (LEGACY_C_IRQ_ENABLE_CLR)

The Legacy C Interrupt Enabled Clear Register (LEGACY_C_IRQ_ENABLE_CLR) is shown in Figure 3-66 and described in Table 3-67. The offset is 1ACh.

**Figure 3-66      Legacy C Interrupt Enabled Clear Register (LEGACY_C_IRQ_ENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTC |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-67      Legacy C Interrupt Enabled Clear Register (LEGACY_C_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTC | Legacy Interrupt C disable. Set to disable the interrupt.<br><br>On read:<br> 0 = interrupt is disabled<br> 1 = interrupt is enabled |
| **End of Table 3-67** | | |

### 3.1.68  LEGACY D Raw Interrupt Status Register (LEGACY_D_IRQ_STATUS_RAW)

The Legacy D Raw Interrupt Status Register (LEGACY_D_IRQ_STATUS_RAW) is shown in Figure 3-67 and described in Table 3-68. The offset is 1B0h.

**Figure 3-67    Legacy D Raw Interrupt Status Register (LEGACY_D_IRQ_STATUS_RAW)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTD |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-68    Legacy D Raw Interrupt Status Register (LEGACY_D_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTD | Legacy Interrupt D raw status. RC mode only.<br> 0 = interrupt is not active<br> 1 = interrupt is active |
| **End of Table 3-68** | | |

### 3.1.69  LEGACY D Interrupt Enabled Status Register (LEGACY_D_IRQ_STATUS)

The Legacy D Interrupt Enabled Status Register (LEGACY_D_IRQ_STATUS) is shown in Figure 3-68 and described in Table 3-69. The offset is 1B4h.

**Figure 3-68    Legacy D Interrupt Enabled Status Register (LEGACY_D_IRQ_STATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTD |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-69    Legacy D Interrupt Enabled Status Register (LEGACY_D_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTD | Legacy Interrupt D status. Set when interrupt is active. Write one to clear the interrupt event. RC mode only. |
| **End of Table 3-69** | | |

### 3.1.70  LEGACY D Interrupt Enabled Set Register (LEGACY_D_IRQ_ENABLE_SET)

The Legacy D Interrupt Enabled Set Register (LEGACY_D_IRQ_ENABLE_SET) is shown in Figure 3-69 and described in Table 3-70. The offset is 1B8h.

**Figure 3-69      Legacy D Interrupt Enabled Set Register (LEGACY_D_IRQ_ENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTD |
| R-0 | | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-70      Legacy D Interrupt Enabled Set Register (LEGACY_D_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTD | Legacy Interrupt D enable. Set to enable the interrupt.<br>On read:<br> 0 = interrupt is disabled<br> 1 = interrupt is enabled |

**End of Table 3-70**

### 3.1.71  LEGACY D Interrupt Enabled Clear Register (LEGACY_D_IRQ_ENABLE_CLR)

The Legacy D Interrupt Enabled Clear Register (LEGACY_D_IRQ_ENABLE_CLR) is shown in Figure 3-70 and described in Table 3-71. The offset is 1BCh.

**Figure 3-70      Legacy D Interrupt Enabled Clear Register (LEGACY_D_IRQ_ENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTD |
| R-0 | | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; *-n* = value after reset

**Table 3-71      Legacy D Interrupt Enabled Clear Register (LEGACY_D_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | INTD | Legacy Interrupt D disable. Set to disable the interrupt.<br>On read:<br> 0 = interrupt is disabled<br> 1 = interrupt is enabled |

**End of Table 3-71**

### 3.1.72 Raw ERR Interrupt Status Register (ERR_IRQ_STATUS_RAW)

The Raw ERR Interrupt Status Register (ERR_IRQ_STATUS_RAW) is shown in Figure 3-71 and described in Table 3-72.The offset is 1C0h.

**Figure 3-71    Raw ERR Interrupt Status Register (ERR_IRQ_STATUS_RAW)**

| 31 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-72    Raw ERR Interrupt Status Register (ERR_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | ERR_AER | ECRC error raw status. |
| 4 | ERR_AXI | AXI tag lookup fatal error raw status. |
| 3 | ERR_CORR | Correctable error raw status. |
| 2 | ERR_NONFATAL | Nonfatal error raw status. |
| 1 | ERR_FATAL | Fatal error raw status. |
| 0 | ERR_SYS | System error (fatal, nonfatal or correctable error) raw status. |
| **End of Table 3-72** | | |

### 3.1.73 ERR Interrupt Enabled Status Register (ERR_IRQ_STATUS)

The ERR Interrupt Enabled Status Register (ERR_IRQ_STATUS) is shown in Figure 3-72 and described in Table 3-73. The offset is 1C4h.

**Figure 3-72    ERR Interrupt Enabled Status Register (ERR_IRQ_STATUS)**

| 31 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-73    ERR Interrupt Enabled Status Register (ERR_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | ERR_AER | ECRC error status. |
| 4 | ERR_AXI | AXI tag lookup fatal error. |
| 3 | ERR_CORR | Correctable error status. |
| 2 | ERR_NONFATAL | Nonfatal error status. |
| 1 | ERR_FATAL | Fatal error status. |
| 0 | ERR_SYS | System Error (fatal, nonfatal or correctable error). |
| **End of Table 3-73** | | |

### 3.1.74 ERR Interrupt Enable Set Register (ERR_IRQ_ENABLE_SET)

The ERR Interrupt Enable Set Register (ERR_IRQ_ENABLE_SET) is shown in Figure 3-73 and described in Table 3-74. The offset is 1C8h.

**Figure 3-73      ERR Interrupt Enable Set Register (ERR_IRQ_ENABLE_SET)**

| 31 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; *-n* = value after reset

**Table 3-74      ERR Interrupt Enable Set Register (ERR_IRQ_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | ERR_AER | ECRC error interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 4 | ERR_AXI | AXI tag lookup fatal error interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 3 | ERR_CORR | Correctable error interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 2 | ERR_NONFATAL | Nonfatal error interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 1 | ERR_FATAL | Fatal error interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 0 | ERR_SYS | System Error (fatal, nonfatal or correctable error) interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| **End of Table 3-74** | | |

### 3.1.75 ERR Interrupt Enable Clear Register (ERR_IRQ_ENABLE_CLR)

The ERR Interrupt Enable Clear Register (ERR_IRQ_ENABLE_CLR) is shown in Figure 3-74 and described in Table 3-75. The offset is 1CCh.

**Figure 3-74      ERR Interrupt Enable Clear Register (ERR_IRQ_ENABLE_CLR)**

| 31 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

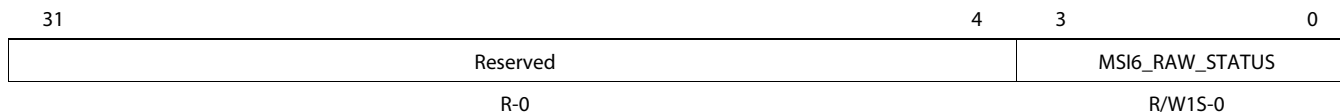Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-75      ERR Interrupt Enable Clear Register (ERR_IRQ_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | ERR_AER | ECRC error interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 4 | ERR_AXI | AXI tag lookup fatal error interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 3 | ERR_CORR | Correctable error interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 2 | ERR_NONFATAL | Nonfatal error interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 1 | ERR_FATAL | Fatal error interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 0 | ERR_SYS | System error (fatal, nonfatal or correctable error) interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| **End of Table 3-75** | | |

### 3.1.76 Power Management and Reset Interrupt Status Register (PMRST_IRQ_STATUS_RAW)

The Power Management and Reset Interrupt Status Register (PMRST_IRQ_STATUS_RAW) is shown in Figure 3-75 and described in Table 3-76. The offset is 1D0h.

**Figure 3-75    Power Management and Reset Interrupt Status Register (PMRST_IRQ_STATUS_RAW)**

| 31 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-76    Power Management and Reset Interrupt Status Register (PMRST_IRQ_STATUS_RAW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | LINK_RST_REQ | Link Request Reset interrupt raw status. |
| 2 | PM_PME | Power management PME message received interrupt raw status. |
| 1 | PM_TO_ACK | Power management ACK received interrupt raw status. |
| 0 | PM_TURNOFF | Power management turnoff message received raw status. |
| **End of Table 3-76** | | |

### 3.1.77 Power Management and Reset Interrupt Enabled Status Register (PMRST_IRQ_STATUS)

The Power Management and Reset Interrupt Enabled Status Register (PMRST_IRQ_STATUS) is shown in Figure 3-76 and described in Table 3-77. The offset is 1D4h.

**Figure 3-76    Power Management and Reset Interrupt Enabled Status Register (PMRST_IRQ_STATUS)**

| 31 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-77    Power Management and Reset Interrupt Enabled Status Register (PMRST_IRQ_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | LINK_RST_REQ | Link request reset interrupt status. |
| 2 | PM_PME | Power management PME message received interrupt status. |
| 1 | PM_TO_ACK | Power management ACK received interrupt status. |
| 0 | PM_TURNOFF | Power management turnoff message received status. |
| **End of Table 3-77** | | |

### 3.1.78 Power Management and Reset Interrupt Enable Set Register (PMRST_ENABLE_SET)

The Power Management and Reset Interrupt Enable Set Register (PMRST_ENABLE_SET) is shown in Figure 3-77 and described in Table 3-78. The offset is 1D8h.

**Figure 3-77     Power Management and Reset Interrupt Enabled Set Register (PMRST_ENABLE_SET)**

| 31 | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-78     Power Management and Reset Interrupt Enable Set Register (PMRST_ENABLE_SET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | LINK_RST_REQ | Link request reset interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 2 | PM_PME | Power management PME message received interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 1 | PM_TO_ACK | Power management ACK received interrupt enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 0 | PM_TURNOFF | Power management turnoff message received enable. Set to enable the interrupt.<br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| **End of Table 3-78** | | |

### 3.1.79  Power Management and Reset Interrupt Enable Clear Register (PMRST_ENABLE_CLR)

The Power Management and Reset Interrupt Enable Clear Register (PMRST_ENABLE_CLR) is shown in Figure 3-78 and described in Table 3-79. The offset is 1DCh.

**Figure 3-78     Power Management and Reset Interrupt Enabled Clear Register (PMRST_ENABLE_CLR)**

| 31 | | | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | | | | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-79     Power Management and Reset Interrupt Enable Clear Register (PMRST_ENABLE_CLR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | LINK_RST_REQ | Link Request Reset interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 2 | PM_PME | Power management PME message received interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 1 | PM_TO_ACK | Power management ACK received interrupt disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| 0 | PM_TURNOFF | Power management Turnoff message received disable. Set to disable the interrupt.<br><br>On read:<br>  0 = interrupt is disabled<br>  1 = interrupt is enabled |
| **End of Table 3-79** | | |

### 3.1.80  Outbound Translation Region N Offset Low and Index Register (OB_OFFSET_INDEXn)

The Outbound Translation Region N Offset Low and Index Register (OB_OFFSET_INDEXn) is shown in Figure 3-79 and described in Table 3-80. The offset is $(200 + N \times 8)$h.

**Figure 3-79    Outbound Translation Region N Offset Low and Index Register (OB_OFFSET_INDEXn)**

| 31 | 20 | 19 | 1 | 0 |
|---|---|---|---|---|
| OB_OFFSETn_LO | | Reserved | | OB_ENABLEn |
| R/W-0 | | R-0 | | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-80    Outbound Translation Region N Offset Low and Index Register (OB_OFFSET_INDEXn)**

| Bit | Field | Description |
|---|---|---|
| 31-20 | OB_OFFSETn_LO | Offset bits [31:20] for translation region N (N=0 to 31). |
| 19-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | OB_ENABLEn | Enable translation region N (N=0 to 31).<br>  0 = outbound translation is disabled<br>  1 = outbound translation is enabled |
| **End of Table 3-80** | | |

### 3.1.81  Outbound Translation Region N Offset High Register (OB_OFFSETn_HI)

The Outbound Translation Region N Offset High Register (OB_OFFSETn_HI) is shown in Figure 3-80 and described in Table 3-81. The offset is $(204+N*8)$h.

**Figure 3-80    Outbound Translation Region N Offset High Register (OB_OFFSETn_HI)**

| 31 | 0 |
|---|---|
| OB_OFFSETn_HI | |
| R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-81    Outbound Translation Region N Offset High Register (OB_OFFSETn_HI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | OB_OFFSETn_HI | Offset bits [63:32] for translation region N (N=0 to 31). |
| **End of Table 3-81** | | |

### 3.1.82 Inbound Translation Bar Match 0 Register (IB_BAR0)

The Inbound Translation Bar Match 0 Register (IB_BAR0) is shown in Figure 3-81 and described in Table 3-82. The offset is 300h.

**Figure 3-81      Inbound Translation Bar Match 0 Register (IB_BAR0)**

| 31 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | IB_BAR0 | |
| R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-82      Inbound Translation Bar Match 0 Register (IB_BAR0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | IB_BAR0 | BAR number to match for inbound translation region 0. |
| **End of Table 3-82** | | |

### 3.1.83 Inbound Translation 0 Start Address Low Register (IB_START0_LO)

The Inbound Translation 0 Start Address Low Register (IB_START0_LO) is shown in Figure 3-82 and described in Table 3-83. The offset is 304h.

**Figure 3-82      Inbound Translation 0 Start Address Low Register (IB_START0_LO)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_START0_LO | | Reserved | |
| R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-83      Inbound Translation 0 Start Address Low Register (IB_START0_LO) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_START0_LO | Start address bits [31:8] for inbound translation region 0. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-83** | | |

### 3.1.84 Inbound Translation 0 Start Address High Register (IB_START0_HI)

The Inbound Translation 0 Start Address High Register (IB_START0_HI) is shown in Figure 3-83 and described in Table 3-84. The offset is 308h.

**Figure 3-83    Inbound Translation 0 Start Address High Register (IB_START0_HI)**

| 31 | 0 |
|---|---|
| IB_START0_HI | |
| R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-84    Inbound Translation 0 Start Address High Register (IB_START0_HI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | IB_START0_HI | Start address bits [63:32] for inbound translation region 0. |
| **End of Table 3-84** | | |

### 3.1.85 Inbound Translation 0 Address Offset Register (IB_OFFSET0)

The Inbound Translation 0 Address Offset Register (IB_OFFSET0) is shown in Figure 3-84 and described in Table 3-85. The offset is 30Ch.

**Figure 3-84    Inbound Translation 0 Address Offset Register (IB_OFFSET0)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_OFFSET0 | | Reserved | |
| R/W-0 | | R-0 | |
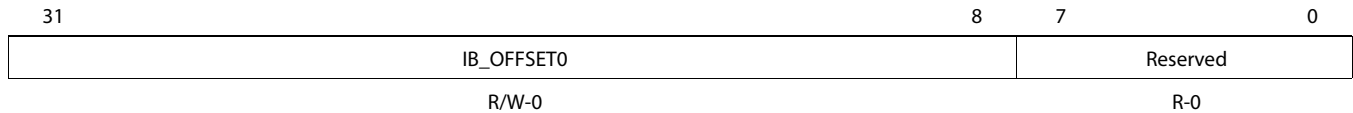
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-85    Inbound Translation 0 Address Offset Register (IB_OFFSET0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_OFFSET0 | Offset address bits [31:8] for inbound translation region 0. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-85** | | |

### 3.1.86  Inbound Translation Bar Match 1 Register (IB_BAR1)

The Inbound Translation Bar Match 1 Register (IB_BAR1)is shown in Figure 3-85 and described in Table 3-86. The offset is 310h.

**Figure 3-85      Inbound Translation Bar Match 1 Register (IB_BAR1)**

| 31 | | 3 | 2 | 0 |
|---|---|---|---|---|
| | Reserved | | | IB_BAR1 |
| | R-0 | | | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-86      Inbound Translation Bar Match 1 Register (IB_BAR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | IB_BAR1 | BAR number to match for inbound translation region 1. |
| **End of Table 3-86** | | |

### 3.1.87  Inbound Translation 1 Start Address Low Register (IB_START1_LO)

The Inbound Translation 1 Start Address Low Register (IB_START1_LO) is shown in Figure 3-86 and described in Table 3-87. The offset is 314h.

**Figure 3-86      Inbound Translation 1 Start Address Low Register (IB_START1_LO)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_START1_LO | | Reserved | |
| R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-87      Inbound Translation 1 Start Address Low Register (IB_START1_LO) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_START1_LO | Start address bits [31:8] for inbound translation region 1. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-87** | | |

### 3.1.88  Inbound Translation 1 Start Address High Register (IB_START1_HI)

The Inbound Translation 1 Start Address High Register (IB_START1_HI) is shown in Figure 3-87 and described in Table 3-88. The offset is 318h.

**Figure 3-87       Inbound Translation 1 Start Address High Register (IB_START1_HI)**

| 31 | 0 |
|---|---|
| IB_START1_HI | |

R/W-0
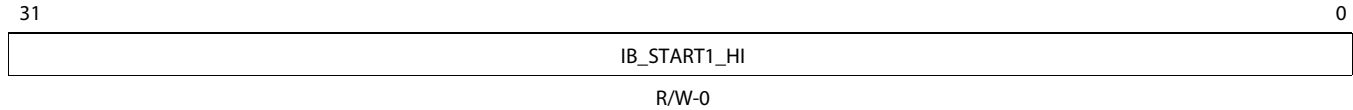
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-88       Inbound Translation 1 Start Address High Register (IB_START1_HI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | IB_START1_HI | Start address bits [63:32] for inbound translation region 1. |
| **End of Table 3-88** | | |

### 3.1.89  Inbound Translation 1 Address Offset Register (IB_OFFSET1)

The Inbound Translation 1 Address Offset Register (IB_OFFSET1) is shown in Figure 3-88 and described in Table 3-89. The offset is 31Ch.

**Figure 3-88       Inbound Translation 1 Address Offset Register (IB_OFFSET1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_OFFSET1 | | Reserved | |

R/W-0                                                                                       R-0
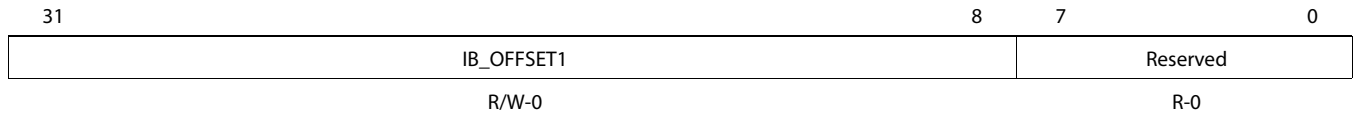
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-89       Inbound Translation 1 Address Offset Register (IB_OFFSET1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_OFFSET1 | Offset address bits [31:8] for inbound translation region 1. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-89** | | |

### 3.1.90  Inbound Translation Bar Match 2 Register (IB_BAR2)

The Inbound Translation Bar Match 2 Register (IB_BAR2) is shown in Figure 3-89 and described in Table 3-90. The offset is 320h.

**Figure 3-89   Inbound Translation Bar Match 2 Register (IB_BAR2)**

| 31 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | IB_BAR2 | |
| R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-90   Inbound Translation Bar Match 2 Register (IB_BAR2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | IB_BAR2 | BAR number to match for inbound translation region 2. |
| **End of Table 3-90** | | |

### 3.1.91  Inbound Translation 2 Start Address Low Register (IB_START2_LO)

The Inbound Translation 2 Start Address Low Register (IB_START2_LO) is shown in Figure 3-90 and described in Table 3-91. The offset is 324h.

**Figure 3-90   Inbound Translation 2 Start Address Low Register (IB_START2_LO)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_START2_LO | | Reserved | |
| R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-91   Inbound Translation 2 Start Address Low Register (IB_START2_LO) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_START2_LO | Start address bits [31:8] for inbound translation region 2. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-91** | | |

### 3.1.92  Inbound Translation 2 Start Address High Register (IB_START2_HI)

The Inbound Translation 2 Start Address High Register (IB_START2_HI) is shown in Figure 3-91 and described in Table 3-92. The offset is 328h.

**Figure 3-91      Inbound Translation 2 Start Address High Register (IB_START2_HI)**

| 31 | 0 |
|---|---|
| IB_START2_HI | |

R/W-0
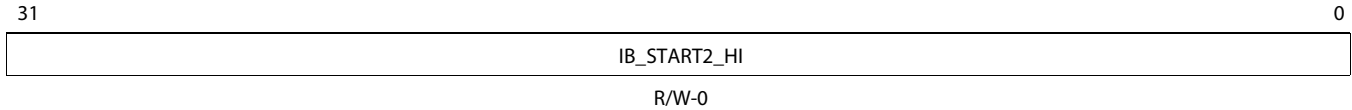
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-92      Inbound Translation 2 Start Address High Register (IB_START2_HI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | IB_START2_HI | Start address bits [63:32] for inbound translation region 2. |
| End of Table 3-92 | | |

### 3.1.93  Inbound Translation 2 Address Offset Register (IB_OFFSET2)

The Inbound Translation 2 Address Offset Register (IB_OFFSET2) is shown in Figure 3-92 and described in Table 3-93. The offset is 32Ch.

**Figure 3-92      Inbound Translation 2 Address Offset Register (IB_OFFSET2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_OFFSET2 | | Reserved | |
| R/W-0 | | R-0 | |
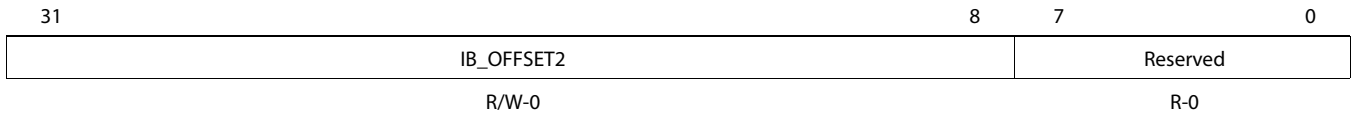
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-93      Inbound Translation 2 Address Offset Register (IB_OFFSET2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_OFFSET2 | Offset address bits [31:8] for inbound translation region 2. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| End of Table 3-93 | | |

### 3.1.94  Inbound Translation Bar Match 3 Register (IB_BAR3)

The Inbound Translation Bar Match 3 Register (IB_BAR3) is shown in Figure 3-93 and described in Table 3-94. The offset is 330h.

**Figure 3-93    Inbound Translation Bar Match 3 Register (IB_BAR3)**

| 31 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | IB_BAR3 | |
| R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-94    Inbound Translation Bar Match 3 Register (IB_BAR3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2-0 | IB_BAR3 | BAR number to match for inbound translation region 3. |
| **End of Table 3-94** | | |

### 3.1.95  Inbound Translation 3 Start Address Low Register (IB_START3_LO)

The Inbound Translation 3 Start Address Low Register (IB_START3_LO) is shown in Figure 3-94 and described in Table 3-95. The offset is 334h.

**Figure 3-94    Inbound Translation 3Start Address Low Register (IB_START3_LO)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_START3_LO | | Reserved | |
| R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-95    Inbound Translation 3 Start Address Low Register (IB_START3_LO) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_START3_LO | Start address bits [31:8] for inbound translation region 3. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-95** | | |

### 3.1.96  Inbound Translation 3Start Address High Register (IB_START3_HI)

The Inbound Translation 3 Start Address High Register (IB_START3_HI) is shown in Figure 3-95 and described in Table 3-96. The offset is 338h.

**Figure 3-95     Inbound Translation 3 Start Address High Register (IB_START3_HI)**

| 31 | 0 |
|---|---|
| IB_START3_HI | |
| R/W-0 | |
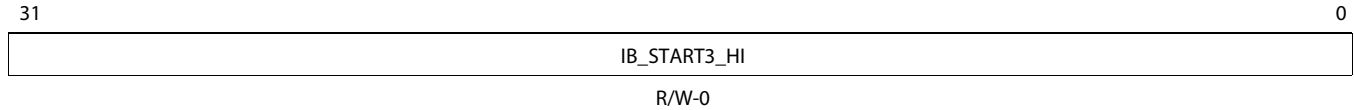
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-96     Inbound Translation 3 Start Address High Register (IB_START3_HI) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | IB_START3_HI | Start address bits [63:32] for inbound translation region 3. |
| **End of Table 3-96** | | |

### 3.1.97  Inbound Translation 3 Address Offset Register (IB_OFFSET3)

The Inbound Translation 3 Address Offset Register (IB_OFFSET3) is shown in Figure 3-96 and described in Table 3-97. The offset is 33Ch.

**Figure 3-96     Inbound Translation 3 Address Offset Register (IB_OFFSET3)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| IB_OFFSET3 | | Reserved | |
| R/W-0 | | R-0 | |
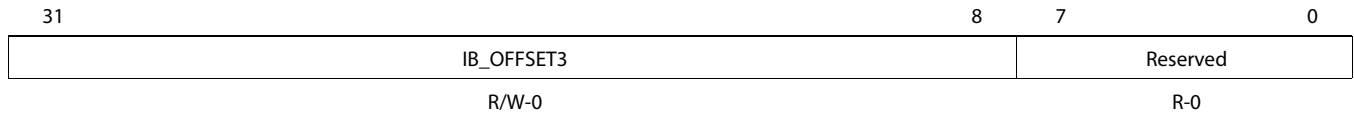
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-97     Inbound Translation 2 Address Offset Register (IB_OFFSET3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | IB_OFFSET3 | Offset address bits [31:8] for inbound translation region 3. |
| 7-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-97** | | |

### 3.1.98  PCS Configuration 0 Register (PCS_CFG0)

The PCS Configuration 0 Register (PCS_CFG0) is shown in Figure 3-97 and described in Table 3-98. The offset is 380h. This register is only available in KeyStone I devices.

**Figure 3-97      PCS Configuration 0 Register (PCS_CFG0)**

| 31 | 29 | 28 | 24 | 23 | 16 | 15 | 14 | 13 | 12 | 11 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Unused | | PCS_SYNC | | PCS_HOLDOFF | | Unused | | PCS_RC_DELAY | | PCS_DET_DELAY | |
| R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | | R/W-2 | | R/W-1 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PCS_SHRT_TM | PCS_STAT186 | PCS_FIX_TERM | PCS_FIX_STD | PCS_L2_ENIDL_OFF | PCS_L0S_RX_OFF | PCS_RXTX_ON | PCS_RXTX_RST |
| R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-98      PCS Configuration 0 Register (PCS_CFG0) Field Description**

| Bit | Field | Description |
|---|---|---|
| 31-29 | Unused | Reserved for future use. |
| 28-24 | PCS_SYNC | Receiver lock/sync control. |
| 23-16 | PCS_HOLDOFF | Receiver initialization holdoff control. |
| 15-14 | Unused | Reserved for future use. |
| 13-12 | PCS_RC_DELAY | Rate change delay. |
| 11-8 | PCS_DET_DELAY | Detection delay. |
| 7 | PCS_SHRT_TM | Enable short times for debug purposes. |
| 6 | PCS_STAT186 | Enable PIPE Spec 1.86 for phystatus behavior. |
| 5 | PCS_FIX_TERM | Set term output to 3'b100 during reset. |
| 4 | PCS_FIX_STD | Fix std output to 2'b10. |
| 3 | PCS_L2_ENIDL_OFF | Deassert enidl during L2 state. |
| 2 | PCS_L0S_RX_OFF | Deassert Rx enable in L0s state. |
| 1 | PCS_RXTX_ON | RX and TX on during reset and TX also on in P1 state. |
| 0 | PCS_RXTX_RST | RX and TX on during reset. |
| **End of Table 3-98** | | |

### 3.1.99 PCS Configuration 1 Register (PCS_CFG1)

The PCS Configuration 1 Register (PCS_CFG1) is shown in Figure 3-98 and described in Table 3-99. The offset is 384h. This register is only available in KeyStone I devices.

**Figure 3-98    PCS Configuration 1 Register (PCS_CFG1)**

| 31      26 | 25        16 | 15       10 | 9      8 | 7        2 | 1      0 |
|------------|--------------|-------------|----------|------------|----------|
| Unused     | PCS_ERR_BIT  | Unused      | PCS_ERR_LN | Unused   | PCS_ERR_MODE |
| R/W-0      | R/W-0        | R/W-0       | R/W-0    | R/W-0      | R/W-0    |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-99    PCS Configuration 1 Register (PCS_CFG1) Field Description**

| Bit   | Field        | Description            |
|-------|--------------|------------------------|
| 31-26 | Unused       | Reserved for future use. |
| 25-16 | PCS_ERR_BIT  | Error bit enable.      |
| 15-10 | Unused       | Reserved for future use. |
| 9-8   | PCS_ERR_LN   | Error lane enable.     |
| 7-2   | Unused       | Reserved for future use. |
| 1-0   | PCS_ERR_MODE | Error injection mode.  |
| End of Table 3-99 | | |

### 3.1.100  PCS Status Register (PCS_STATUS)

The PCS Status Register (PCS_STATUS) is shown in Figure 3-99 and described in Table 3-100. The offset is 388h. This register is only available in KeyStone I devices.

**Figure 3-99    PCS Status Register (PCS_STATUS)**

| 31 |
|----|
| Reserved |
| R-0 |

| 15       | 14       12 | 11      10 | 9      8 | 7      6 | 5      4 | 3      2 | 1      0 |
|----------|-------------|------------|----------|----------|----------|----------|----------|
| Reserved | PCS_REV     | Reserved   | PCS_LN_EN | Reserved | PCS_TX_EN | Reserved | PCS_RX_EN |
| R-0      | R-1         | R-0        | R-3      | R-0      | R-0      | R-0      | R-0      |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-100    PCS Status Register (PCS_STATUS) Field Descriptions**

| Bit   | Field     | Description                              |
|-------|-----------|------------------------------------------|
| 31-15 | Reserved  | Reads return 0 and writes have no effect. |
| 14-12 | PCS_REV   | PCS RTL Revision.                        |
| 11-10 | Reserved  | Reads return 0 and writes have no effect. |
| 9-8   | PCS_LN_EN | PCS lanes enabled status.                |
| 7-6   | Reserved  | Reads return 0 and writes have no effect. |
| 5-4   | PCS_TX_EN | PCS transmitters enabled status.         |
| 3-2   | Reserved  | Reads return 0 and writes have no effect. |
| 1-0   | PCS_RX_EN | PCS receivers enabled status.            |
| End of Table 3-100 | | |

### 3.1.101  SerDes Configuration Lane 0 Register (SERDES_CFG0)

The SerDes Configuration Lane 0 Register (SERDES_CFG0) is shown in Figure 3-100 and described in Table 3-101. The offset is 390h. This register is only available in KeyStone I devices.

**Figure 3-100      SerDes Configuration Lane 0 Register (SERDES_CFG0)**

| 31 | | | | 21 | 20 | | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | TX_LOOPBACK | | | TX_MSYNC | TX_CM | TX_INVPAIR |
| R-0 | | | | | R/W-0 | | | R/W-1 | R/W-1 | R/W-0 |

| 15 | 14 | 13 | 12 | | 9 | 8 | 6 | 5 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RX_LOOPBACK | RX_ENOC | RX_EQ | | | | RX_CDR | | RX_LOS | | RX_ALIGN | | RX_INVPAIR |
| R/W-0 | R/W-1 | R/W-1 | | | | R/W-2 | | R/W-4 | | R/W-0 | | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-101      SerDes Configuration Lane 0 Register (SERDES_CFG0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20-19 | TX_LOOPBACK | Enable TX loopback. Set both bits high to enable.<br>0 = Disabled<br>1h = Reserved<br>2h = Loopback enabled, TX driver disabled. The loopback path covers all the stages of the transmitter except the TX output itself. A differential current is passed to the receiver. The magnitude of this current is dependent on SWING. The transmit driver itself is disabled.<br>3h = Loopback enabled, TX driver enabled. Same as above, but the transmit driver operates normally |
| 18 | TX_MSYNC | Synchronisation master. Enables the channel as the master lane for synchronization purposes. Set to 1 for master lane, set to 0 only for slave lanes in aggregated links. |
| 17 | TX_CM | Common mode adjustment. Always write 1 to this register field. |
| 16 | TX_INVPAIR | Invert polarity. Inverts polarity of TXP and TXN.<br>0 = Normal polarity. TXP is considered to be positive data and TXN is negative.<br>1= Inverted polarity. TXP is considered to be negative data and TXN is positive. |
| 15-14 | RX_LOOPBACK | Enable RX loopback. Set both bits high to enable.<br>0 = Disabled<br>1h = Reserved<br>2h = Reserved<br>3h = RX loopback enabled |
| 13 | RX_ENOC | Enable Rx offset compensation.<br>0 = Compensation disabled<br>1 = Compensation enabled |
| 12-9 | RX_EQ | Enable Rx adaptive equalization. Enables and configures the adaptive equalizer to compensate for loss in the transmission media. |

**Table 3-101    SerDes Configuration Lane 0 Register (SERDES_CFG0) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 8-6 | RX_CDR | Clock/data recovery. Configures the clock/data recovery algorithm.<br>0 = First order, threshold of 1. Phase offset tracking up to ±488ppm. Suitable for use in asynchronous systems with low frequency offset.<br>1h = First order, threshold of 17. Phase offset tracking up to ±325ppm. Suitable for use in synchronous systems. Offers superior rejection of random jitter, but is less responsive to systematic variation such as sinusoidal jitter.<br>2h = Second order, high precision, threshold of 1. Highest precision frequency offset matching but relatively poor response to changes in frequency offset, and long lock time. Suitable for use in systems with fixed frequency offset.<br>3h = Second order, high precision, threshold of 17. Highest precision frequency offset matching but poorest response to changes in frequency offset, and longest lock time. Suitable for use in systems with fixed frequency offset and low systematic variation.<br>4h = Second order, low precision, threshold of 1. Best response to changes in frequency offset and fastest lock time, but lowest precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.<br>5h = Second order, low precision, threshold of 17. Good response to changes in frequency offset and fast lock time, but low precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.<br>Others = Reserved |
| 5-3 | RX_LOS | Loss of signal. Enables loss of signal detection.<br>0 = Disabled.<br>4h = Enabled.<br>Others = Reserved. |
| 2-1 | RX_ALIGN | Symbol alignment. Enables internal or external symbol alignment.<br>0 = Alignment disabled. No symbol alignment will be performed while this setting is selected, or when switching to this selection from another.<br>1 = Comma alignment enabled. Symbol alignment will be performed whenever a misaligned comma symbol is received.<br>Others = Reserved |
| 0 | RX_INVPAIR | Invert polarity. Inverts polarity of RXP and RXN.<br>0 = Normal polarity. RXP is considered to be positive data and RXN is negative.<br>1= Inverted polarity. RXP is considered to be negative data and RXN is positive. |
| **End of Table 3-101** | | |

## 3.1.102 SerDes Configuration Lane 1 Register (SERDES_CFG1)

The SerDes Configuration Lane 1 (SERDES_CFG1) is shown in Figure 3-101 and described in Table 3-102. The offset is 394h. This register is only available in KeyStone I devices.

**Figure 3-101      SerDes Configuration Lane 1 Register (SERDES_CFG1)**

| 31 | | | 21 | 20 | | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | TX_LOOPBACK | | | TX_MSYNC | TX_CM | TX_INVPAIR |
| R-0 | | | | R/W-0 | | | R/W-0 | R/W-1 | R/W-0 |

| 15 | 14 | 13 | 12 | | 9 | 8 | 6 | 5 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RX_LOOPBACK | RX_ENOC | | RX_EQ | | | RX_CDR | | RX_LOS | | RX_ALIGN | | RX_INVPAIR |
| R/W-0 | R/W-1 | | R/W-1 | | | R/W-2 | | R/W-4 | | R/W-0 | | R/W-0 |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-102      SerDes Configuration Lane 1 Register (SERDES_CFG1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20-19 | TX_LOOPBACK | Enable TX loopback. Set both bits high to enable.<br>0 = Disabled<br>1h = Reserved<br>2h = Loopback enabled, TX driver disabled. The loopback path covers all the stages of the transmitter except the TX output itself. A differential current is passed to the receiver. The magnitude of this current is dependent on SWING. The transmit driver itself is disabled.<br>3h = Loopback enabled, TX driver enabled. Same as above, but the transmit driver operates normally |
| 18 | TX_MSYNC | Synchronisation master. Enables the channel as the master lane for synchronization purposes. Set to 1 for master lane, set to 0 only for slave lanes in aggregated links. |
| 17 | TX_CM | Common mode adjustment. Always write 1 to this register field. |
| 16 | TX_INVPAIR | Invert polarity. Inverts polarity of TXP and TXN.<br>0 = Normal polarity. TXP is considered to be positive data and TXN is negative.<br>1= Inverted polarity. TXP is considered to be negative data and TXN is positive. |
| 15-14 | RX_LOOPBACK | Enable RX loopback. Set both bits high to enable.<br>0 = Disabled<br>1h = Reserved<br>2h = Reserved<br>3h = RX loopback enabled |
| 13 | RX_ENOC | Enable Rx offset compensation.<br>0 = Compensation disabled<br>1 = Compensation enabled |
| 12-9 | RX_EQ | Enable Rx adaptive equalization. Enables and configures the adaptive equalizer to compensate for loss in the transmission media. |

**Table 3-102     SerDes Configuration Lane 1 Register (SERDES_CFG1) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 8-6 | RX_CDR | Clock/data recovery. Configures the clock/data recovery algorithm.<br>0 = First order, threshold of 1. Phase offset tracking up to ±488ppm. Suitable for use in asynchronous systems with low frequency offset.<br>1h = First order, threshold of 17. Phase offset tracking up to ±325ppm. Suitable for use in synchronous systems. Offers superior rejection of random jitter, but is less responsive to systematic variation such as sinusoidal jitter.<br>2h = Second order, high precision, threshold of 1. Highest precision frequency offset matching but relatively poor response to changes in frequency offset, and long lock time. Suitable for use in systems with fixed frequency offset.<br>3h = Second order, high precision, threshold of 17. Highest precision frequency offset matching but poorest response to changes in frequency offset, and longest lock time. Suitable for use in systems with fixed frequency offset and low systematic variation.<br>4h = Second order, low precision, threshold of 1. Best response to changes in frequency offset and fastest lock time, but lowest precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.<br>5h = Second order, low precision, threshold of 17. Good response to changes in frequency offset and fast lock time, but low precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.<br>Others = Reserved |
| 5-3 | RX_LOS | Loss of signal. Enables loss of signal detection.<br>0 = Disabled.<br>4h = Enabled.<br>Others = Reserved. |
| 2-1 | RX_ALIGN | Symbol alignment. Enables internal or external symbol alignment.<br>0 = Alignment disabled. No symbol alignment will be performed while this setting is selected, or when switching to this selection from another.<br>1 = Comma alignment enabled. Symbol alignment will be performed whenever a misaligned comma symbol is received.<br>Others = Reserved |
| 0 | RX_INVPAIR | Invert polarity. Inverts polarity of RXP and RXN.<br>0 = Normal polarity. RXP is considered to be positive data and RXN is negative.<br>1= Inverted polarity. RXP is considered to be negative data and RXN is positive. |
| **End of Table 3-102** | | |

## 3.2 Configuration Registers Common to Type 0 and Type 1 Headers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.2.1 Register Summary

**Table 3-103    Configuration Registers Common to Type 0 and Type 1 Headers**

| Offset¹ | Acronym | Register Description | Section |
|---------|---------|---------------------|---------|
| 1000h | VENDOR_DEVICE_ID | Vendor and Device Identification Register | Section 3.2.2 |
| 1004h | STATUS_COMMAND | Status and Command Register | Section 3.2.3 |
| 1008h | CLASSCODE_REVID | Class Code and Revision ID Register | Section 3.2.4 |
| **End of Table 3-103** | | | |

1. The actual addresses of these registers are device specific. See your device-specific data manual to verify the register addresses

### 3.2.2 Vendor and Device Identification Register (VENDOR_DEVICE_ID)

The Vendor and Device Identification Register (VENDOR_DEVICE_ID) is shown in Figure 3-102 and described in Table 3-104. The offset is 1000h.

**Figure 3-102    Vendor and Device Identification Register (VENDOR_DEVICE_ID)**

| 31 | 16 | 15 | 0 |
|----|----|----|---|
| DEVICE_ID | | VENDOR_ID | |
| R-x | | R-104Ch | |
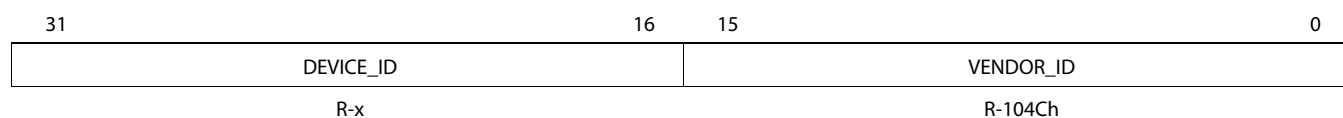
Legend: R = Read only; W = Write only; -n = value after reset; -x = see the device-specific data sheet for the value

**Table 3-104    Vendor and Device Identification Register (VENDOR_DEVICE_ID) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-16 | DEVICE_ID | PCIe Device ID. Writable from internal bus interface. See the device-specific data sheet for the value. |
| 15-0 | VENDOR_ID | PCIe Vendor ID. Writable from internal bus interface. |
| **End of Table 3-104** | | |

## 3.2.3  Status and Command Register (STATUS_COMMAND)

The Status and Command Register (STATUS_COMMAND) is shown in Figure 3-103 and described in Table 3-105. The offset is 1004h.

**Figure 3-103    Status and Command Register (STATUS_COMMAND)**

| 31 | 30 | 29 | 28 | 27 | 26  25 |
|---|---|---|---|---|---|
| PARITY_ERROR | SIG_SYS_ERROR | RX_MST_ABORT | RX_TGT_ABORT | SIG_TGT_ABORT | Reserved |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 |

| 24 | 23  21 | 20 | 19 | 18  11 | 10 | 9 |
|---|---|---|---|---|---|---|
| DAT_PAR_ERROR | Reserved | CAP_LIST | INT_STAT | Reserved | INTX_DIS | Reserved |
| R/W1C-0 | R-0 | R-1 | R-0 | R-0 | R/W-0 | R-0 |

| 8 | 7 | 6 | 5  3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| SERR_EN | Reserved | PAR_ERR_RESP | Reserved | BUS_MS | MEM_SP | IO_SP |
| R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-105    Status and Command Register (STATUS_COMMAND) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | PARITY_ERROR | Set if function receives poisoned TLP. |
| 30 | SIG_SYS_ERROR | Set if function sends an ERR_FATAL or ERR_NONFATAL message and SERR enable bit is set to one. |
| 29 | RX_MST_ABORT | Set when a requester receives a completion with unsupported request completion status. |
| 28 | RX_TGT_ABORT | Set when a requester receives a completion with completer abort status. |
| 27 | SIG_TGT_ABORT | Set when a function acting as a completer terminates a request by issuing completer abort completion status to the requester. |
| 26-25 | Reserved | Reads return 0 and writes have no effect. |
| 24 | DAT_PAR_ERROR | This bit is set by a requester if the Parity Error Enable bit is set in its Command register and either the condition that the requester receives a poisoned completion or the condition that the requester poisons a write request is true. |
| 23-21 | Reserved | Reads return 0 and writes have no effect. |
| 20 | CAP_LIST | For PCIe, this field must be set to 1. |
| 19 | INT_STAT | Indicates that the function has received an interrupt. |
| 18-11 | Reserved | Reads return 0 and writes have no effect. |
| 10 | INTX_DIS | Setting this bit disables generation of INTx messages. |
| 9 | Reserved | Reads return 0 and writes have no effect. |
| 8 | SERR_EN | When set, it enables generation of the appropriate PCI Express error messages to the Root Complex. |
| 7 | Reserved | Reads return 0 and writes have no effect. |
| 6 | PAR_ERR_RESP | This bit controls whether or not the device responds to detected parity errors (poisoned TLP). This error is typically reported as an unsupported request and may also result in a non-fatal error message if SERR_EN=1. If this bit is set, the PCIESS will respond normally to parity errors. If this bit is cleared, the PCIESS will ignore detected parity errors. |
| 5-3 | Reserved | Reads return 0 and writes have no effect. |
| 2 | BUS_MS | Enables mastership of the bus. |
| 1 | MEM_SP | This bit is set to enable the device to respond to memory accesses. |
| 0 | IO_SP | This bit is set to enable the device to respond to I/O accesses. This functionality is not supported in PCIESS and therefore this bit is set to 0. |
| **End of Table 3-105** | | |

### 3.2.4 Class Code and Revision ID Register (CLASSCODE_REVID)

The Class Code and Revision ID Register (CLASSCODE_REVID) is shown in Figure 3-104 and described in Table 3-106. The offset is 1008h.

**Figure 3-104    Class Code and Revision ID Register (CLASSCODE_REVID)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| CLASSCODE | | REVID | |
| R-0h | | R-1h | |
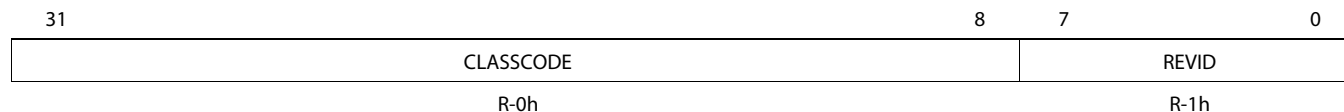
Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-106    Class Code and Revision ID Register (CLASSCODE_REVID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | CLASSCODE | PCIe class code per PCIe Base Specifications Revision 2.0. Writable from internal bus interface. |
| 15-0 | REVID | Updated with each revision of hardware. Writable from internal bus interface. |
| **End of Table 3-106** | | |

## 3.3 Configuration Type 0 Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.3.1 Register Summary

**Table 3-107    Configuration Type0 Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 100Ch | BIST_HEADER | BIST, Header Type, Latency Time and Cache Line Size Register | Section 3.3.2 |
| 1010h | BAR0 | Base Address Register 0 | Section 3.3.3 |
| 1010h | BAR0_MASK | BAR0 Mask Register | Section 3.3.4 |
| 1014h | BAR1 | Base Address Register 1 | Section 3.3.5 |
| 1014h | BAR1_MASK | BAR1 Mask Register | Section 3.3.6 |
| 1014h | BAR1 | Base Address Register 1 (64bit BAR0) | Section 3.3.7 |
| 1014h | BAR1_MASK | BAR1 Mask Register (64bit BAR0) | Section 3.3.8 |
| 1018h | BAR2 | Base Address Register 2 | Section 3.3.9 |
| 1018h | BAR2_MASK | BAR2 Mask Register | Section 3.3.10 |
| 101Ch | BAR3 | Base Address Register 3 | Section 3.3.11 |
| 101Ch | BAR3_MASK | BAR3 Mask Register | Section 3.3.12 |
| 101Ch | BAR3 | Base Address Register 3 (64 bit BAR2) | Section 3.3.13 |
| 101Ch | BAR3_MASK | BAR3 Mask Register (64bit BAR2) | Section 3.3.14 |
| 1020h | BAR4 | Base Address Register 4 | Section 3.3.15 |
| 1020h | BAR4_MASK | BAR4 Mask Register | Section 3.3.16 |
| 1024h | BAR5 | Base Address Register 5 | Section 3.3.17 |
| 1024h | BAR5_MASK | BAR5 Mask Register | Section 3.3.18 |
| 1024h | BAR5 | Base Address Register 5 (64 bit BAR4) | Section 3.3.19 |
| 1024h | BAR5_MASK | BAR5 Mask Register (64bit BAR4) | Section 3.3.20 |
| 102Ch | SUBSYS_VNDR_ID | Subsystem and Subsystem Vendor ID | Section 3.3.21 |
| 1030h | EXPNSN_ROM | Expansion ROM Base Address | Section 3.3.22 |
| 1034h | CAP_PTR | Capabilities Pointer | Section 3.3.23 |
| 103Ch | INT_PIN | Interrupt Pin Register | Section 3.3.24 |
| **End of Table 3-107** | | | |

1. The actual addresses of these registers are device specific. See your device-specific data manual to verify the register addresses

### 3.3.2 BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER)

The BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER) is shown in Figure 3-105 and described in Table 3-108. The offset is 100Ch.

**Figure 3-105 BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER)**

| 31 | 30 | 29 28 | 27 24 | 23 | 22 16 |
|---|---|---|---|---|---|
| BIST_CAP | START_BIST | Reserved | COMP_CODE | MULFUN_DEVe | HDR_TYPE |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

| 15 8 | 7 0 |
|---|---|
| LAT_TMR | CACHELN_SIZ |
| R-0 | R/W-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-108 BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | BIST_CAP | Returns a one for BIST capability and zero otherwise. Not supported by PCIESS. |
| 30 | START_BIST | Write a one to start BIST. Not supported by PCIESS. |
| 29-28 | Reserved | Reads return 0 and writes have no effect. |
| 27-24 | COMP_CODE | Completion code. Not supported by PCIESS. |
| 23 | MULFUN_DEV | Returns 1 if it is a multi-function device. Writable from internal bus interface. |
| 22-16 | HDR_TYPE | Configuration header format.<br>0 = EP mode<br>1 = RC mode |
| 15-8 | LAT_TMR | Not applicable in PCIe |
| 7-0 | CACHELN_SIZ | Not applicable in PCIe |
| **End of Table 3-108** | | |

### 3.3.3 Base Address Register 0 (BAR0)

The Base Address Register 0 (BAR0) is shown in Figure 3-106 and described in Table 3-109. The offset is 1010h.

**Figure 3-106 Base Address Register 0 (BAR0)**

| 31 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | BASE_ADDR | | | PREFETCH | TYPE | | MEM_SPACE |
| | R/W-0 | | | R-0 | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-109 Base Address Register 0 (BAR0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR0 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-109** | | |

### 3.3.4 BAR0 Mask Register (BAR0_MASK)

The BAR0 Mask Register (BAR0_MASK) is shown in Figure 3-107 and described in Table 3-110. The offset is 1010h (same as BAR0, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-107    BAR0 Mask Register (BAR0_MASK)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | BAR_MASK | | BAR_ENABLED |
| | W-0 | | W-0 |
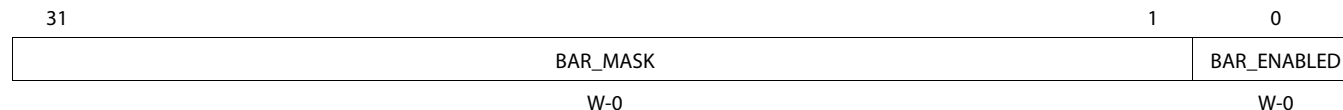
Legend: W = Write only; *-n* = value after reset

**Table 3-110    BAR0 Mask Register (BAR0_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR0 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR0 Enable. The bit is writeable only, not readable.<br>0 = BAR0 is disabled.<br>1 = BAR0 is enabled. |
| **End of Table 3-110** | | |

### 3.3.5 Base Address Register 1 (BAR1)

The Base Address Register 1 (BAR1) is shown in Figure 3-108 and described in Table 3-111. The offset is 1014h.

**Figure 3-108    Base Address Register 1 (BAR1)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |
| R/W-0 | | R-1 | R-0 | | R-0 |
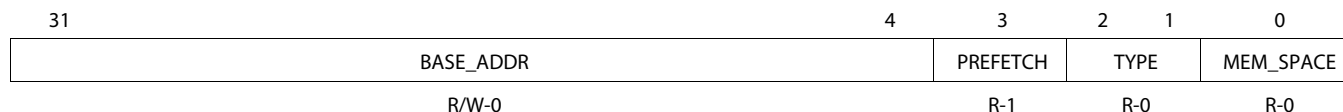
Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-111    Base Address Register 1 (BAR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR1 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br><br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br><br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-111** | | |

### 3.3.6 BAR1 Mask Register (BAR1_MASK)

The BAR1 Mask Register (BAR1_MASK) is shown in Figure 3-109 and described in Table 3-112. The offset is 1014h (same as BAR1, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-109    BAR1 Mask Register (BAR1_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |
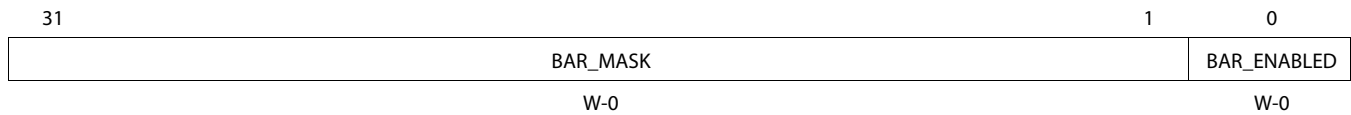
Legend: W = Write only; -*n* = value after reset

**Table 3-112    BAR1 Mask Register (BAR1_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR1 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR1 Enable. The bit is writeable only, not readable.<br>0 = BAR1 is disabled.<br>1 = BAR1 is enabled. |
| **End of Table 3-112** | | |

### 3.3.7  Base Address Register 1 (BAR1) (64bit BAR0)

The Base Address Register 1 (BAR1) (64bit BAR0) is shown in Figure 3-110 and described in Table 3-113. The offset is 1014h.

**Figure 3-110    Base Address Register 1 (64bit BAR0) (BAR1)**

| 31 | 0 |
|---|---|
| BASE_ADDR | |
| R/W-0 | |
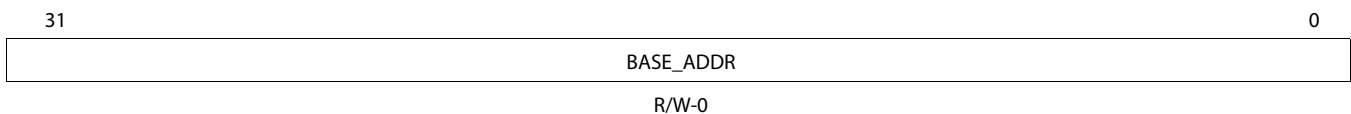
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-113    Base Address Register 1 (64bit BAR0) (BAR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BASE_ADDR | Upper 32 bits of BAR0 address if BAR0 is a 64-bit BAR. Based on the configuration of BAR1 Mask register, not all bits may be writable. |
| **End of Table 3-113** | | |

### 3.3.8 BAR1 Mask Register (BAR1_MASK) (64bit BAR0)

The BAR1 Mask Register (BAR1_MASK) (64bit BAR0) is shown in Figure 3-111 and described in Table 3-114. The offset is 1014h (same as BAR1, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-111 BAR1 Mask Register (BAR1_MASK)**

| 31 | 0 |
|---|---|
| BAR_MASK | |

W-0
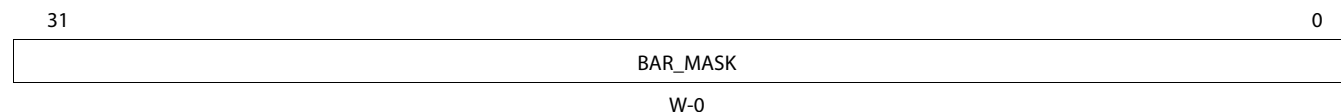
Legend: W = Write only; -*n* = value after reset

**Table 3-114 BAR1 Mask Register (BAR1_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BAR_MASK | Upper 32 bits of BAR0 mask value if BAR0 is a 64-bit BAR. Indicates which BAR bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| End of Table 3-114 | | |

### 3.3.9 Base Address Register 2 (BAR2)

The Base Address Register 2 (BAR2) is shown in Figure 3-112 and described in Table 3-115. The offset is 1018h.

**Figure 3-112 Base Address Register 2 (BAR2)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |

| R/W-0 | R-1 | R-0 | R-0 |
|---|---|---|---|
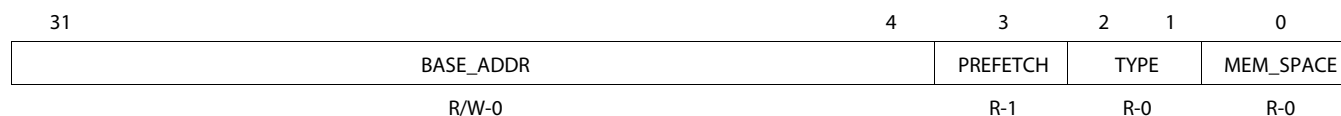
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-115 Base Address Register 2 (BAR2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR2 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| End of Table 3-115 | | |

### 3.3.10 BAR2 Mask Register (BAR2_MASK)

The BAR2 Mask Register (BAR2_MASK) is shown in Figure 3-113 and described in Table 3-116. The offset is 1018h (same as BAR2, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-113     BAR2 Mask Register (BAR2_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |

Legend: W = Write only; -n = value after reset

**Table 3-116     BAR2 Mask Register (BAR2_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR2 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR2 Enable. The bit is writeable only, not readable.<br>0 = BAR2 is disabled.<br>1 = BAR2 is enabled. |
| **End of Table 3-116** | | |

### 3.3.11 Base Address Register 3 (BAR3)

The Base Address Register 3 (BAR3) is shown in Figure 3-114 and described in Table 3-117. The offset is 101Ch.

**Figure 3-114     Base Address Register 3 (BAR3)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |
| R/W-0 | | R-1 | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-117     Base Address Register 3 (BAR3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR3 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-117** | | |

### 3.3.12 BAR3 Mask Register (BAR3_MASK)

The BAR3 Mask Register (BAR3_MASK) is shown in Figure 3-115 and described in Table 3-118. The offset is 101Ch (same as BAR3, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-115    BAR3 Mask Register (BAR3_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |
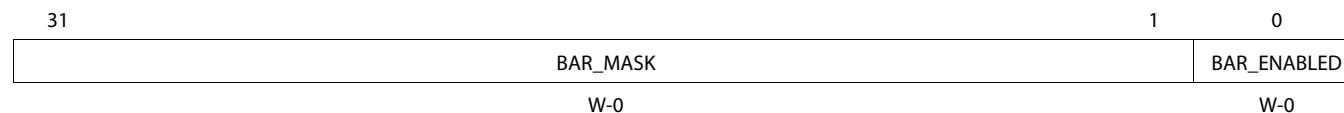
Legend: W = Write only; -*n* = value after reset

**Table 3-118    BAR3 Mask Register (BAR3_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR3 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR3 Enable. The bit is writeable only, not readable.<br>0 = BAR3 is disabled.<br>1 = BAR3 is enabled. |
| **End of Table 3-118** | | |

### 3.3.13 Base Address Register 3 (BAR3) (64bit BAR2)

The Base Address Register 3 (BAR3) (64bit BAR2) is shown in Figure 3-116 and described in Table 3-119. The offset is 101Ch.

**Figure 3-116    Base Address Register 3 (64bit BAR2) (BAR3)**

| 31 | 0 |
|---|---|
| BASE_ADDR | |
| R/W-0 | |
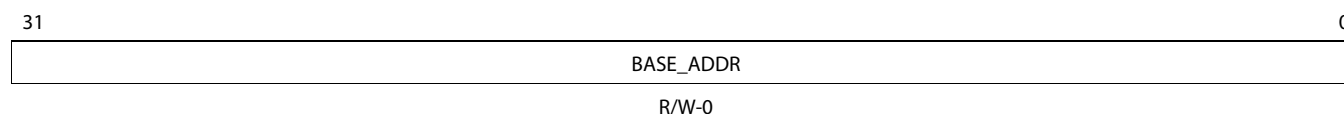
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-119    Base Address Register 3 (64bit BAR2) (BAR3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BASE_ADDR | Upper 32 bits of BAR2 address if BAR2 is a 64-bit BAR. Based on the configuration of BAR3 Mask register, not all bits may be writable. |
| **End of Table 3-119** | | |

### 3.3.14 BAR3 Mask Register (BAR3_MASK) (64bit BAR2)

The BAR3 Mask Register (BAR3_MASK) (64bit BAR2) is shown in Figure 3-117 and described in Table 3-120. The offset is 101Ch (same as BAR3, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-117    BAR3 Mask Register (BAR3_MASK)**

| 31 | 0 |
|---|---|
| BAR_MASK | |

W-0
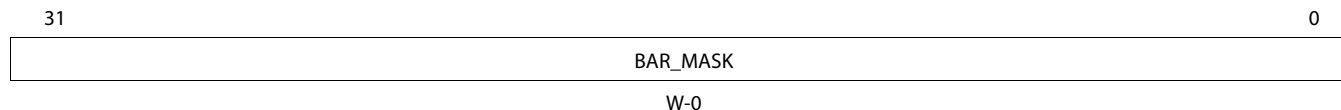
Legend: W = Write only; -n = value after reset

**Table 3-120    BAR3 Mask Register (BAR3_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BAR_MASK | Upper 32 bits of BAR2 mask value if BAR2 is a 64-bit BAR. Indicates which BAR bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| End of Table 3-120 | | |

### 3.3.15 Base Address Register 4 (BAR4)

The Base Address Register 4 (BAR4) is shown in Figure 3-118 and described in Table 3-121. The offset is 1020h.

**Figure 3-118    Base Address Register 4 (BAR4)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |

R/W-0        R-1     R-0     R-0
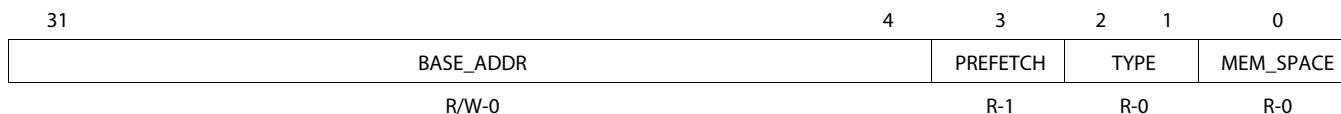
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-121    Base Address Register 4 (BAR4) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR4 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| End of Table 3-121 | | |

### 3.3.16 BAR4 Mask Register (BAR4_MASK)

The BAR4 Mask Register (BAR4_MASK) is shown in Figure 3-119 and described in Table 3-122. The offset is 1020h (same as BAR4, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-119    BAR4 Mask Register (BAR4_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |

Legend: W = Write only; *-n* = value after reset

**Table 3-122    BAR4 Mask Register (BAR4_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR4 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR4 Enable. The bit is writeable only, not readable.<br>0 = BAR4 is disabled.<br>1 = BAR4 is enabled. |
| **End of Table 3-122** | | |

### 3.3.17 Base Address Register 5 (BAR5)

The Base Address Register 5 (BAR5) is shown in Figure 3-120 and described in Table 3-123. The offset is 1024h.

**Figure 3-120    Base Address Register 5 (BAR5)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |
| R/W-0 | | R-1 | R-0 | | R-0 |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-123    Base Address Register 5 (BAR5) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Based on the configuration of BAR5 Mask register, not all bits may be writable. |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-123** | | |

### 3.3.18   BAR5 Mask Register (BAR5_MASK)

The BAR5 Mask Register (BAR5_MASK) is shown in Figure 3-121 and described in Table 3-124. The offset is 1024h (same as BAR5, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1   "BAR Mask Registers" for details).

**Figure 3-121    BAR5 Mask Register (BAR5_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |
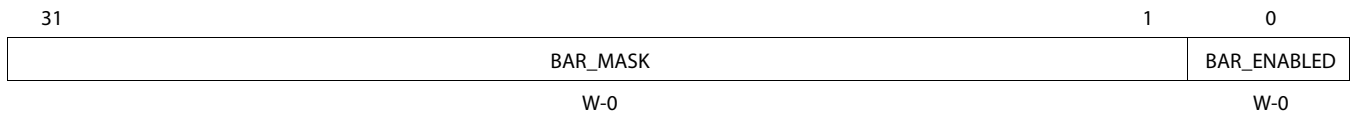
Legend: W = Write only; -n = value after reset

**Table 3-124    BAR5 Mask Register (BAR5_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR5 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR5 Enable. The bit is writeable only, not readable.<br>0 = BAR5 is disabled.<br>1 = BAR5 is enabled. |
| **End of Table 3-124** | | |

### 3.3.19   Base Address Register 5 (BAR5) (64bit BAR4)

The Base Address Register 5 (BAR5) (64bit BAR4) is shown in Figure 3-122 and described in Table 3-125. The offset is 1024h.

**Figure 3-122    Base Address Register 5 (64bit BAR4) (BAR5)**

| 31 | 0 |
|---|---|
| BASE_ADDR | |
| R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-125    Base Address Register 5 (64bit BAR4) (BAR5) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BASE_ADDR | Upper 32 bits of BAR4 address if BAR4 is a 64-bit BAR. Based on the configuration of BAR5 Mask register, not all bits may be writable. |
| **End of Table 3-125** | | |

### 3.3.20 BAR5 Mask Register (BAR5_MASK) (64bit BAR4)

The BAR5 Mask Register (BAR5_MASK) (64bit BAR4) is shown in Figure 3-123 and described in Table 3-126. The offset is 1024h (same as BAR5, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-123    BAR5 Mask Register (BAR5_MASK)**

| 31 | 0 |
|---|---|
| BAR_MASK | |

W-0
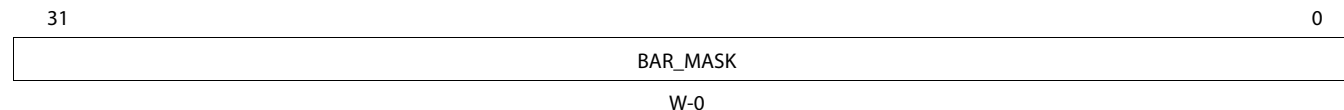
Legend: W = Write only; -n = value after reset

**Table 3-126    BAR5 Mask Register (BAR5_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BAR_MASK | Upper 32 bits of BAR4 mask value if BAR4 is a 64-bit BAR. Indicates which BAR bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| End of Table 3-126 | | |

### 3.3.21 Subsystem and Subsystem Vendor ID (SUBSYS_VNDR_ID)

The Subsystem and Subsystem Vendor ID (SUBSYS_VNDR_ID) is shown in Figure 3-124 and described in Table 3-127. The offset is 102Ch.

**Figure 3-124    Subsystem and Subsystem Vendor ID (SUBSYS_VNDR_ID)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| SUBSYS_ID | | SUBSYS_VEN_ID | |
| R-1 | | R-0 | |
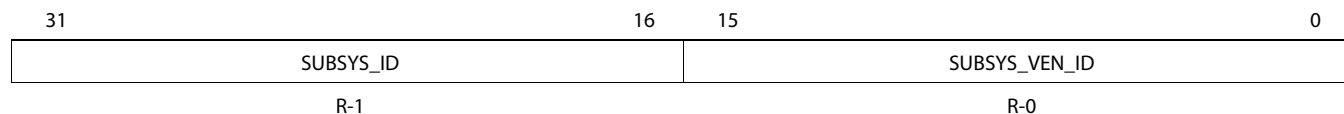
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-127    Subsystem and Subsystem Vendor ID (SUBSYS_VNDR_ID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-16 | SUBSYS_ID | PCIe Subsystem ID. Writable from internal bus interface. |
| 15-0 | SUBSYS_VEN_ID | PCIe Subsystem Vendor ID. Writable from internal bus interface. |
| End of Table 3-127 | | |

### 3.3.22 Expansion ROM Base Address (EXPNSN_ROM)

The Expansion ROM Base Address (EXPNSN_ROM) is shown in Figure 3-125 and described in Table 3-128. The offset is 1030h.

**Figure 3-125    Expansion ROM Base Address (EXPNSN_ROM)**

| 31 | 11 | 10 | 1 | 0 |
|---|---|---|---|---|
| EXP_ROM_BASE_ADDR | | Reserved | | EXP_ROM_EN |
| R-0 | | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-128    Expansion ROM Base Address (EXPNSN_ROM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-11 | EXP_ROM_BASE_ADDR | Address of Expansion ROM |
| 10-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | EXP_ROM_EN | Expansion ROM Enable |
| **End of Table 3-128** | | |

### 3.3.23 Capabilities Pointer (CAP_PTR)

The Capabilities Pointer (CAP_PTR) is shown in Figure 3-126 and described in Table 3-129. The offset is 1034h.

**Figure 3-126    Capabilities Pointer (CAP_PTR)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CAP_PTR | |
| R-0 | | R-40 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-129    Capabilities Pointer (CAP_PTR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reads return 0 and writes have no effect. |
| 7-0 | CAP_PTR | First Capability Pointer. By default, it points to Power Management Capability structure. Writable from internal bus interface. |
| **End of Table 3-129** | | |

### 3.3.24 Interrupt Pin Register (INT_PIN)

The Interrupt Pin Register (INT_PIN) is shown in Figure 3-127 and described in Table 3-130. The offset is 103Ch.

**Figure 3-127    Interrupt Pin Register (INT_PIN)**

| 31 | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
|---|---|---|---|---|---|---|---|
| MAX_LATENCY | | MIN_GRANT | | INT_PIN | | INT_LINE | |
| R-0 | | R-0 | | R-1 | | R/W-FFh | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-130    Interrupt Pin Register (INT_PIN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-24 | MAX_LATENCY | Not applicable to PCI Express. |
| 23-16 | MIN_GRANT | Not applicable to PCI Express. |
| 15-8 | INT_PIN | Interrupt Pin. It identifies the legacy interrupt message that the device uses. For single function configuration, the core only uses INTA. This register is writable through internal bus interface.<br>0 = Legacy interrupt is not being used<br>1h = INTA<br>2h = INTB<br>3h = INTC<br>4h = INTD<br>Others = Reserved |
| 7-0 | INT_LINE | Interrupt Line. Value is system software specified. |
| **End of Table 3-130** | | |

## 3.4 Configuration Type 1 Registers

See the device-specific data manual and PCIe Address Space section for the base address details.

### 3.4.1 Register Summary

**Table 3-131    Configuration Type 1 Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 100Ch | BIST_HEADER | BIST, Header Type, Latency Time and Cache Line Size Register | Section 3.4.2 |
| 1010h | BAR0 | Base Address Register 0 | Section 3.4.3 |
| 1010h | BAR0_MASK | BAR0 Mask Register | Section 3.4.4 |
| 1014h | BAR1 | Base Address Register 1 | Section 3.4.5 |
| 1014h | BAR1_MASK | BAR1 Mask Register | Section 3.4.6 |
| 1014h | BAR1 | Base Address Register 1 (64 bit BAR0) | Section 3.4.7 |
| 1014h | BAR1_MASK | BAR1 Mask Register 1 (64 bit BAR0) | Section 3.4.8 |
| 1018h | BUSNUM | Latency Timer and Bus Number Register | Section 3.4.9 |
| 101Ch | SECSTAT | Secondary Status and IO Space Register | Section 3.4.10 |
| 1020h | MEMSPACE | Memory Limit and Base Register | Section 3.4.11 |
| 1024h | PREFETCH_MEM | Prefetchable Memory Limit and Base Register | Section 3.4.12 |
| 1028h | PREFETCH_BASE | Prefetchable Memory Base Upper 32 bits register | Section 3.4.13 |
| 102Ch | PREFETCH_LIMIT | Prefetchable Limit Upper 32 bits register | Section 3.4.14 |
| 1030h | IOSPACE | IO Base and Limit Upper 16 bits register | Section 3.4.15 |
| 1034h | CAP_PTR | Capabilities Pointer | Section 3.4.16 |
| 1038h | EXPNSN_ROM | Expansion ROM Base Address | Section 3.4.17 |
| 103Ch | BRIDGE_INT | Bridge Control and Interrupt Register | Section 3.4.18 |
| **End of Table 3-131** | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.4.2 BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER)

The BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER) is shown in Figure 3-128 and described in Table 3-132. The offset is 100Ch.

**Figure 3-128    BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER)**

| 31 | 30 | 29 | 28 | 27 | 24 | 23 | 22 | 16 |
|---|---|---|---|---|---|---|---|---|
| BIST_CAP | START_BIST | Reserved | | COMP_CODE | | MULFUN_DEV | HDR_TYPE | |
| R-0 | R-0 | R-0 | | R-0 | | R-0 | R-1 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| LAT_TMR | | CACHELN_SIZE | |
| R-0 | | R-0 | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-132    BIST, Header Type, Latency Time and Cache Line Size Register (BIST_HEADER) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | BIST_CAP | Returns a 1 for BIST capability and 0 otherwise. Not supported by PCIESS. |
| 30 | START_BIST | Write a one to start BIST. Not supported by PCIESS. |
| 29-28 | Reserved | Reads return 0 and writes have no effect. |
| 27-24 | COMP_CODE | Completion Code. Not supported by PCIESS. |
| 23 | MULFUN_DEV | Returns 1 if it is a multi-function device. Writable from internal bus interface. |
| 22-16 | HDR_TYPE | Configuration Header Format.<br>    0 = EP mode<br>    1 = RC mode |
| 15-8 | LAT_TMR | Not applicable in PCIe |
| 7-0 | CACHELN_SIZE | Not applicable in PCIe |
| **End of Table 3-132** | | |

### 3.4.3 Base Address Register 0 (BAR0)

The Base Address Register 0 (BAR0) is shown in Figure 3-129 and described in Table 3-133. The offset is 1010h.

**Figure 3-129    Base Address Register 0 (BAR0)**

| 31 | | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | BASE_ADDR | | | PREFETCH | TYPE | | MEM_SPACE |
| | R-0 | | | R-0 | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-133    Base Address Register 0 (BAR0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Actual writable bits are determined by BAR0 Mask Register |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved<br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-133** | | |

### 3.4.4 BAR0 Mask Register (BAR0_MASK)

The BAR0 Mask Register (BAR0_MASK) is shown in Figure 3-130 and described in Table 3-134. The offset is 1010h (same as BAR0, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-130     BAR0 Mask Register (BAR0_MASK)**

| 31 | | 1 | 0 |
|---|---|---|---|
| | BAR_MASK | | BAR_ENABLED |
| | W-0 | | W-0 |

Legend: W = Write only; *-n* = value after reset

**Table 3-134     BAR0 Mask Register (BAR0_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR0 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR0 Enable. The bit is writeable only, not readable.<br>0 = BAR0 is disabled.<br>1 = BAR0 is enabled. |
| **End of Table 3-134** | | |

### 3.4.5 Base Address Register 1 (BAR1)

The Base Address Register 1 (BAR1) is shown in Figure 3-131 and described in Table 3-135. The offset is 1014h.

**Figure 3-131     Base Address Register 1 (BAR1)**

| 31 | | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| | BASE_ADDR | | PREFETCH | TYPE | | MEM_SPACE |
| | R-0 | | R-0 | R-0 | | R-0 |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-135     Base Address Register 1 (BAR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | BASE_ADDR | Base Address. Actual writable bits are determined by BAR1 Mask Register |
| 3 | PREFETCH | For memory BARs, it indicates whether the region is prefetchable. Writable from internal bus interface.<br>0 = Non-prefetchable.<br>1= Prefetchable.<br><br>For I/O Bars, it is used as second least significant bit (LSB) of the base address. Writable from internal bus interface. |
| 2-1 | TYPE | For memory BARs, they determine the BAR type. Writable from internal bus interface.<br>0 = 32-bit BAR.<br>2h = 64-bit BAR.<br>Others = Reserved.<br><br>For I/O BARs, bit 2 is the least significant bit (LSB) of the base address and bit 1 is 0. Writable from internal bus interface. |
| 0 | MEM_SPACE | Writable from internal bus interface.<br>0 = Memory BAR.<br>1 = I/O BAR. |
| **End of Table 3-135** | | |

### 3.4.6  BAR1 Mask Register (BAR1_MASK)

The BAR1 Mask Register (BAR1_MASK) is shown in Figure 3-132 and described in Table 3-136. The offset is 1014h (same as BAR1, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1  "BAR Mask Registers" for details).

**Figure 3-132    BAR1 Mask Register (BAR1_MASK)**

| 31 | 1 | 0 |
|---|---|---|
| BAR_MASK | | BAR_ENABLED |
| W-0 | | W-0 |
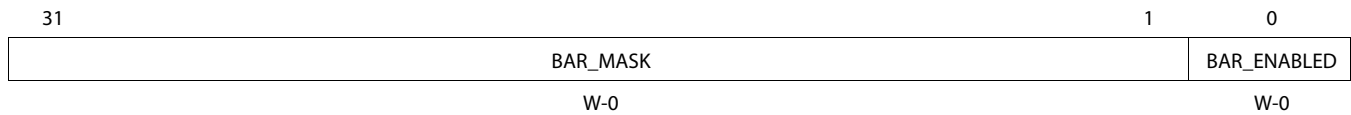
Legend: W = Write only; -n = value after reset

**Table 3-136    BAR1 Mask Register (BAR1_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | BAR_MASK | Indicates which BAR1 bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| 0 | BAR_ENABLED | BAR1 Enable. The bit is writeable only, not readable.<br>0 = BAR1 is disabled.<br>1 = BAR1 is enabled. |
| **End of Table 3-136** | | |

### 3.4.7  Base Address Register 1 (BAR1) (64bit BAR0)

The Base Address Register 1 (BAR1) (64bit BAR0) is shown in Figure 3-133 and described in Table 3-137. The offset is 1014h.

**Figure 3-133    Base Address Register 1 (64bit BAR0)**

| 31 | 0 |
|---|---|
| BASE_ADDR | |
| R-0 | |
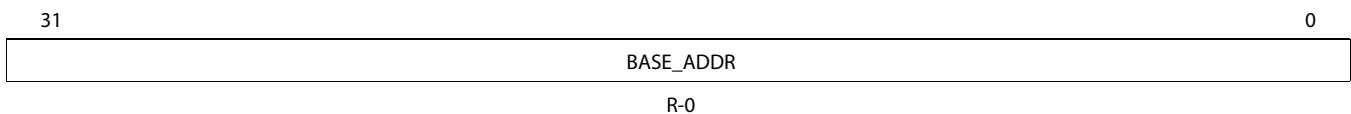
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-137    Base Address Register 1 (64bit BAR0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | BASE_ADDR | Upper 32 bits of BAR0 address if BAR0 is a 64-bit BAR. Based on the configuration of BAR1 Mask register, not all bits may be writable. |
| **End of Table 3-137** | | |

### 3.4.8 BAR1 Mask Register (BAR1_MASK) (64bit BAR0)

The BAR1 Mask Register (BAR1_MASK) (64bit BAR0) is shown in Figure 3-134 and described in Table 3-138. The offset is 1014h (same as BAR1, but requires DBI_CS2 bit set in CMD_STATUS register, see 2.7.3.1 "BAR Mask Registers" for details).

**Figure 3-134    BAR1 Mask Register (BAR1_MASK)**

| 31 | 0 |
|---|---|
| BAR_MASK | |
| W-0 | |
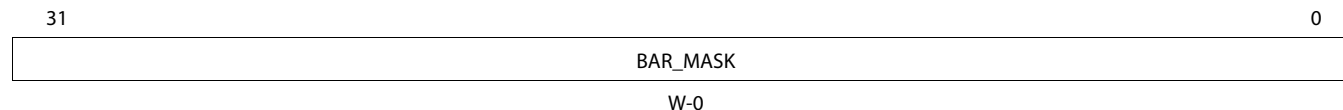
Legend: W = Write only; -n = value after reset

**Table 3-138    BAR1 Mask Register (BAR1_MASK) Field Descriptions**

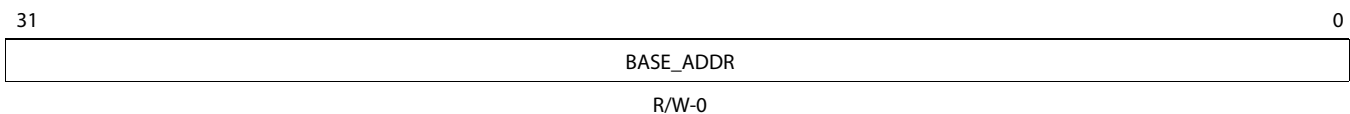| Bit | Field | Description |
|---|---|---|
| 31-0 | BAR_MASK | Upper 32 bits of BAR0 mask value if BAR0 is a 64-bit BAR. Indicates which BAR bits to mask (non-writeable) from host, which determines the size of the BAR. The bits are writeable only, not readable. |
| **End of Table 3-138** | | |

### 3.4.9 Latency Timer and Bus Number Register (BUSNUM)

The Latency Timer and Bus Number Register (BUSNUM) is shown in Figure 3-135 and described in Table 3-139. The offset is 1018h.

**Figure 3-135    Base Latency Timer and Bus Number Register (BUSNUM)**

| 31          24 | 23          16 | 15          8 | 7          0 |
|---|---|---|---|
| SEC_LAT_TMR | SUB_BUS_NUM | SEC_BUS_NUM | PRI_BUS_NUM |
| R-0 | R/W-0 | R/W-0 | R/W-0 |
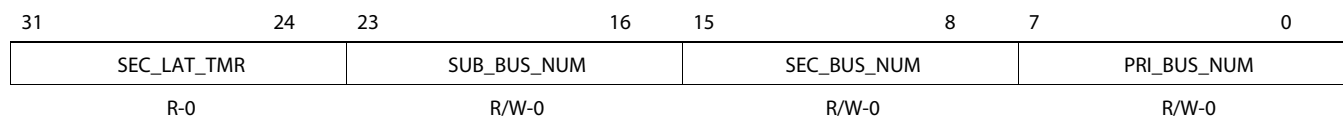
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-139    Base Latency Timer and Bus Number Register (BUSNUM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-24 | SEC_LAT_TMR | Secondary Latency Timer. Not applicable in PCI Express |
| 23:16 | SUB_BUS_NUM | Subordinate Bus Number. This is highest bus number on downstream interface. |
| 15:8 | SEC_BUS_NUM | Secondary Bus Number. It is typically 1h for RC. |
| 7:0 | PRI_BUS_NUM | Primary Bus Number. It is 0 for RC and nonzero for switch devices only. |
| **End of Table 3-139** | | |

## 3.4.10  Secondary Status and IO Base/Limit Register (SECSTAT)

The Secondary Status and IO Space Register (SECSTAT) is shown in Figure 3-136 and described in Table 3-140. The offset is 101Ch.

**Figure 3-136    Secondary Status and IO Space Register (SECSTAT)**

| 31 | 30 | 29 | 28 | 27 | 26  25 | 24 | 23  16 |
|---|---|---|---|---|---|---|---|
| DTCT_PERROR | RX_SYS_ERROR | RX_MST_ABORT | RX_TGT_ABORT | TX_TGT_ABORT | Reserved | MST_DPERR | Reserved |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | R/W1C-0 | R-0 |

| 15          12 | 11          9 | 8 | 7          4 | 3          1 | 0 |
|---|---|---|---|---|---|
| IO_LIMIT | Reserved | IO_LIMIT_ADDR | IO_BASE | Reserved | IO_BASE_ADDR |
| R/W-0 | R-0 | R-0 | R/W-0 | R-0 | R-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-140    Base Secondary Status and IO Space Register (SECSTAT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | DTCT_PERROR | Detected Parity Error. Set if the function receives a poisoned TLP. |
| 30 | RX_SYS_ERROR | Received System Error. Set if the function receives an ERR_FATAL or ERR_NONFATAL message. |
| 29 | RX_MST_ABORT | Received Master Abort. Set if the function receives a completion with unsupported request completion status. |
| 28 | RX_TGT_ABORT | Received Target Abort. Set if the function receives a completion with completer abort completion status. |
| 27 | TX_TGT_ABORT | Signaled Target Abort. Set if the function completes a posted or non-posted request as a completer abort error. |
| 26-25 | Reserved | Reads return 0 and writes have no effect. |
| 24 | MST_DPERR | Master Data Parity Error. Set if the parity error enable bit is set in the Bridge Control Register and either the condition that the requester receives a poisoned completion or the condition that the requester poisons a write request is true. |
| 23-16 | Reserved | Reads return 0 and writes have no effect. |
| 15-12 | IO_LIMIT | Upper 4 bits of 16bit IO Space Limit Address. |
| 11-9 | Reserved | Reads return 0 and writes have no effect. |
| 8 | IO_LIMIT_ADDR | Indicates addressing for IO Limit Address. Writable from internal bus interface.<br>    0 = 16-bit IO addressing.<br>    1 = 32-bit IO addressing. |
| 7-4 | IO_BASE | Upper 4 bits of 16bit IO Space Base Address. |
| 3-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | IO_BASE_ADDR | Indicates addressing for the IO Base Address. Writable from internal bus interface.<br>    0 = 16-bit IO addressing.<br>    1 = 32-bit IO addressing. |
| **End of Table 3-140** | | |

### 3.4.11 Memory Limit and Base Register (MEMSPACE)

The Memory Limit and Base Register (MEMSPACE) is shown in Figure 3-137 and described in Table 3-141. The offset is 1020h.

**Figure 3-137    Memory Limit and Base Register (MEMSPACE)**

| 31 | 20 | 19 | 16 | 15 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| MEM_LIMIT | | Reserved | | MEM_BASE | | Reserved | |
| R/W-0 | | R-0 | | R/W-0 | | R-0 | |

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

**Table 3-141    Memory Limit and Base Register (MEMSPACE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-20 | MEM_LIMIT | Upper 12 bits of 32bit Memory Limit Address. |
| 19-16 | Reserved | Reads return 0 and writes have no effect. |
| 15-4 | MEM_BASE | Upper 12 bits of 32bit Memory Base Address. |
| 3-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-141** | | |

### 3.4.12 Prefetchable Memory Limit and Base Register (PREFETCH_MEM)

The Prefetchable Memory Limit and Base Register (PREFETCH_MEM) is shown in Figure 3-138 and described in Table 3-142. The offset is 1024h.

**Figure 3-138    Prefetchable Memory Limit and Base Register (PREFETCH_MEM)**

| 31 | 20 | 19 | 17 | 16 | 15 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| PREFETCH_LIMIT | | Reserved | | PRE_LIMIT_ADDR | PREFETCH_BASE | | Reserved | | PRE_BASE_ADDR |
| R/W-0 | | R-0 | | R-0 | R-0 | | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-142    Prefetchable Memory Limit and Base Register (PREFETCH_MEM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-20 | PREFETCH_LIMIT | Upper 12 bits of 32bit prefetchable memory limit address (end address). |
| 19-17 | Reserved | Reads return 0 and writes have no effect. |
| 16 | PRE_LIMIT_ADDR | Indicates addressing for prefetchable memory limit address (end address). Writable from internal bus interface.<br>0 = 32-bit memory addressing<br>1 = 64-bit memory addressing |
| 15-4 | PREFETCH_BASE | Upper 12 bits of 32bit prefetchable memory base address (start address). |
| 3-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | PRE_BASE_ADDR | Indicates addressing for the prefetchable memory base address (start address). Writable from internal bus interface.<br>0 = 32-bit memory addressing<br>1 = 64-bit memory addressing |
| **End of Table 3-142** | | |

### 3.4.13 Prefetchable Memory Base Upper 32 bits Register (PREFETCH_BASE)

The Prefetchable Memory Base Upper 32 bits Register (PREFETCH_BASE) is shown in Figure 3-139 and described in Table 3-143. The offset is 1028h.

**Figure 3-139      Prefetchable Memory Base Upper 32 bits Register (PREFETCH_BASE)**

| 31 | 0 |
|---|---|
| PREFETCH_BASE | |

R/W-0
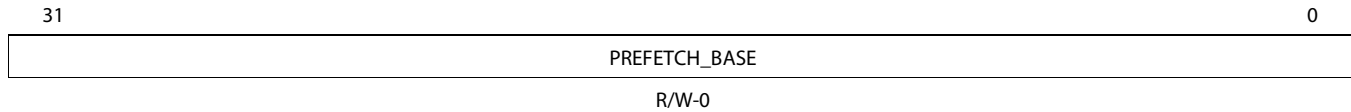
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-143      Prefetchable Memory Base Upper 32 bits Register (PREFETCH_BASE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | PREFETCH_BASE | Upper 32 bits of Prefetchable Memory Base Address. Used with 64bit prefetchable memory addressing only. |
| **End of Table 3-143** | | |

### 3.4.14 Prefetchable Limit Upper 32 bits Register (PREFETCH_LIMIT)

The Prefetchable Limit Upper 32 bits Register (PREFETCH_LIMIT) is shown in Figure 3-140 and described in Table 3-144. The offset is 102Ch.

**Figure 3-140      Prefetchable Limit Upper 32 bits Register (PREFETCH_LIMIT)**

| 31 | 0 |
|---|---|
| PREFETCH_LIMIT | |

R/W-0
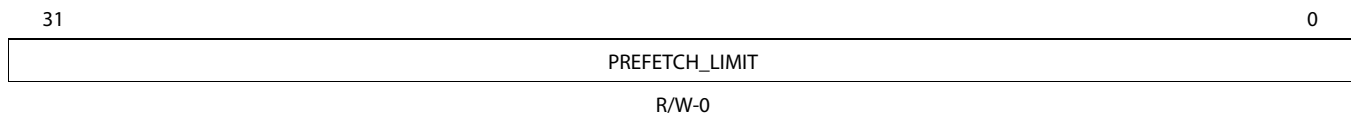
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-144      Prefetchable Limit Upper 32 bits Register (PREFETCH_LIMIT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | PREFETCH_LIMIT | Upper 32 bits of Prefetchable Memory Limit Address. Used with 64 bit prefetchable memory addressing only. |
| **End of Table 3-144** | | |

### 3.4.15 IO Base and Limit Upper 16 bits Register (IOSPACE)

The IO Base and Limit Upper 16 bits Register (IOSPACE) is shown in Figure 3-141 and described in Table 3-145. The offset is 1030h.

**Figure 3-141    IO Base and Limit Upper 16 bits Register (IOSPACE)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| IOBASE | | IOLIMIT | |
| R/W-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-145    IO Base and Limit Upper 16 bits Register (IOSPACE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-16 | IO_BASE | Upper 16 bits of IO Base Address. Used with 32 bit IO space addressing only. |
| 15-0 | IO_LIMIT | Upper 16 bits of IO Limit Address. Used with 32 bit IO space addressing only. |
| **End of Table 3-145** | | |

### 3.4.16 Capabilities Pointer (CAP_PTR)

The Capabilities Pointer (CAP_PTR) is shown in Figure 3-142 and described in Table 3-146. The offset is 1034h.

**Figure 3-142    Capabilities Pointer (CAP_PTR)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CAP_PTR | |
| R-0 | | R-40 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-146    Capabilities Pointer (CAP_PTR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reads return 0 and writes have no effect. |
| 7-0 | CAP_PTR | First Capability Pointer. By default, it points to Power Management Capability structure. Writable from internal bus interface. |
| **End of Table 3-146** | | |

### 3.4.17 Expansion ROM Base Address (EXPNSN_ROM)

The Expansion ROM Base Address (EXPNSN_ROM) is shown in Figure 3-143 and described in Table 3-147. The offset is 1038h.

**Figure 3-143    Expansion ROM Base Address (EXPNSN_ROM)**

| 31 | 11 | 10 | 1 | 0 |
|---|---|---|---|---|
| EXP_ROM_BASE_ADDR | | Reserved | | EXP_ROM_EN |
| R-0 | | R-0 | | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-147    Expansion ROM Base Address (EXPNSN_ROM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-11 | EXP_ROM_BASE_ADDR | Address of Expansion ROM |
| 10-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | EXP_ROM_EN | Expansion ROM Enable |
| **End of Table 3-147** | | |

### 3.4.18 Bridge Control and Interrupt Register (BRIDGE_INT)

The Bridge Control and Interrupt Register (BRIDGE_INT) is shown in Figure 3-144 and described in Table 3-148. The offset is 103Ch.

**Figure 3-144    Bridge Control and Interrupt Register (BRIDGE_INT)**

| 31 | | | | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | SERREN_STATUS | TIMER_STATUS | SEC_TIMER | PRI_TIMER |
| R-0 | | | | | R-0 | R-0 | R-0 | R-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| B2B_EN | SEC_BUS_RST | MST_ABORT_MODE | VGA_DECODE | VGA_EN | ISA_EN | SERR_EN | PERR_RESP_EN |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| INT_PIN | | INT_LINE | |
| R-1 | | R/W-FFh | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-148    Bridge Control and Interrupt Register (BRIDGE_INT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-28 | Reserved | Reads return 0 and writes have no effect. |
| 27 | SERREN_STATUS | Discard Timer SERR Enable Status. Not Applicable to PCI Express. Hardwired to 0. |
| 26 | TIMER_STATUS | Discard Timer Status. Not applicable to PCI Express. Hardwired to 0. |
| 25 | SEC_TIMER | Secondary Discard Timer. Not applicable to PCI Express. Hardwired to 0. |
| 24 | PRI_TIMER | Primary Discard Timer. Not applicable to PCI Express. Hardwired to 0. |
| 23 | B2B_EN | Fast Back to Back Transactions Enable. Not applicable to PCI Express. Hardwired to 0. |
| 22 | SEC_BUS_RST | Secondary Bus Reset. |
| 21 | MST_ABORT_MODE | Master Abort Mode. Not applicable to PCI Express. Hardwired to 0. |
| 20 | VGA_DECODE | VGA 16 bit Decode |
| 19 | VGA_EN | VGA Enable |
| 18 | ISA_EN | ISA Enable |
| 17 | SERR_EN | SERR Enable. Set to enable forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages. |
| 16 | PERR_RESP_EN | Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status Register. |
| 15-8 | INT_PIN | Interrupt Pin. It identifies the legacy interrupt message that the device uses. For single function configuration, the core only uses INTA. This register is writable through internal bus interface.<br>  0 = Legacy interrupt is not being used<br>  1h = INTA<br>  2h = INTB<br>  3h = INTC<br>  4h = INTD<br>  Others = Reserved. |
| 7-0 | INT_LINE | Interrupt Line. Value is system software specified. |
| **End of Table 3-148** | | |

## 3.5 Power Management Capability Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.5.1 Register Summary

**Table 3-149    Power Management Capability Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 1040h | PMCAP | Power Management Capability Register | Section 3.5.2 |
| 1044h | PM_CTL_STAT | Power Management Control and Status Register | Section 3.5.3 |
| End of Table 3-149 | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.5.2 Power Management Capability Register (PMCAP)

The Power Management Capability Register (PMCAP) is shown in Figure 3-145 and described in Table 3-150. The offset is 1040h.

**Figure 3-145    Power Management Capability Register (PMCAP)**

| 31        27 | 26 | 25 | 24        22 | 21 | 20 | 19 | 18        16 |
|---|---|---|---|---|---|---|---|
| PME_SUPP_N | D2_SUPP_N | D1_SUPP_N | AUX_CURR_N | DSI_N | Reserved | PME_CLK | PME_SPEC_VER |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-3 |

| 15                                          8 | 7                                          0 |
|---|---|
| PM_NEXT_PTR | PM_CAP_ID |
| R-50 | R-1 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-150    Power Management Capability Register (PMCAP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-27 | PME_SUPP_N | PME Support. Identifies the power states from which generates PME Messages. A value of 0 for any bit indicates that the device (or function) is not capable of generating PME Messages while in that power state. Writable from internal bus interface.<br>  Bit 31 = If set, PME Messages can be generated from D3cold.<br>  Bit 30 = If set, PME Messages can be generated from D3hot.<br>  Bit 29 = If set, PME Messages can be generated from D2.<br>  Bit 28 = If set, PME Messages can be generated from D1.<br>  Bit 27 = If set, PME Messages can be generated from D0. |
| 26 | D2_SUPP_N | D2 Support. Writable from internal bus interface. |
| 25 | D1_SUPP_N | D1 Support. Writable from internal bus interface. |
| 24-22 | AUX_CURR_N | Auxiliary Current. Writable from internal bus interface. |
| 21 | DSI_N | Device Specific Initialization. Writable from internal bus interface. |
| 20 | Reserved | Reads return 0 and writes have no effect. |
| 19 | PME_CLK | PME Clock. Hardwired to zero. |
| 18-16 | PME_SPEC_VER | Power Management Specification Version. Writable from internal bus interface. |
| 15-8 | PM_NEXT_PTR | Next capability pointer. By default, it points to Message Signaled Interrupt structure.Writable from internal bus interface. |
| 7-0 | PM_CAP_ID | Power Management Capability ID. |
| End of Table 3-150 | | |

### 3.5.3 Power Management Control and Status Register (PM_CTL_STAT)

The Power Management Control and Status Register (PM_CTL_STAT) is shown in Figure 3-146 and described in Table 3-151. The offset is 1044h.

**Figure 3-146     Power Management Control and Status Register (PM_CTL_STAT)**

| 31 | | 24 | 23 | 22 | 21 | | 16 |
|---|---|---|---|---|---|---|---|
| DATA_REG | | | CLK_CTRL_EN | B2_B3_SUPPORT | Reserved | | |
| R-0 | | | R-0 | R-0 | R-0 | | |

| 15 | 14 | 13 | 12 | 9 | 8 | 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PME_STATUS | DATA_SCALE | | DATA_SELECT | | PME_EN | Reserved | | NO_SOFT_RST | Reserved | PWR_STATE | |
| R/W1C-0 | R-0 | | R-0 | | R/W-0 | R-0 | | R-0 | R-0 | R/W-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-151     Power Management Control and Status Register (PM_CTL_STAT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-24 | DATA_REG | Data register for additional information. Not supported. |
| 23 | CLK_CTRL_EN | Bus Power/Clock Control Enable. Hardwired to zero. |
| 22 | B2_B3_SUPPORT | B2 and B3 support. Hardwired to zero. |
| 21-16 | Reserved | Reads return 0 and writes have no effect. |
| 15 | PME_STATUS | PME Status. Indicates if a previously enabled PME event occurred or not. |
| 14-13 | DATA_SCALE | Data Scale. Not supported. |
| 12-9 | DATA_SELECT | Data Select. Not supported. |
| 8 | PME_EN | PME Enable. Value of 1 indicates device is enabled to generate PME. Writable from internal bus interface. |
| 7-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | NO_SOFT_RST | No Soft Reset. It is set to disable reset during a transition from D3 to D0. Writable from internal bus interface. |
| 2 | Reserved | Reads return 0 and writes have no effect. |
| 1-0 | PWR_STATE | Power State. Controls the device power state. Writes are ignored if the state is not supported. Writable from internal bus interface.<br> 0 = D0 power state<br> 1h = D1 power state<br> 2h = D2 power state<br> 3h = D3 power states |
| **End of Table 3-151** | | |

## 3.6 Message Signaled Interrupts Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.6.1 Register Summary

**Table 3-152      Message Signaled Interrupts Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 1050h | MSI_CAP | MSI Capabilities Register | Section 3.6.2 |
| 1054h | MSI_LOW32 | MSI Lower 32 Bits register | Section 3.6.3 |
| 1058h | MSI_UP32 | MSI Upper 32 Bits register | Section 3.6.4 |
| 105Ch | MSI_DATA | MSI Data Register (Offset is 0x1058 if 64-bit addressing not enabled) | Section 3.6.5 |
| **End of Table 3-152** | | | |

1.  The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.6.2 MSI Capabilities Register (MSI_CAP)

The MSI Capabilities Register (MSI_CAP) is shown in Figure 3-147 and described in Table 3-153. The offset is 1050h.

**Figure 3-147    MSI Capabilities Register (MSI_CAP)**

| 31 | 24 | 23 | 22 | 20 | 19 | 17 | 16 | 15 | 8 | 7 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | 64BIT_EN | MULT_MSG_EN | | MULT_MSG_CAP | | MSI_EN | NEXT_CAP | | CAP_ID | |
| R-0 | | R-1 | R/W-0 | | R-0 | | R/W-0 | R-70h | | R-5h | |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-153    MSI Capabilities Register (MSI_CAP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-24 | Reserved | Reads return 0 and writes have no effect. |
| 23 | 64BIT_EN | 64-bit addressing enabled. Writable from internal bus interface. |
| 22-20 | MULT_MSG_EN | Multiple message enabled. Indicates that multiple message mode is enabled by software. Number of messages enabled must not be greater than multiple message capable value.<br>0 = 1<br>1h = 2<br>2h = 4<br>3h = 8<br>4h = 16<br>5h = 32<br>Others = Reserved |
| 19-17 | MULT_MSG_CAP | Multiple message capable. Writable from internal bus interface.<br>0 = 1<br>1h = 2<br>2h = 4<br>3h = 8<br>4h = 16<br>5h = 32<br>Others = Reserved |
| 16 | MSI_EN | MSI Enabled. When set, INTx must be disabled. |
| 15-8 | NEXT_CAP | Next capability pointer. By default, it points to PCI Express Capabilities structure. Writable from internal bus interface. |
| 7-0 | CAP_ID | MSI Capability ID. |
| **End of Table 3-153** | | |

### 3.6.3 MSI Lower 32 Bits Register (MSI_LOW32)

The MSI Lower 32 Bits Register (MSI_LOW32) is shown in Figure 3-148 and described in Table 3-154. The offset is 1054h.

**Figure 3-148    MSI Lower 32 Bits Register (MSI_LOW32)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| LOW32_ADDR | | Reserved | |
| R/W-0 | | R-0 | |
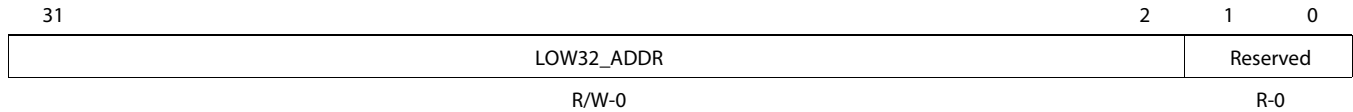
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-154    MSI Lower 32 Bits Register (MSI_LOW32) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | LOW32_ADDR | Lower 32 bits address |
| 1-0 | Reserved | Reads return 0 and writes have no effect. |

### 3.6.4 MSI Upper 32 Bits Register (MSI_UP32)

The MSI Upper 32 Bits Register (MSI_UP32) is shown in Figure 3-149 and described in Table 3-155. The offset is 1058h.

**Figure 3-149    MSI Upper 32 Bits Register (MSI_UP32)**

| 31 | 0 |
|---|---|
| UP32_ADDR | |
| R/W-0 | |
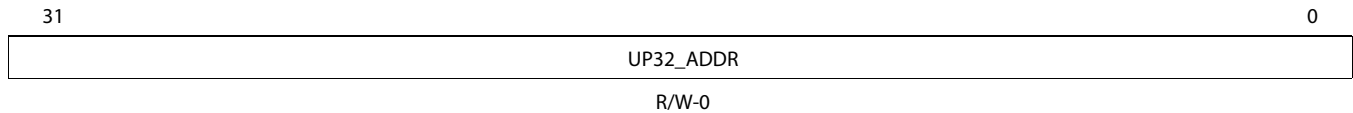
Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-155    MSI Upper 32 Bits Register (MSI_UP32) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | UP32_ADDR | Upper 32 bits address |
| **End of Table 3-155** | | |

### 3.6.5 MSI Data Register (MSI_DATA)

The MSI Data Register (MSI_DATA) is shown in Figure 3-150 and described in Table 3-156. The offset is 105Ch (the offset is 1058h if 64-bit addressing not enabled).

**Figure 3-150    MSI Data Register (MSI_DATA)**
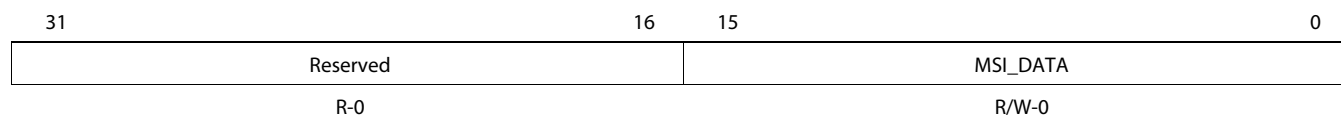
| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | MSI_DATA | |
| R-0 | | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

**Table 3-156    MSI Data Register (MSI_DATA) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-16 | Reserved | Reads return 0 and writes have no effect. |
| 15-0 | MSI_DATA | MSI Data |
| **End of Table 3-156** | | |

## 3.7 PCI Express Capabilities Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.7.1 Register Summary

**Table 3-157     PCI Express Capabilities Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 1070h | PCIE_CAP | PCI Express Capabilities Register | Section 3.7.2 |
| 1074h | DEVICE_CAP | Device Capabilities Register | Section 3.7.3 |
| 1078h | DEV_STAT_CTRL | Device Status and Control Register | Section 3.7.4 |
| 107Ch | LINK_CAP | Link Capabilities Register | Section 3.7.5 |
| 1080h | LINK_STAT_CTRL | Link Status and Control Register | Section 3.7.6 |
| 1084h | SLOT_CAP | Slot Capabilities Register (RC Mode only) | Section 3.7.7 |
| 1088h | SLOT_STAT_CTRL | Slot Status and Control Register (RC Mode only) | Section 3.7.8 |
| 108Ch | ROOT_CTRL_CAP | Root Control and Capabilities Register (RC Mode only) | Section 3.7.9 |
| 1090h | ROOT_STATUS | Root Status and Control Register (RC Mode only) | Section 3.7.10 |
| 1094h | DEV_CAP2 | Device Capabilities 2 Register | Section 3.7.11 |
| 1098h | DEV_STAT_CTRL2 | Device Status and Control Register 2 | Section 3.7.12 |
| 10A0h | LINK_CTRL2 | Link Control Register 2 | Section 3.7.13 |
| **End of Table 3-157** | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.7.2 PCI Express Capabilities Register (PCIE_CAP)

The PCI Express Capabilities Register (PCIE_CAP) is shown in Figure 3-151 and described in Table 3-158. The offset is 1070h.

**Figure 3-151    PCI Express Capabilities Register (PCIE_CAP)**

| 31 | 30 | 29 | 25 | 24 | 23 | 20 | 19 | 16 |
|----|----|----|----|----|----|----|----|----|
| Reserved | | INT_MSG | | SLT_IMPL_N | DPORT_TYPE | | PCIE_CAP | |
| R-0 | | R-0 | | R-0 | R-4h/0h | | R-2 | |

| 15 | 8 | 7 | 0 |
|----|----|----|----|
| NEXT_CAP | | CAP_ID | |
| R-0 | | R-10h | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-158    PCI Express Capabilities Register (PCIE_CAP) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-30 | Reserved | Reads return 0 and writes have no effect. |
| 29-25 | INT_MSG | Interrupt Message Number. Updated by hardware and writable through internal bus Interface. |
| 24 | SLT_IMPL_N | Slot Implemented. Writable from internal bus interface. |
| 23-20 | DPORT_TYPE | Device Port Type.<br>　0 = EP type<br>　4h = RC type<br>　Others = Reserved |
| 19-16 | PCIE_CAP | PCI Express Capability Version |
| 15-8 | NEXT_CAP | Next capability pointer. Writable from internal bus interface. |
| 7-0 | CAP_ID | PCIe Capability ID. |
| **End of Table 3-158** | | |

### 3.7.3 Device Capabilities Register (DEVICE_CAP)

The Device Capabilities Register (DEVICE_CAP) is shown in Figure 3-152 and described in Table 3-159. The offset is 1074h.

**Figure 3-152    Device Capabilities Register (DEVICE_CAP)**

| 31 | 28 | 27 | 26 | 25 | | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | PWR_LIMIT_SCALE | | PWR_LIMIT_VALUE | | | Reserved | |
| R-0 | | R-0 | | R-0 | | | R-0 | |

| 15 | 14 | 12 | 11 | 9 | 8 | 6 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ERR_RPT | Reserved | | L1_LATENCY | | L0_LATENCY | | EXT_TAG_FLD | PHANTOM_FLD | | MAX_PAYLD_SZ | |
| R-1 | R-0 | | R-0h/3h | | R-0h/4h | | R-0 | R-0 | | R-1 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-159    Device Capabilities Register (DEVICE_CAP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-28 | Reserved | Reads return 0 and writes have no effect. |
| 27-26 | PWR_LIMIT_SCALE | Captured Slot Power Limit Scale. For upstream ports (EP ports) only. |
| 25-18 | PWR_LIMIT_VALUE | Captured Slot Power Limit Value. For upstream ports (EP ports) only. |
| 17-16 | Reserved | Reads return 0 and writes have no effect. |
| 15 | ERR_RPT | Role-based Error Reporting. Writable from internal bus interface. |
| 14-12 | Reserved | Reads return 0 and writes have no effect. |
| 11-9 | L1_LATENCY | Endpoint L1 Acceptable Latency. Must be 0 in RC mode. It is 3h for EP mode. |
| 8-6 | L0_LATENCY | Endpoint L0s Acceptable Latency. Must be 0 in RC mode. It is 4h for EP mode. |
| 5 | EXT_TAG_FLD | Extended Tag Field Supported. Writable from internal interface but should not be as the hardware is not capable. |
| 4-3 | PHANTOM_FLD | Phantom Field Supported. Writable from internal bus interface. |
| 2-0 | MAX_PAYLD_SZ | Maximum Payload size supported. Writable from internal bus interface. |
| **End of Table 3-159** | | |

### 3.7.4 Device Status and Control Register (DEV_STAT_CTRL)

The Device Status and Control Register (DEV_STAT_CTRL) is shown in Figure 3-153 and described in Table 3-160. The offset is 1078h.

**Figure 3-153    Device Status and Control Register (DEV_STAT_CTRL)**

| 31 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----|----|----|----|----|----|----|----|
| Reserved | | TPEND | AUX_PWR | UNSUP_RQ_DET | FATAL_ERR | NFATAL_ERR | CORR_ERR |
| R-0 | | R-0 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

| 15 | 14 | 12 | 11 | 10 | 9 | 8 |
|----|----|----|----|----|----|----|
| Reserved | MAX_REQ_SZ | | NO_SNOOP | AUX_PWR_PM_EN | PHANTOM_EN | XTAG_FIELD_EN |
| R-0 | R/W-2 | | R/W-1 | R/W/S-0 | R/W-0 | R/W-0 |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|
| MAX_PAYLD | | RELAXED | UNSUP_REQ_RP | FATAL_ERR_RP | NFATAL_ERR_RP | CORR_ERR_RP |
| R/W-0 | | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-160    Device Status and Control Register (DEV_STAT_CTRL) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-22 | Reserved | Reads return 0 and writes have no effect. |
| 21 | TPEND | Transaction Pending |
| 20 | AUX_PWR | Auxiliary Power Detected |
| 19 | UNSUP_RQ_DET | Unsupported Request Detected |
| 18 | FATAL_ERR | Fatal Error Detected |
| 17 | NFATAL_ERR | Non-fatal Error Detected |
| 16 | CORR_ERR | Correctable Error Detected |
| 15 | Reserved | Reads return 0 and writes have no effect. |
| 14-12 | MAX_REQ_SZ | Maximum Read Request Size |
| 11 | NO_SNOOP | Enable no snoop |
| 10 | AUX_PWR_PM_EN | AUX Power PM Enable |
| 9 | PHANTOM_EN | Phantom Function Enable |
| 8 | XTAG_FIELD_EN | Extended Tag Field Enable |
| 7-5 | MAX_PAYLD | Maximum Payload Size |
| 4 | RELAXED | Enable Relaxed Ordering |
| 3 | UNSUP_REQ_RP | Enable Unsupported Request Reporting |
| 2 | FATAL_ERR_RP | Fatal Error Reporting Enable |
| 1 | NFATAL_ERR_RP | Non-fatal Error Reporting Enable |
| 0 | CORR_ERR_RP | Correctable Error Reporting Enable |
| **End of Table 3-160** | | |

### 3.7.5 Link Capabilities Register (LINK_CAP)

The Link Capabilities Register (LINK_CAP) is shown in Figure 3-154 and described in Table 3-161. The offset is 107Ch.

**Figure 3-154   Link Capabilities Register (LINK_CAP)**

| 31 | | | | | 24 |
|---|---|---|---|---|---|
| PORT_NUM | | | | | |
| R-0 | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 |
|---|---|---|---|---|---|---|
| Reserved | | BW_NOTIFY_CAP | DLL_REP_CAP | DOWN_ERR_REP_CAP | CLK_PWR_MGMT | L1_EXIT_LAT |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-6 |

| 15 | 14 | 12 | 11 | 10 | 9 |
|---|---|---|---|---|---|
| L1_EXIT_LAT | LOS_EXIT_LAT | | AS_LINK_PM | | MAX_LINK_WIDTH |
| R-6 | R-5 | | R-1 | | R-2 |

| 4 | 3 | 0 |
|---|---|---|
| MAX_LINK_WIDTH | MAX_LINK_SPEED | |
| R-2 | R-2 | |
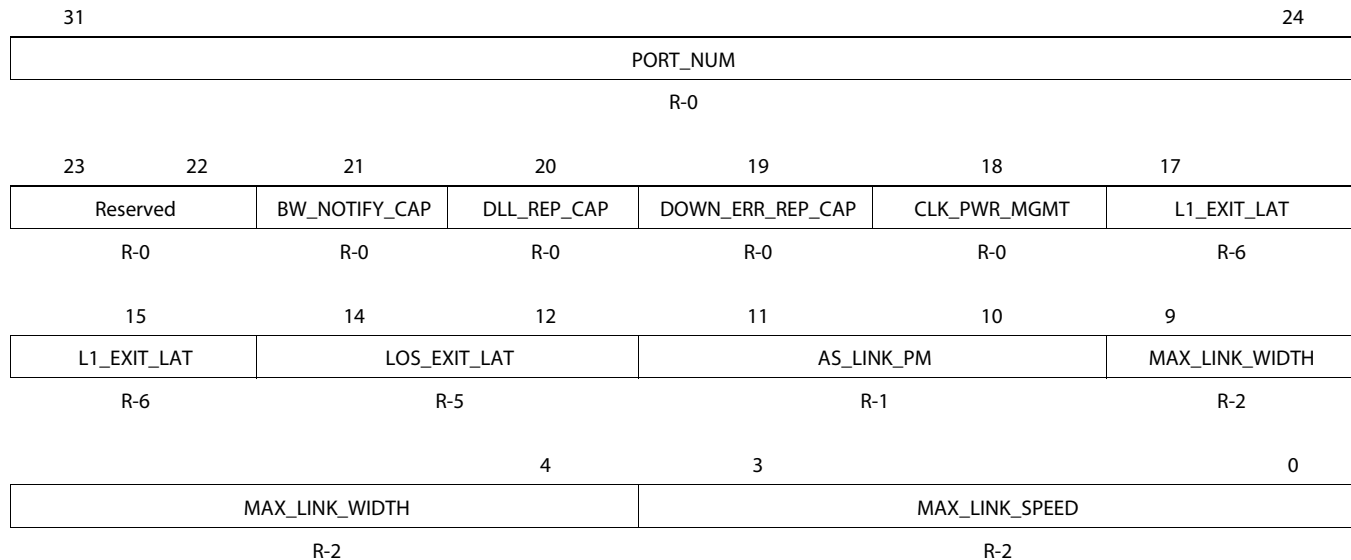
Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-161   Link Capabilities Register (LINK_CAP) Field Descriptions (Part 1 of 2)**

| Bit | Field | Description |
|---|---|---|
| 31-24 | PORT_NUM | Port Number. Writable from internal bus interface. |
| 23-22 | Reserved | Reads return 0 and writes have no effect. |
| 21 | BW_NOTIFY_CAP | Link Bandwidth Notification Capable.<br>0 = For upstream ports (EP ports)<br>1 = For downstream ports (RC ports) |
| 20 | DLL_REP_CAP | Data Link Layer Active Reporting Capable.<br>0 = For upstream ports (EP ports)<br>1 = For downstream ports (RC ports) |
| 19 | DOWN_ERR_REP_CAP | Surprise Down Error Reporting Capable. Not supported. Always zero. |
| 18 | CLK_PWR_MGMT | Clock Power Management. Writable from internal bus interface.<br><br>For upstream ports (EP Ports), a value of 1h in this bit indicates that the component tolerates the removal of any reference clock(s) in the L1 and L2/L3 Ready Link states. A value of 0 indicates the reference clock(s) must not be removed in these Link states.<br><br>For downstream ports (RC Ports), this bit is always 0. |
| 17-15 | L1_EXIT_LAT | L1 Exit Latency when common clock is used. Writable from internal bus interface.<br>0 = Less than 64 ns<br>1h = 64 ns to less than 128 ns<br>2h = 128 ns to less than 256 ns<br>3h = 256 ns to less than 512 ns<br>4h = 512 ns to less than 1 µs<br>5h = 1 µs to less than 2 µs<br>6h = 2 µs to less than 4 µs<br>7h = More than 4 µs |

**Table 3-161    Link Capabilities Register (LINK_CAP) Field Descriptions (Part 2 of 2)**

| Bit | Field | Description |
|---|---|---|
| 14-12 | LOS_EXIT_LAT | L0s Exit Latency. Writable from internal bus interface.<br>0 = Less than 64 ns<br>1h = 64 ns to less than 128 ns<br>2h = 128 ns to less than 256 ns<br>3h = 256 ns to less than 512 ns<br>4h = 512 ns to less than 1 µs<br>5h = 1 µs to less than 2 µs<br>6h = 2 µs to less than 4 µs<br>7h = More than 4 µs |
| 11-10 | AS_LINK_PM | Active State Link Power Management Support. Writable from internal bus interface.<br>1h = L0s entry supported.<br>3h = L0s and L1 supported.<br>Others = Reserved. |
| 9-4 | MAX_LINK_WIDTH | Maximum Link Width (×N – corresponding to N Lanes). Writable from internal bus interface.<br>1h = ×1<br>2h = ×2<br>Others = Reserved. |
| 3-0 | MAX_LINK_SPEED | Maximum Link Speed. Writable from internal bus interface.<br>1h = 2.5GT/s Link speed supported.<br>2h = 5.0 GT/s and 2.5 GT/s Link speeds supported.<br>Others = Reserved. |
| **End of Table 3-161** | | |

### 3.7.6 Link Status and Control Register (LINK_STAT_CTRL)

The Link Status and Control Register (LINK_STAT_CTRL) is shown in Figure 3-155 and described in Table 3-162. The offset is 1080h.

**Figure 3-155    Link Status and Control Register (LINK_STAT_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 |
|---|---|---|---|---|---|
| LINK_BW_STATUS | LINK_BW_MGMT_STATUS | DLL_ACTIVE | SLOT_CLK_CFG | LINK_TRAINING | Reserved |
| R/W1C-0 | R/W1C-0 | R-0 | R-1 | R-0 | R-0 |

| 25 | | | 20 | 19 | | | 16 |
|---|---|---|---|---|---|---|---|
| NEGOTIATED_LINK_WD | | | | LINK_SPEED | | | |
| R-1 | | | | R-1 | | | |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | LINK_BW_INT_EN | LINK_BW_MGMT_INT_EN | HW_AUTO_WIDTH_DIS | CLK_PWR_MGMT_EN |
| R-0 | | R-0 | R-0 | R-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EXT_SYNC | COMMON_CLK_CFG | RETRAIN_LINK | LINK_DISABLE | RCB | Reserved | ACTIVE_LINK_PM | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R-0 | R/W-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-162    Link Status and Control Register (LINK_STAT_CTRL) Field Descriptions (Part 1 of 2)**

| Bit | Field | Description |
|---|---|---|
| 31 | LINK_BW_STATUS | Link Autonomous Bandwidth Status. |
| | | This bit is Set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. |
| | | This bit must be set if the Physical Layer reports a speed or width change was initiated by the downstream component that was indicated as an autonomous change. |
| | | Not applicable and reserved for EP. |
| 30 | LINK_BW_MGMT_STATUS | Link Bandwidth Management Status. |
| | | This bit is Set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: |
| | | • A Link retraining has completed following a write of 1b to the Retrain Link bit |
| | | • Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. |
| | | This bit must be set if the Physical Layer reports a speed or width change was initiated by the downstream component that was not indicated as an autonomous change. |
| | | Not applicable and reserved for EP. |
| 29 | DLL_ACTIVE | Data Link Layer Active |
| | | This bit indicates the status of the Data Link Control and Management State Machine. It returns a 1 to indicate the DL_Active state, 0 otherwise. |
| 28 | SLOT_CLK_CFG | Slot Clock Configuration. Writable from internal bus interface. |
| | | This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. |
| 27 | LINK_TRAINING | Link Training. Not applicable to EP. |
| 26 | UNDEF | Undefined for PCI Express. |

**Table 3-162    Link Status and Control Register (LINK_STAT_CTRL) Field Descriptions (Part 2 of 2)**

| Bit | Field | Description |
|---|---|---|
| 25-20 | NEGOTIATED_LINK_WD | Negotiated Link Width. Set automatically by hardware after link initialization. |
| 19-16 | LINK_SPEED | Link Speed. Set automatically by hardware after link initialization. |
| 15-12 | Reserved | Reads return 0 and writes have no effect. |
| 11 | LINK_BW_INT_EN | Link Autonomous Bandwidth Interrupt Enable. Not applicable and is reserved for EP. |
| 10 | LINK_BW_MGMT_INT_EN | Link Bandwidth Management Interrupt Enable. Not applicable and is reserved for EP. |
| 9 | HW_AUTO_WIDTH_DIS | Hardware Autonomous Width Disable. Not supported and hardwired to zero. |
| 8 | CLK_PWR_MGMT_EN | Enable Clock Power Management. |
| 7 | EXT_SYNC | Extended Synchronization. |
| 6 | COMMON_CLK_CFG | Common Clock Configuration.<br>0 = Indicates that this device and the device at the opposite end of the link are operating with separate reference clock sources.<br>1 = Indicates that this device and the device at the opposite end of the link are operating with a common clock source. |
| 5 | RETRAIN_LINK | Retrain Link. Not applicable and reserved for EP. |
| 4 | LINK_DISABLE | This bit disables the link by directing the LTSSM to the Disabled state when set. Not applicable and is reserved for EP. |
| 3 | RCB | Read Completion Boundary. Writable via internal bus interface for RC.<br>0 = 64 bytes<br>1 = 128 bytes |
| 2 | Reserved | Reads return 0 and writes have no effect. |
| 1-0 | ACTIVE_LINK_PM | Active State Link Power Management Control<br>0 = Disabled.<br>1h = L0s entry enabled.<br>2h = L1 entry enabled.<br>3h = L0s and L1 entry enabled. |
| **End of Table 3-162** | | |

### 3.7.7 Slot Capabilities Register (SLOT_CAP)

The Slot Capabilities Register (SLOT_CAP) (RC Mode only) is shown in Figure 3-156 and described in Table 3-163. This register is for RC mode only and the offset is 1084h.

**Figure 3-156    Slot Capabilities Register (SLOT_CAP)**

| 31 | 19 | 18 | 17 | 16 | 15 |
|---|---|---|---|---|---|
| SLOT_NUM | | CMD_COMP_SUPP | EML_PRESENT | PWR_LMT_SCALE | |
| R-0 | | R-0 | R-0 | R-0 | |

| 14 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PWR_LMT_VALUE | | HP_CAP | HP_SURPRISE | PWR_IND | ATTN_IND | MRL_SENSOR | PWR_CTL | ATTN_BUTTON |
| R-0 | | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-163    Slot Capabilities Register (SLOT_CAP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-19 | SLOT_NUM | Physical Slot Number. Writable from internal bus interface. |
| 18 | CMD_COMP_SUPP | No Command Complete Support. Writable from internal bus interface. |
| | | When Set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. |
| 17 | EML_PRESENT | Electromechanical Interlock Present. Writable from internal bus interface. |
| | | When Set, this bit indicates that an Electromechanical Interlock is implemented on the chassis for this slot. |
| 16-15 | PWR_LMT_SCALE | Slot Power Limit Scale. Writable from internal bus interface. |
| 14-7 | PWR_LMT_VALUE | Slow Power Limit Value. Writable from internal bus interface. |
| 6 | HP_CAP | Hot Plug Capable. Writable from internal bus interface. |
| 5 | HP_SURPRISE | Hot Plug Surprise. Writable from internal bus interface. |
| 4 | PWR_IND | Power Indicator Present. Writable from internal bus interface. |
| 3 | ATTN_IND | Attention Indicator Present. Writable from internal bus interface. |
| 2 | MRL_SENSOR | MRL Sensor Present. Writable from internal bus interface. |
| 1 | PWR_CTL | Power Controller Present. Writable from internal bus interface. If there is no power controller, software must ensure that system power is up before reading Presence Detect state. |
| 0 | ATTN_BUTTON | Attention Indicator Present. Writable from internal bus interface. |
| **End of Table 3-163** | | |

### 3.7.8 Slot Status and Control Register (SLOT_STAT_CTRL)

The Slot Status and Control Register (SLOT_STAT_CTRL) is shown in Figure 3-157 and described in Table 3-164. This register is for RC mode only and the offset is 1088h.

**Figure 3-157    Slot Status and Control Register (SLOT_STAT_CTRL)**

| 31 | | | | | | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | DLL_STATE |
| R-0 | | | | | | | R/W1C-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| EM_LOCK | PRESENCE_DET | MRL_STATE | CMD_COMLETE | PRESENCE_CHG | MRL_CHANGE | PWR_FAULT | ATTN_PRESSED |
| R-0 | R-1 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

| 15 | | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DLL_CHG_EN | EM_LOCK_CTL | PM_CTL | PM_IND_CTL | |
| R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-3 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ATTN_IND_CTL | | HP_INT_EN | CMD_CMP_INT_EN | PRS_DET_CHG_EN | MRL_CHG_EN | PWR_FLT_DET_EN | ATTN_BUTT_EN |
| R/W-3 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 3-164    Slot Status and Control Register (SLOT_STAT_CTRL) Field Descriptions (Part 1 of 2)**

| Bit | Field | Description |
|---|---|---|
| 31-25 | Reserved | Reads return 0 and writes have no effect. |
| 24 | DLL_STATE | Data Link Layer State Changed |
| 23 | EM_LOCK | Electromechanical Lock Status |
| 22 | PRESENCE_DET | Presence Detect State |
| 21 | MRL_STATE | MRL Sensor State |
| 20 | CMD_COMLETE | Command Completed |
| 19 | PRESENCE_CHG | Presence Detect Changed |
| 18 | MRL_CHANGE | MRL Sensor Changed |
| 17 | PWR_FAULT | Power Fault Detected |
| 16 | ATTN_PRESSED | Attention Button Pressed. |
| 15-13 | Reserved | Reads return 0 and writes have no effect. |
| 12 | DLL_CHG_EN | Data Link Layer State Changed Enable. |
| 11 | EM_LOCK_CTL | Electromechanical Interlock Control. |
| 10 | PM_CTL | Power Controller Control |
| 9-8 | PM_IND_CTL | Power Indicator Control |
| 7-6 | ATTN_IND_CTL | Attention Indicator Control. |
| 5 | HP_INT_EN | Hot Plug Interrupt Enable. |
| 4 | CMD_CMP_INT_EN | Command Completed Interrupt Enable. |
| 3 | PRS_DET_CHG_EN | Presence Detect Changed Enable. |
| 2 | MRL_CHG_EN | MRL Sensor Changed Enable. |

**Table 3-164     Slot Status and Control Register (SLOT_STAT_CTRL) Field Descriptions (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 1 | PWR_FLT_DET_EN | Power Fault Detected Enable. |
| 0 | ATTN_BUTT_EN | Attention Button Pressed Enable. |
| **End of Table 3-164** | | |

## 3.7.9  Root Control and Capabilities Register (ROOT_CTRL_CAP)

The Root Control and Capabilities Register (ROOT_CTRL_CAP) is shown in Figure 3-158 and described in Table 3-165. This register is for RC mode only and the offset is 108Ch.

**Figure 3-158     Root Control and Capabilities Register (ROOT_CTRL_CAP)**

| 31 | | | | | 17 | 16 |
|----|----|----|----|----|----|----|
| Reserved | | | | | | CRS_SW |
| R-0 | | | | | | R-0 |

| 15 | | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|
| Reserved | | | CRS_SW_EN | PME_INT_EN | SERR_FATAL_ERR | SERR_NFATAL_ERR | SERR_EN |
| R-0 | | | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-165     Root Control and Capabilities Register (ROOT_CTRL_CAP) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-17 | Reserved | Reads return 0 and writes have no effect. |
| 16 | CRS_SW | CRS Software Visibility. Not supported and set to 0. |
| 15-5 | Reserved | Reads return 0 and writes have no effect. |
| 4 | CRS_SW_EN | CRS Software Visibility Enable. Not supported and set to 0x0. |
| 3 | PME_INT_EN | PME Interrupt Enable |
| 2 | SERR_FATAL_ERR | System Error on Fatal Error Enable |
| 1 | SERR_NFATAL_ERR | System Error on Non-fatal Error Enable |
| 0 | SERR_EN | System Error on Correctable Error Enable |
| **End of Table 3-165** | | |

### 3.7.10 Root Status and Control Register (ROOT_STATUS)

The Root Status and Control Register (ROOT_STATUS) is shown in Figure 3-159 and described in Table 3-166. This register is for RC mode only and the offset is 1090h.

**Figure 3-159    Root Status and Control Register (ROOT_STATUS)**

| 31 | 18 | 17 | 16 | 15 | 0 |
|---|---|---|---|---|---|
| Reserved | | PME_PEND | PME_STATUS | PME_REQ_ID | |
| R-0 | | R-0 | R/W1C-0 | R-0 | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; -n = value after reset

**Table 3-166    Root Status and Control Register (ROOT_STATUS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-18 | Reserved | Reads return 0 and writes have no effect. |
| 17 | PME_PEND | Indicates that another PME is pending when the PME Status bit is Set. |
| 16 | PME_STATUS | Indicates that PME was asserted by the PME Requester. |
| 15-0 | PME_REQ_ID | ID of the last PME Requester. |
| | | This field is only valid when the PME Status bit is Set. |
| **End of Table 3-166** | | |

### 3.7.11 Device Capabilities 2 Register (DEV_CAP2)

The Device Capabilities 2 Register (DEV_CAP2) is shown in Figure 3-160 and described in Table 3-167. The offset is 1094h.

**Figure 3-160    Device Capabilities 2 Register (DEV_CAP2)**

| 31 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|
| Reserved | | CMPL_TO_DIS_SUPP | CMPL_TO_EN | |
| R-0 | | R-1 | R-Fh | |

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

**Table 3-167    Device Capabilities 2 Register (DEV_CAP2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reads return 0 and writes have no effect. |
| 4 | CMPL_TO_DIS_SUPP | Completion timeout disable supported |
| 3-0 | CMPL_TO_EN | Completion timeout ranges supported. Applicable to RC/EP that issue requests on own behalf. |
| **End of Table 3-167** | | |

## 3.7.12 Device Status and Control Register 2 (DEV_STAT_CTRL2)

The Device Status and Control Register 2 (DEV_STAT_CTRL2) is shown in Figure 3-161 and described in Table 3-168. The offset is 1098h.

**Figure 3-161    Device Status and Control Register 2 (DEV_STAT_CTRL2)**

| 31 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|
| Reserved | | CMPL_TO_DIS | CMPL_TO | |
| R-0 | | R/W-0 | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-168    Device Status and Control Register 2 (DEV_STAT_CTRL2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reads return 0 and writes have no effect. |
| 4 | CMPL_TO_DIS | Completion timeout disable |
| 3-0 | CMPL_TO | Completion timeout value.<br><br>It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms.<br>0 = Default range: 50 s to 50 ms.<br>1h = 50 s to 100 s.<br>2h = 1 ms to 10 ms.<br>5h = 16 ms to 55 ms.<br>6h = 65 ms to 210 ms.<br>9h = 260 ms to 900 ms.<br>Ah = 1 s to 3.5 s.<br>Dh = 4 s to 13 s.<br>Eh = 17 s to 64 s.<br>Others = Reserved. |
| **End of Table 3-168** | | |

### 3.7.13 Link Control Register 2 (LINK_CTRL2)

The Link Control Register 2 (LINK_CTRL2) is shown in Figure 3-162 and described in Table 3-169. The offset is 10A0h.

**Figure 3-162    Link Control Register 2 (LINK_CTRL2)**

| 31 | | | | | 17 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | | | | | DE_EMPH |
| R-0 | | | | | | R-1 |

| 15 | 13 | 12 | 11 | 10 | 9 |
|---|---|---|---|---|---|
| Reserved | | POLL_DEEMPH | CMPL_SOS | ENTR_MOD_COMPL | TX_MARGIN |
| R-0 | | R/W/S-0 | R/W/S-0 | R/W/S-0 | R/W/S-0 |

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| TX_MARGIN | SEL_DEEMPH | HW_AUTO_SPEED_DIS | ENTR_COMPL | TGT_SPEED | |
| R/W-0 | R/W-0 | R/W/S-0 | R/W/S-0 | R/W/S-2 | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-169    Link Control Register 2 (LINK_CTRL2) Field Descriptions (Part 1 of 2)**

| Bit | Field | Description |
|---|---|---|
| 31-17 | Reserved | Reads return 0 and writes have no effect. |
| 16 | DE_EMPH | Current De-emphasis level<br>0 = -6 dB<br>1 = -3.5 dB |
| 15-13 | Reserved | Reads return 0 and writes have no effect. |
| 12 | POLL_DEEMPH | De-emphasis level in polling-compliance state<br>This bit sets the de-emphasis level in Polling Compliance state if the entry occurred due to the Enter Compliance bit being 1.<br>0 = -6 dB<br>1 = -3.5 dB |
| 11 | CMPL_SOS | Compliance SOS.<br>When this bit is set to 1, the LTSSM is required to send SKP Ordered Sets periodically in between the modified compliance patterns. |
| 10 | ENTR_MOD_COMPL | Enter modified compliance.<br>When this bit is set to 1, the device transmits Modified Compliance Pattern if the LTSSM enters Polling Compliance substate. |
| 9-7 | TX_MARGIN | Value of non-de-emphasized voltage level at transmitter pins. |
| 6 | SEL_DEEMPH | Selectable De-emphasis.<br>When the Link is operating at 5.0 GT/s speed, this bit selects the level of de-emphasis for an upstream component.<br>When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect.<br>0 = -6 dB<br>1 = -3.5 dB |
| 5 | HW_AUTO_SPEED_DIS | Hardware Autonomous Speed Disable.<br>0 = Enable hardware to change the link speed.<br>1 = Disables hardware from changing the Link speed for device specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. |

**Table 3-169    Link Control Register 2 (LINK_CTRL2) Field Descriptions (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 4 | ENTR_COMPL | Enter Compliance. <br><br> Software is permitted to force a Link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1 in both components on a Link and then initiating a hot reset on the Link. |
| 3-0 | TGT_SPEED | Target Link Speed. <br> 1h = 2.5 GT/s Target Link Speed. <br> 2h = 5.0 GT/s Target Link Speed. <br> Others = Reserved. |
| **End of Table 3-169** | | |

## 3.8 PCI Express Extended Capabilities Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.8.1 Register Summary

**Table 3-170    PCI Express Extended Capabilities Registers**

| Offset[1] | Acronym | Register Description | Section |
|---|---|---|---|
| 1100h | PCIE_EXTCAP | PCI Express Extended Capabilities Header | Section 3.8.2 |
| 1104h | PCIE_UNCERR | PCI Express Uncorrectable Error Status Register | Section 3.8.3 |
| 1108h | PCIE_UNCERR_MASK | PCI Express Uncorrectable Error Mask Register | Section 3.8.4 |
| 110Ch | PCIE_UNCERR_SVRTY | PCI Express Uncorrectable Error Severity Register | Section 3.8.5 |
| 1110h | PCIE_CERR | PCI Express Correctable Error Status Register | Section 3.8.6 |
| 1114h | PCIE_CERR_MASK | PCI Express Correctable Error Mask Register | Section 3.8.7 |
| 1118h | PCIE_ACCR | PCI Express Advanced Capabilities and Control Register | Section 3.8.8 |
| 111Ch | HDR_LOG0 | Header Log Register 0 | Section 3.8.9 |
| 1120h | HDR_LOG1 | Header Log Register 1 | Section 3.8.10 |
| 1124h | HDR_LOG2 | Header Log Register 2 | Section 3.8.11 |
| 1128h | HDR_LOG3 | Header Log Register 3 | Section 3.8.12 |
| 112Ch | ROOT_ERR_CMD | Root Error Command Register | Section 3.8.13 |
| 1130h | ROOT_ERR_ST | Root Error Status Register | Section 3.8.14 |
| 1134h | ERR_SRC_ID | Error Source Identification Register | Section 3.8.15 |
| **End of Table 3-170** | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.8.2 PCI Express Extended Capabilities Header (PCIE_EXTCAP)

The PCI Express Extended Capabilities Header (PCIE_EXTCAP) is shown in Figure 3-163 and described in Table 3-171. The offset is 1100h.

**Figure 3-163    PCI Express Extended Capabilities Header (PCIE_EXTCAP)**

| 31 | 20 | 19 | 16 | 15 | 0 |
|---|---|---|---|---|---|
| NEXT_CAP | | EXT_CAP_VER | | EXT_CAP_ID | |
| R-0 | | R-1 | | R-1 | |

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

**Table 3-171    PCI Express Extended Capabilities Header (PCIE_EXTCAP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-20 | NEXT_CAP | Next Capability Pointer |
| 19-16 | EXT_CAP_VER | Extended Capability Version |
| 15-0 | EXT_CAP_ID | PCIe Extended Capability ID |
| **End of Table 3-171** | | |

### 3.8.3 PCI Express Uncorrectable Error Status Register (PCIE_UNCERR)

The PCI Express Uncorrectable Error Status Register (PCIE_UNCERR) is shown in Figure 3-164 and described in Table 3-172. The offset is 1104h.

**Figure 3-164    PCI Express Uncorrectable Error Status Register (PCIE_UNCERR)**

| 31 | | | | 21 | 20 | 19 | 18 | 17 | 16 |
|----|--|--|--|----|----|----|----|----|----|
| Reserved | | | | | UR_ERR_ST | ECRC_ERR_ST | MTLP_ERR_ST | RCVR_OF_ST | UCMP_ST |
| R-0 | | | | | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 |

| 15 | 14 | 13 | 12 | 11 | | |
|----|----|----|----|----|--|--|
| CMPL_ABRT_ST | CMPL_TMOT_ST | FCP_ERR_ST | PSND_TLP_ST | Reserved | | |
| R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R-0 | | |

| 6 | 5 | 4 | 3 | | 0 |
|---|---|---|---|--|---|
| Reserved | SRPS_DN_ST | DLP_ERR_ST | Reserved | | |
| R-0 | R/S-0 | R/W1C/S-0 | R-0 | | |

Legend: R = Read only; W = Write only; W1C = Write 1 to clear, write of 0 has no effect; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-172    Table 70. PCI Express Uncorrectable Error Status Register (PCIE_UNCERR) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20 | UR_ERR_ST | Unsupported Request Error Status |
| 19 | ECRC_ERR_ST | ECRC Error Status |
| 18 | MTLP_ERR_ST | Malformed TLP Status |
| 17 | RCVR_OF_ST | Receiver Overflow Status |
| 16 | UCMP_ST | Unexpected Completion Status |
| 15 | CMPL_ABRT_ST | Completer Abort Status |
| 14 | CMPL_TMOT_ST | Completion Timeout Status |
| 13 | FCP_ERR_ST | Flow Control Protocol Error Status |
| 12 | PSND_TLP_ST | Poisoned TLP Status |
| 11-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | SRPS_DN_ST | Surprise Down Error Status. Not supported. |
| 4 | DLP_ERR_ST | Data Link Protocol Error Status |
| 3-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-172** | | |

### 3.8.4 PCI Express Uncorrectable Error Mask Register (PCIE_UNCERR_MASK)

The PCI Express Uncorrectable Error Mask Register (PCIE_UNCERR_MASK) is sFigure 3-165 and described in Table 3-173. The offset is 1108h.

**Figure 3-165    PCI Express Uncorrectable Error Mask Register (PCIE_UNCERR_MASK)**

| 31 | | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | UR_ERR_MSK | ECRC_ERR_MSK | MTLP_ERR_MSK | RCVR_OF_MSK | UCMP_MSK |
| R-0 | | | R/W/S-0 | R/W/S-0 | R/W/S-0 | R/W/S-0 | R/W/S-0 |

| 15 | 14 | 13 | 12 | 11 | | | 0 |
|---|---|---|---|---|---|---|---|
| CMPL_ABRT_MSK | CMPL_TMOT_MSK | FCP_ERR_MSK | PSND_TLP_MSK | Reserved | | | |
| R/W/S-0 | R/W/S-0 | R/W/S-0 | R/W/S-0 | R-0 | | | |

| 6 | 5 | 4 | 3 | | | | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | SRPS_DN_MSK | DLP_ERR_MSK | Reserved | | | | |
| R-0 | R/S-0 | R/W/S-0 | R-0 | | | | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-173    PCI Express Uncorrectable Error Mask Register (PCIE_UNCERR_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20 | UR_ERR_MSK | Unsupported Request Error Mask |
| 19 | ECRC_ERR_MSK | ECRC Error Mask |
| 18 | MTLP_ERR_MSK | Malformed TLP Mask |
| 17 | RCVR_OF_MSK | Receiver Overflow Mask |
| 16 | UCMP_MSK | Unexpected Completion Mask |
| 15 | CMPL_ABRT_MSK | Completer Abort Mask |
| 14 | CMPL_TMOT_MSK | Completion Timeout Mask |
| 13 | FCP_ERR_MSK | Flow Control Protocol Error Mask |
| 12 | PSND_TLP_MSK | Poisoned TLP Mask |
| 11-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | SRPS_DN_MSK | Surprise Down Error Mask. Not supported. |
| 4 | DLP_ERR_MSK | Data Link Protocol Error Mask |
| 3-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-173** | | |

### 3.8.5 PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY)

The PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY) is shown in Figure 3-166 and described in Table 3-174. The offset is 110Ch.

**Figure 3-166    PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY)**

| 31 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | UR_ERR_SVRTY | ECRC_ERR_SVRTY | MTLP_ERR_SVRTY | RCVR_OF_SVRTY | UCMP_SVRTY |
| R-0 | | R/W/S-0 | R/W/S-0 | R/W/S-1 | R/W/S-1 | R/W/S-0 |

| 15 | 14 | 13 | 12 | 11 | | |
|---|---|---|---|---|---|---|
| CMPL_ABRT_SVRTY | CMPL_TMOT_SVRTY | FCP_ERR_SVRTY | PSND_TLP_SVRTY | Reserved | | |
| R/W/S-0 | R/W/S-0 | R/W/S-1 | R/W/S-0 | R-0 | | |

| 6 | | 5 | 4 | 3 | | 0 |
|---|---|---|---|---|---|---|
| Reserved | | SRPS_DN_SVRTY | DLP_ERR_SVRTY | Reserved | | |
| R-0 | | R/S-1 | R/W/S-1 | R-0 | | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-174    PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY) Field Descriptions  (Part 1 of 2)**

| Bit | Field | Description |
|---|---|---|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20 | UR_ERR_SVRTY | Unsupported Request Error Severity<br>0 = Non-fatal<br>1 = Fatal |
| 19 | ECRC_ERR_ SVRTY | ECRC Error Severity<br>0 = Non-fatal<br>1 = Fatal |
| 18 | MTLP_ERR_ SVRTY | Malformed TLP Severity<br>0 = Non-fatal<br>1 = Fatal |
| 17 | RCVR_OF_ SVRTY | Receiver Overflow Severity<br>0 = Non-fatal<br>1 = Fatal |
| 16 | UCMP_ SVRTY | Unexpected Completion Severity<br>0 = Non-fatal<br>1 = Fatal |
| 15 | CMPL_ABRT_ SVRTY | Completer Abort Severity<br>0 = Non-fatal<br>1 = Fatal |
| 14 | CMPL_TMOT_ SVRTY | Completion Timeout Severity<br>0 = Non-fatal<br>1 = Fatal |
| 13 | FCP_ERR_ SVRTY | Flow Control Protocol Error Severity<br>0 = Non-fatal<br>1 = Fatal |
| 12 | PSND_TLP_ SVRTY | Poisoned TLP Severity<br>0 = Non-fatal<br>1 = Fatal |
| 11-6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | SRPS_DN_ SVRTY | Surprise Down Error Severity. Not supported. |

**Table 3-174     PCI Express Uncorrectable Error Severity Register (PCIE_UNCERR_SVRTY) Field Descriptions  (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 4 | DLP_ERR_ SVRTY | Data Link Protocol Error Severity<br>0 = Non-fatal<br>1 = Fatal |
| 3-0 | Reserved | Reads return 0 and writes have no effect. |
| **End of Table 3-174** | | |

## 3.8.6  PCI Express Correctable Error Status Register (PCIE_CERR)

The PCI Express Correctable Error Status Register (PCIE_CERR) is shown in Figure 3-167 and described in Table 3-175. The offset is 1110h.

**Figure 3-167     PCI Express Correctable Error Status Register (PCIE_CERR)**

| 31 | | 14 | 13 | 12 | 11 | 9 | 8 |
|----|--|----|----|----|----|---|---|
| Reserved | | | ADV_NFERR_ST | RPLY_TMR_ST | Reserved | | RPLT_RO_ST |
| R-0 | | | R/W1C/S-0 | R/W1C/S-0 | R-0 | | R/W1C/S-0 |

| 7 | 6 | 5 | | | 1 | 0 |
|---|---|---|--|--|---|---|
| BAD_DLLP_ST | BAD_TLP_ST | Reserved | | | | RCVR_ERR_ST |
| R/W1C/S-0 | R/W1C/S-0 | R-0 | | | | R/W1C/S-0 |

Legend: R = Read only; W1C = Write 1 to clear, write of 0 has no effect; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-175     PCI Express Correctable Error Status Register (PCIE_CERR) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-14 | Reserved | Reads return 0 and writes have no effect. |
| 13 | ADV_NFERR_ST | Advisory Non-Fatal Error Status |
| 12 | RPLY_TMR_ST | Replay Timer Timeout Status |
| 11-9 | Reserved | Reads return 0 and writes have no effect. |
| 8 | RPLT_RO_ST | REPLAY_NUM Rollover Status |
| 7 | BAD_DLLP_ST | Bad DLLP Status |
| 6 | BAD_TLP_ST | Bad TLP Status |
| 5-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | RCVR_ERR_ST | Receiver Error Status |
| **End of Table 3-175** | | |

## 3.8.7 PCI Express Correctable Error Mask Register (PCIE_CERR_MASK)

The PCI Express Correctable Error Mask Register (PCIE_CERR_MASK) is shown in Figure 3-168 and described in Table 3-176. The offset is 1114h.

**Figure 3-168   PCI Express Correctable Error Mask Register (PCIE_CERR_MASK)**

| 31 | | 14 | 13 | 12 | 11 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | ADV_NFERR_MSK | RPLY_TMR_MSK | Reserved | | RPLT_RO_MSK |
| R-0 | | | R/W/S-1 | R/W/S-0 | R-0 | | R/W/S-0 |

| 7 | 6 | 5 | | | | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BAD_DLLP_MSK | BAD_TLP_MSK | Reserved | | | | | RCVR_ERR_MSK |
| R/W/S-0 | R/W/S-0 | R-0 | | | | | R/W/S-0 |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-176   PCI Express Correctable Error Mask Register (PCIE_CERR_MASK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-14 | Reserved | Reads return 0 and writes have no effect. |
| 13 | ADV_NFERR_MSK | Advisory Non Fatal Error Mask<br>This bit is Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting. |
| 12 | RPLY_TMR_MSK | Reply Timer Timeout Mask |
| 11-9 | Reserved | Reads return 0 and writes have no effect. |
| 8 | RPLT_RO_MSK | REPLAY_NUM Rollover Mask |
| 7 | BAD_DLLP_MSK | Bad DLLP Mask |
| 6 | BAD_TLP_MSK | Bad TLP Mask |
| 5-1 | Reserved | Reads return 0 and writes have no effect. |
| 0 | RCVR_ERR_MSK | Receiver Error Mask |
| **End of Table 3-176** | | |

### 3.8.8 PCI Express Advanced Error Capabilities and Control Register (PCIE_ACCR)

The PCI Express Advanced Error Capabilities and Control Register (PCIE_ACCR) is shown in Figure 3-169 and described in Table 3-177. The offset is 1118h.

**Figure 3-169    PCI Express Advanced Error Capabilities and Control Register (PCIE_ACCR)**

| 31                9 | 8 | 7 | 6 | 5 | 4                0 |
|---|---|---|---|---|---|
| Reserved | ECRC_CHK_EN | ECRC_CHK_CAP | ECRC_GEN_EN | ECRC_GEN_CAP | FRST_ERR_PTR |
| R-0 | R/W/S-0 | R-1 | R/W/S-0 | R-1 | R/S-0 |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-177    PCI Express Advanced Error Capabilities and Control Register (PCIE_ACCR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-9 | Reserved | Reads return 0 and writes have no effect. |
| 8 | ECRC_CHK_EN | ECRC Check Enable<br>  0 = ECRC check is disabled<br>  1 = ECRC check is enabled |
| 7 | ECRC_CHK_CAP | ECRC Check Capable |
| 6 | ECRC_GEN_EN | ECRC Generation Enable<br>  0 = ECRC generation is disabled<br>  1 = ECRC generation is enabled |
| 5 | ECRC_GEN_CAP | ECRC Generation Capability |
| 4-0 | FRST_ERR_PTR | First Error Pointer<br><br>The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register. |
| **End of Table 3-177** | | |

### 3.8.9 Header Log Register 0 (HDR_LOG0)

The Header Log Register 0 (HDR_LOG0) is shown in Figure 3-170 and described in Table 3-178. The offset is 111Ch.

**Figure 3-170    Header Log Register 0 (HDR_LOG0)**

| 31                                                    0 |
|---|
| HDR_DW0 |
| R/S-0 |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-178    Header Log Register 0 (HDR_LOG0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | HDR_DW0 | First DWORD of Header for a detected error. |
| **End of Table 3-178** | | |

### 3.8.10 Header Log Register 1 (HDR_LOG1)

The Header Log Register 1 (HDR_LOG1) is shown in Figure 3-171 and described in Table 3-179. The offset is 1120h.

**Figure 3-171    Header Log Register 1 (HDR_LOG1)**

| 31 | 0 |
|---|---|
| HDR_DW1 | |
| R/S-0 | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-179    Header Log Register 1 (HDR_LOG1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | HDR_DW1 | Second DWORD of header for a detected error. |
| **End of Table 3-179** | | |

### 3.8.11 Header Log Register 2 (HDR_LOG2)

The Header Log Register 2 (HDR_LOG2) is shown in Figure 3-172 and described in Table 3-180. The offset is 1124h.

**Figure 3-172    Header Log Register 2 (HDR_LOG2)**

| 31 | 0 |
|---|---|
| HDR_DW2 | |
| R/S-0 | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -*n* = value after reset

**Table 3-180    Header Log Register 2 (HDR_LOG2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | HDR_DW2 | Third DWORD of header for a detected error. |
| **End of Table 3-180** | | |

### 3.8.12 Header Log Register 3 (HDR_LOG3)

The Header Log Register 3 (HDR_LOG3) is shown in Figure 3-173 and described in Table 3-181. The offset is 1128h.

**Figure 3-173    Header Log Register 3 (HDR_LOG3)**

| 31 | 0 |
|---|---|
| HDR_DW3 | |
| R/S-0 | |

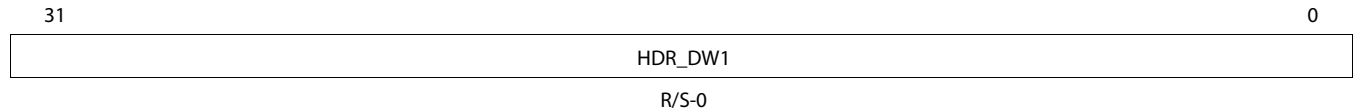Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -n = value after reset

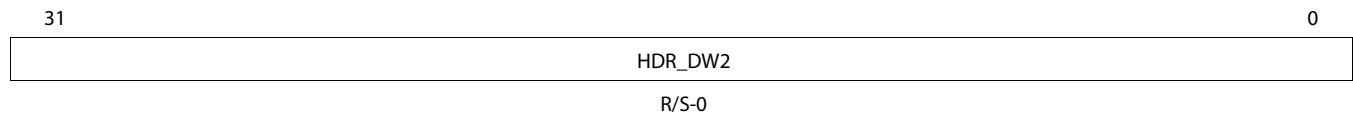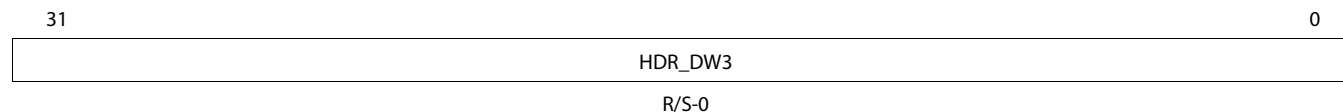**Table 3-181    Header Log Register 3 (HDR_LOG3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | HDR_DW3 | Fourth DWORD of Header for a detected error. |
| **End of Table 3-181** | | |

### 3.8.13 Root Error Command Register (ROOT_ERR_CMD)

The Root Error Command Register (ROOT_ERR_CMD) is shown in Figure 3-174 and described in Table 3-182. The offset is 112Ch.

**Figure 3-174    Root Error Command Register (ROOT_ERR_CMD)**

| 31 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | FERR_RPT_EN | NFERR_RPT_EN | CERR_RPT_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-182    Root Error Command Register (ROOT_ERR_CMD) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reads return 0 and writes have no effect. |
| 2 | FERR_RPT_EN | Fatal Error Reporting Enable.<br>0 = Error reporting is disabled<br>1 = Error reporting is enabled |
| 1 | NFERR_RPT_EN | Nonfatal Error Reporting Enable.<br>0 = Error reporting is disabled<br>1 = Error reporting is enabled |
| 0 | CERR_RPT_EN | Correctable Error Reporting Enable.<br>0 = Error reporting is disabled<br>1 = Error reporting is enabled |
| **End of Table 3-182** | | |

### 3.8.14  Root Error Status Register (ROOT_ERR_ST)

The Root Error Status Register (ROOT_ERR_ST) is shown in Figure 3-175 and described in Table 3-183. The offset is 1130h.

**Figure 3-175    Root Error Status Register (ROOT_ERR_ST)**

| 31 | | 27 | 26 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| AER_INT_MSG | | | Reserved | | | | | | |
| R-0 | | | R-0 | | | | | | |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | FERR_RCV | NFERR | UNCOR_FATAL | MULT_FNF | ERR_FNF | MULT_COR | CORR_ERR |
| R-0 | | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 | R/W1C/S-0 |

Legend: R = Read only; W1C = Write 1 to clear, write of 0 has no effect; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-183    Root Error Status Register (ROOT_ERR_ST) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-27 | AER_INT_MSG | AER Interrupt Message Number. Writable through internal bus interface. |
| 26-7 | Reserved | Reads return 0 and writes have no effect. |
| 6 | FERR_RCV | Fatal Error Messages Received. |
| 5 | NFERR | Non-Fatal Error Messages Received. |
| 4 | UNCOR_FATAL | First Uncorrectable Fatal Received. |
| 3 | MULT_FNF | Multiple Uncorrectable Error (ERR_FATAL/NONFATAL) Received. |
| 2 | ERR_FNF | Uncorrectable Error (ERR_FATAL/NONFATAL) Received. |
| 1 | MULT_COR | Multiple Correctable Error (ERR_COR) Received. |
| 0 | CORR_ERR | Correctable Error (ERR_COR) Received. |
| **End of Table 3-183** | | |

### 3.8.15  Error Source Identification Register (ERR_SRC_ID)

The Error Source Identification Register (ERR_SRC_ID) is shown in Figure 3-141 and described in Table 3-145. The offset is 1134h.

**Figure 3-176    IO Base and Limit Upper 16 bits Register (IOSPACE)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| FNF_SRC_ID | | CORR_SRC_ID | |
| R/S-0 | | R/S-0 | |

Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-184    IO Base and Limit Upper 16 bits Register (IOSPACE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-16 | FNF_SRC_ID | Fatal or Non-Fatal error source identification |
| 15-0 | CORR_SRC_ID | Correctable error source identification |
| **End of Table 3-184** | | |

## 3.9 Port Logic Registers

Please see the device-specific data manual and PCIe Address Space section for the base address details.

### 3.9.1 Register Summary

**Table 3-185    Port Logic Registers**

| Offset[1] | Acronym | Register Description | Section |
|-----------|---------|---------------------|---------|
| 1700h | PL_ACKTIMER | Ack Latency Time and Replay Timer | Section 3.9.2 |
| 1704h | PL_OMSG | Other Message Register | Section 3.9.3 |
| 1708h | PL_FORCE_LINK | Port Force Link Register | Section 3.9.4 |
| 170Ch | ACK_FREQ | Ack Frequency Register | Section 3.9.5 |
| 1710h | PL_LINK_CTRL | Port Link Control Register | Section 3.9.6 |
| 1714h | LANE_SKEW | Lane Skew Register | Section 3.9.7 |
| 1718h | SYM_NUM | Symbol Number Register | Section 3.9.8 |
| 171Ch | SYMTIMER_FLTMASK | Symbol Timer and Filter Mask Register | Section 3.9.9 |
| 1720h | FLT_MASK2 | Filter Mask Two register | Section 3.9.10 |
| 1728h | DEBUG0 | Debug 0 register | Section 3.9.11 |
| 172Ch | DEBUG1 | Debug 1 register | Section 3.9.12 |
| 180Ch | PL_GEN2 | Gen2 Register | Section 3.9.13 |
| **End of Table 3-185** | | | |

1. The actual addresses of these registers are device specific. See the device-specific data manual to verify the register addresses

### 3.9.2 Ack Latency Time and Replay Timer (PL_ACKTIMER)

The Ack Latency Time and Replay Timer (PL_ACKTIMER) is shown in Figure 3-177 and described in Table 3-186. The offset is 1700h.

**Figure 3-177    Ack Latency Time and Replay Timer (PL_ACKTIMER)**
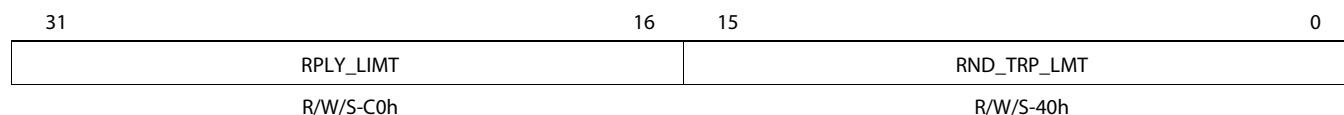
| 31 | 16 | 15 | 0 |
|----|----|----|---|
| RPLY_LIMT | | RND_TRP_LMT | |
| R/W/S-C0h | | R/W/S-40h | |

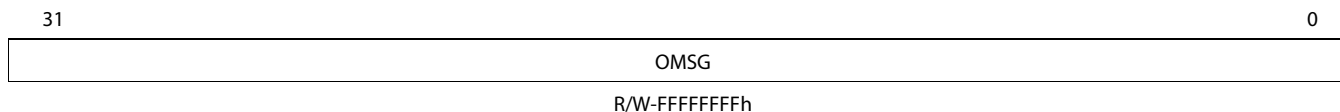Legend: R = Read only; W = Write only; S = Sticky, a hot reset will not clear this field; -n = value after reset

**Table 3-186    Ack Latency Time and Replay Timer (PL_ACKTIMER) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-16 | RPLY_LIMT | Replay Time Limit. The replay timer expires when it reaches this limit. |
| 15-0 | RND_TRP_LMT | Round Trip Latency Time Limit. The Ack/Nak latency timer expires when it reaches this limit. |
| **End of Table 3-186** | | |

### 3.9.3 Other Message Register (PL_OMSG)

The Other Message Register (PL_OMSG) is shown in Figure 3-178 and described in Table 3-187. The offset is 1704h.

**Figure 3-178    Other Message Register (PL_OMSG)**

| 31 | 0 |
|---|---|
| OMSG | |

R/W-FFFFFFFFh

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

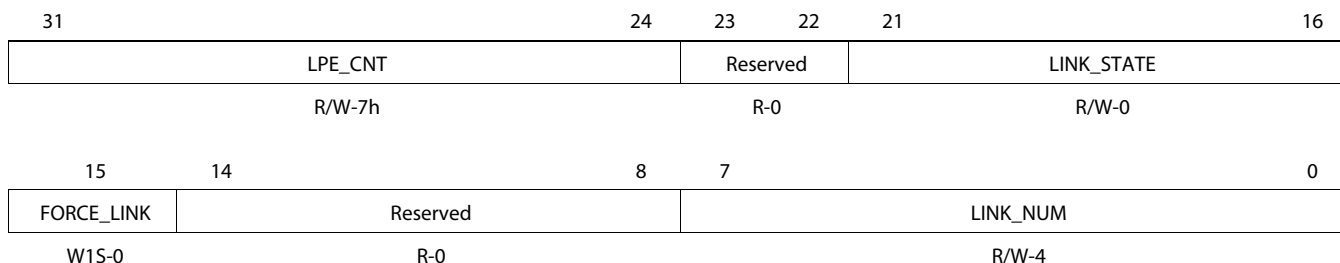**Table 3-187    Other Message Register (PL_OMSG) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | OMSG | Other Message Register. It can be used to send a specific PCI Express message in which case this register is programmed with the payload and bit 0 of Port Link Control Register is set to transmit the message. |
| **End of Table 3-187** | | |

### 3.9.4 Port Force Link Register (PL_FORCE_LINK)

The Port Force Link Register (PL_FORCE_LINK) is shown in Figure 3-179 and described in Table 3-188. The offset is 1708h.

**Figure 3-179    Port Force Link Register (PL_FORCE_LINK)**

| 31 | 24 | 23 | 22 | 21 | 16 |
|---|---|---|---|---|---|
| LPE_CNT | | Reserved | | LINK_STATE | |
| R/W-7h | | R-0 | | R/W-0 | |

| 15 | 14 | 8 | 7 | 0 |
|---|---|---|---|---|
| FORCE_LINK | Reserved | | LINK_NUM | |
| W1S-0 | R-0 | | R/W-4 | |

Legend: R = Read only; W = Write only; W1S = Write 1 to set, write of 0 has no effect; -n = value after reset

**Table 3-188    Port Force Link Register (PL_FORCE_LINK) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-24 | LPE_CNT | Low Power Entrance Count |
| 23-22 | Reserved | Reads return 0 and writes have no effect. |
| 21-16 | LINK_STATE | Link State. The link state that the PCIe will be forced to when FORCE_LINK field is set. Please see Appendix A.1 for LTSSM states encoded values. |
| 15 | FORCE_LINK | Force Link. Forces the link to the state specified by the LINK_STATE field. The Force Link pulse will trigger link re-negotiation. |
| 14-8 | Reserved | Reads return 0 and writes have no effect. |
| 7-0 | LINK_NUM | Link Number. Not used for EP. |
| **End of Table 3-188** | | |

### 3.9.5 Ack Frequency Register (ACK_FREQ)

The Ack Frequency Register (ACK_FREQ) is shown in Figure 3-180 and described in Table 3-189. The offset is 170Ch.

**Figure 3-180 Ack Frequency Register (ACK_FREQ)**

| 31 | 30 | 29 | 27 | 26 | 24 | 23 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | ASPM_L1 | L1_ENTRY_LATENCY | | L0S_ENTRY_LATENCY | | COMM_NFTS | |
| R-0 | R/W-0 | R/W-3h | | R/W-3h | | R/W-Fh | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| NFTS | | ACK_FREQ | |
| R/W-64h | | R/W-0 | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-189 Ack Frequency Register (ACK_FREQ) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | Reserved | Reads return 0 and writes have no effect. |
| 30 | ASPM_L1 | Set to allow entering ASPM L1 even when link partner did not to L0s. When cleared, the ASPM L1 state is entered only after idle period during which both RX and TX are in L0s. |
| 29-27 | L1_ENTRY_LATENCY | L1 entrance latency. The latency is set to 2^L1_ENTRY_LATENCY microseconds with the max being 64 microseconds.<br>0 = 1 µs<br>1h = 2 µs<br>2h = 4 µs<br>3h = 8 µs<br>4h = 16 µs<br>5h = 32 µs<br>6h or 7h = 64 µs |
| 26-24 | L0S_ENTRY_LATENCY | L0s entrance latency. The latency is set to L0S_ENTRY_LATENCY+1 microseconds. Maximum is 7 microseconds.<br>0 = 1 µs<br>1h = 2 µs<br>2h = 3 µs<br>3h = 4 µs<br>4h = 5 µs<br>5h = 6 µs<br>6h or 7h = 7 µs |
| 23-16 | COMM_NFTS | Number of fast training sequences when common clock is used and when transitioning from L0s to L0. |
| 15-8 | NFTS | Number of fast training sequences to be transmitted when transitioning from L0s to L0. Value of 0 is not supported. |
| 7-0 | ACK_FREQ | Ack Frequency. The value range is 0x0-0xFF. A value of 0 in the ACK Frequency field indicates that this ACK frequency control feature is disabled. A value of N (N>0) indicates that the module will ack N TLPs received by sending an ACK DLLP. |
| **End of Table 3-189** | | |

### 3.9.6 Port Link Control Register (PL_LINK_CTRL)

The Port Link Control Register (PL_LINK_CTRL) is shown in Figure 3-181 and described in Table 3-190. The offset is 1710h.

**Figure 3-181    Port Link Control Register (PL_LINK_CTRL)**

| 31 | 22 | 21 | 16 | 15 | 12 | 11 | 8 |
|----|----|----|----|----|----|----|----|
| Reserved | | LNK_MODE | | Reserved | | LINK_RATE | |
| R-0 | | R/W-3h | | R-0 | | R/W-1 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|
| FLNK_MODE | Reserved | DLL_EN | Reserved | RST_ASRT | LPBK_EN | SCRM_DIS | OMSG_REQ |
| R/W-0 | R-0 | R/W-1 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-190    Port Link Control Register (PL_LINK_CTRL) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-22 | Reserved | Reads return 0 and writes have no effect. |
| 21-16 | LNK_MODE | Link Mode Enable. (×N – corresponding to N Lanes).<br>1h = ×1<br>3h = ×2<br>7h = ×4<br>Fh = ×8<br>1Fh = ×16<br>3Fh = ×32<br>Others = Reserved |
| 15-12 | Reserved | Reads return 0 and writes have no effect. |
| 11-8 | LINK_RATE | Default link rate. For 2.5 GT/s it is 0x1. This register does not affect any functionality. |
| 7 | FLNK_MODE | Fast link mode. Set all internal timers to fast mode for simulation purposes. |
| 6 | Reserved | Reads return 0 and writes have no effect. |
| 5 | DLL_EN | DLL Link Enable. Enable link initialization. |
| 4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | RST_ASRT | Reset assert. Triggers a recovery and forces the LTSSM to the Hot Reset state. Downstream ports (RC ports) only. |
| 2 | LPBK_EN | Loopback Enable. Turn on loopback. |
| 1 | SCRM_DIS | Scramble Disable. Turn off data scrambling. |
| 0 | OMSG_REQ | Other message request. Set to transmit the message contained in the Other Message Register (PL_OMSG). |
| **End of Table 3-190** | | |

### 3.9.7 Lane Skew Register (LANE_SKEW)

The Lane Skew Register (LANE_SKEW) is shown in Figure 3-182 and described in Table 3-191. The offset is 1714h.

**Figure 3-182    Lane Skew Register (LANE_SKEW)**

| 31 | 30          26 | 25 | 24 | 23                          0 |
|----|------|------|------|------|
| L2L_DESKEW | Reserved | ACK_DISABLE | FC_DISABLE | LANE_SKEW |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-191    Lane Skew Register (LANE_SKEW) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31 | L2L_DESKEW | Disable Lane to Lane Deskew. |
| 30-26 | Reserved | Reads return 0 and writes have no effect. |
| 25 | ACK_DISABLE | Disable Ack and Nak DLLP transmission. |
| 24 | FC_DISABLE | Flow Control Disable. Set to disable transmission of Flow Control DLLPs. |
| 23-0 | LANE_SKEW | Insert Lane Skew for Transmit. The value is in units of one symbol time. Thus a value 0x02 will force a skew of two symbol times for that lane. Max allowed is 5 symbol times. This 24 bit field is used for programming skew for eight lanes with three bits per lane. |
| **End of Table 3-191** | | |

### 3.9.8  Symbol Number Register (SYM_NUM)

The Symbol Number Register (SYM_NUM) is shown in Figure 3-183 and described in Table 3-192. The offset is 1718h.

**Figure 3-183  Symbol Number Register (SYM_NUM)**

| 31 | 29 | 28 | | 24 | 23 | | 19 | 18 | |
|----|----|----|----|----|----|----|----|----|----|
| MAX_FUNC | | FCWATCH_TIMER | | | ACK_LATENCY_TIMER | | | REPLAY_TIMER | |
| R/W-0 | | R/W-0 | | | R/W-0 | | | R/W-4h | |

| 14 | 13 | | 11 | 10 | | 8 | 7 | | 4 | 3 | | 0 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| REPLAY_TIMER | | Reserved | | SKP_COUNT | | | NUM_TS2_SYMBOLS | | | TS_COUNT | | |
| R/W-4h | | R-0 | | R/W-3h | | | R/W-Ah | | | R/W-Ah | | |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-192  Symbol Number Register (SYM_NUM) Field Descriptions**

| Bit | Field | Description |
|-----|-------|-------------|
| 31-29 | MAX_FUNC | Configuration requests targeted at function numbers above this value will result in UR response. |
| 28-24 | FCWATCH_TIMER | Timer Modifier for Flow Control Watchdog Timer. Increases the timer value for Flow Control watchdog timer in increments of 16 clock cycles. |
| 23-19 | ACK_LATENCY_TIMER | Timer Modifier for Ack/Nak Latency Timer. Increases the timer value for the Ack/Nak latency timer in increments of 64 clock cycles. |
| 18-14 | REPLAY_TIMER | Timer Modifier for Replay Timer. Increases the timer value for the replay timer in increments of 64 clock cycles. |
| 13-11 | Reserved | Reads return 0 and writes have no effect. |
| 10-8 | SKP_COUNT | Number of SKP Symbols. |
| 7-4 | NUM_TS2_SYMBOLS | Number of TS2 Symbols. This field does not affect any functionality. |
| 3-0 | TS_COUNT | Number of TS Symbols. Set the number of TS identifier symbols that are sent in TS1 and TS2 ordered sets. |
| **End of Table 3-192** | | |

### 3.9.9 Symbol Timer and Filter Mask Register (SYMTIMER_FLTMASK)

The Symbol Timer and Filter Mask Register (SYMTIMER_FLTMASK) is shown in Figure 3-184 and described in Table 3-193. The offset is 171Ch.

**Figure 3-184    Symbol Timer and Filter Mask Register (SYMTIMER_FLTMASK)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|----|----|----|----|----|----|----|----|
| F1_CFG_DROP | F1_IO_DROP | F1_MSG_DROP | F1_CPL_ECRC_DROP | F1_ECRC_DROP | F1_CPL_LEN_TEST | F1_CPL_ATTR_TEST | F1_CPL_TC_TEST |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|----|----|----|----|----|----|----|----|
| F1_CPL_FUNC_TEST | F1_CPL_REQID_TEST | F1_CPL_TAGERR_TEST | F1_LOCKED_RD_AS_UR | F1_CFG1_RE_AS_US | F1_UR_OUT_OF_BAR | F1_UR_POISON | F1_UR_FUN_MISMATCH |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14    11 | 10                                0 |
|----|----------|-------------------------------------|
| FC_WDOG_DISABLE | Reserved | SKP_VALUE |
| R/W-0 | R-0 | R/W-500h |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-193    Symbol Timer and Filter Mask Register (SYMTIMER_FLTMASK) Field Descriptions  (Part 1 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 31 | F1_CFG_DROP | Set to allow CFG TLPs on RC<br>0 = Not allow CFG transaction being received on RC.<br>1 = Allow CFG transaction being received on RC. |
| 30 | F1_IO_DROP | Set to allow IO TLPs on RC<br>0 = Not allow IO transaction being received on RC.<br>1 = Allow IO transaction being received on RC. |
| 29 | F1_MSG_DROP | Set to allow MSG TLPs on RC<br>0 = Not allow MSG transaction being received on RC.<br>1 = Allow MSG transaction being received on RC. |
| 28 | F1_CPL_ECRC_DROP | Set to allow Completion TLPs with ECRC to pass up<br>0 = Discard completion TLPs with ECRC errors.<br>1 = Allow completion TLPs with ECRC errors to be passed up. |
| 27 | F1_ECRC_DROP | Set to allow TLPs with ECRC error to pass up<br>0 = Discard TLPs with ECRC errors.<br>1 = Allow TLPs with ECRC errors to be passed up. |
| 26 | F1_CPL_LEN_TEST | Set to mask length match for received completion TLPs<br>0 = Enforce length match for received completion TLPs.<br>1 = Mask length match for received completion TLPs. |
| 25 | F1_CPL_ATTR_TEST | Set to mask attribute match on received completion TLPs<br>0 = Enforce attribute match for received completion TLPs.<br>1 = Mask attribute match on received completion TLPs. |
| 24 | F1_CPL_TC_TEST | Set to mask traffic match on received completion TLPs<br>0 = Enforce traffic class match for received completion TLPs.<br>1 = Mask traffic class match on received completion TLPs. |
| 23 | F1_CPL_FUNC_TEST | Set to mask function match on received completion TLPs<br>0 = Enforce function match for received completion TLPs.<br>1 = Mask function match for received completion TLPs. |

**Table 3-193    Symbol Timer and Filter Mask Register (SYMTIMER_FLTMASK) Field Descriptions  (Part 2 of 2)**

| Bit | Field | Description |
|-----|-------|-------------|
| 22 | F1_CPL_REQID_TEST | Set to mask request ID match on received completion TLPs<br>0 = Enforce request ID match for received completion TLPs.<br>1 = Mask request ID match for received completion TLPs. |
| 21 | F1_CPL_TAGERR_TEST | Set to mask tag error rules for received completion TLPs<br>0 = Enforce tag error rules for received completion TLPs.<br>1 = Mask tag error rules for received completion TLPs. |
| 20 | F1_LOCKED_RD_AS_UR | Set to treat locked read TLPs as supported for EP, UR for RC<br>0 = Treat locked read TLPs as UR for EP, supported for RC.<br>1 = Treat locked read TLPs as supported for EP, UR for RC. |
| 19 | F1_CFG1_RE_AS_US | Set to treat type 1 CFG TLPs as supported for EP, UR for RC<br>0 = Treat type 1 CFG TLPs as UR for EP and supported for RC.<br>1 = Treat type 1 CFG TLPs as supported for EP and UR for RC. |
| 18 | F1_UR_OUT_OF_BAR | Set to treat out-of-BAR TLPs as supported requests<br>0 = Treat out-of-BAR TLPs as UR.<br>1 = Treat out-of-BAR TLPs as supported requests. |
| 17 | F1_UR_POISON | Set to treat poisoned TLPs as supported requests<br>0 = Treat poisoned TLPs as UR.<br>1 = Treat poisoned TLPs as supported requests. |
| 16 | F1_UR_FUN_MISMATCH | Set to treat function mismatched TLPs as supported requests<br>0 = Treat function mismatched TLPs as UR.<br>1 = Treat function mismatched TLPs as supported requests. |
| 15 | FC_WDOG_DISABLE | Set to disable FC watchdog timer<br>0 = Enable Flow Control watchdog timer.<br>1 = Disable Flow Control watchdog timer. |
| 14-11 | Reserved | Reads return 0 and writes have no effect. |
| 10-0 | SKP_VALUE | Number of symbol times to wait between transmitting SKP ordered sets. For example, for a setting of 1536 decimal, the wait will be for 1537 symbol times. |
| **End of Table 3-193** | | |

### 3.9.10 Filter Mask Register 2 (FLT_MASK2)

The Filter Mask Register 2 (FLT_MASK2) is shown in Figure 3-185 and described in Table 3-194. The offset is 1720h.

**Figure 3-185    Filter Mask Register 2 (FLT_MASK2)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | FLUSH_REQ | DLLP_ABORT | VMSG1_DROP | VMSG0_DROP |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

Legend: R = Read only; W = Write only; *-n* = value after reset

**Table 3-194    Filter Mask Register 2 (FLT_MASK2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reads return 0 and writes have no effect. |
| 3 | FLUSH_REQ | Set to enable the filter to handle flush request<br>0 = Disable the filter to handle flush request.<br>1 = Enable the filter to handle flush request. |
| 2 | DLLP_ABORT | Set to disable DLLP abort for unexpected CPL<br>0 = Enable DLLP abort for unexpected CPL.<br>1 = Disable DLLP abort for unexpected CPL. |
| 1 | VMSG1_DROP | Set to disable dropping of Vendor MSG Type 1<br>0 = Enable dropping of Vendor MSG Type 1. It will be passed to internal bus interface.<br>1 = Disable dropping of Vendor MSG Type 1. |
| 0 | VMSG0_DROP | Set to disable dropping of Vendor MSG Type 0 with UR reporting<br>0 = Enable dropping of Vendor MSG Type 0 with UR error reporting. It will be passed to internal bus interface.<br>1 = Disable dropping of Vendor MSG Type 0 with UR error reporting. |
| **End of Table 3-194** | | |

### 3.9.11  Debug 0 Register (DEBUG0)

The Debug 0 Register (DEBUG0) is shown in Figure 3-186 and described in Table 3-195. The offset is 1728h.

**Figure 3-186    Debug 0 Register (DEBUG0)**

| 31      28 | 27 | 26 | 25 | 24 | 23                              8 |
|---|---|---|---|---|---|
| TS_LINK_CTRL | TS_LANE_K237 | TS_LINK_K237 | RCVD_IDLE0 | RCVD_IDLE1 | PIPE_TXDATA |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

| 7 | 6 | 5 | 4                                                    0 |
|---|---|---|---|
| PIPE_TXDATAK | TXB_SKIP_TX | LTSSM_STATE |
| R-0 | R-0 | R-0 |

Legend: R = Read only; W = Write only; -n = value after reset

**Table 3-195    Debug 0 Register (DEBUG0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-28 | TS_LINK_CTRL | Link control bits advertised by link partner. |
| 27 | TS_LANE_K237 | Currently receiving k237 (PAD) in place of lane number. |
| 26 | TS_LINK_K237 | Currently receiving k237 (PAD) in place of link number. |
| 25 | RCVD_IDLE0 | Receiver is receiving logical idle. |
| 24 | RCVD_IDLE1 | 2nd symbol is also idle (16bit PHY interface only). |
| 23-8 | PIPE_TXDATA | PIPE Transmit data. Reset value is zero but changes at every clock after that. |
| 7-6 | PIPE_TXDATAK | PIPE transmit K indication. |
| 5 | TXB_SKIP_TX | A skip ordered set has been transmitted. |
| 4-0 | LTSSM_STATE | LTSSM current state. Please see Appendix A.1 for the names of the LTSSM states corresponding to the encoded values. |
| **End of Table 3-195** | | |

### 3.9.12  Debug 1 Register (DEBUG1)

The Debug 1 Register (DEBUG1) is shown in Figure 3-187 and described in Table 3-196. The offset is 172Ch.

**Figure 3-187    Debug 1 Register (DEBUG1)**

| 31 | 30 | 29 | 28 | 27 | 26 |
|---|---|---|---|---|---|
| SCRAMBLER_DISABLE | LINK_DISABLE | LINK_IN_TRAINING | RCVR_REVRS_POL_EN | TRAINING_RST_N | Reserved |
| R-0 | R-0 | R-0 | R-0 | R-1 | R-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|
| Reserved | PIPE_TXDETECTRX_LB | PIPE_TXCOMPLIANCE | PIPE_TXCOMPLIANCE | APP_INIT_RST | Reserved | |
| R-0 | R-0 | R-1 | R-0 | R-0 | R-0 | |

| 15 | 8 | 7 | 5 |
|---|---|---|---|
| RMLH_TS_LINK_NUM | | Reserved | |
| R-0 | | R-0 | |

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| XMLH_LINK_UP | RMLH_INSKIP_RCV | RMLH_TS1_RCVD | RMLH_TS2_RCVD | RMLH_RCVD_LANE_REV |
| R-0 | R-0 | R-0 | R-0 | R-0 |

Legend: R = Read only; W = Write only; -*n* = value after reset

**Table 3-196    Debug 1 Register (DEBUG1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31 | SCRAMBLER_DISABLE | Scrambling disabled for the link. |
| 30 | LINK_DISABLE | LTSSM in DISABLE state. Link inoperable. |
| 29 | LINK_IN_TRAINING | LTSSM performing link training. |
| 28 | RCVR_REVRS_POL_EN | LTSSM testing for polarity reversal. |
| 27 | TRAINING_RST_N | LTSSM-negotiated link reset. |
| 26-23 | Reserved | Reads return 0 and writes have no effect. |
| 22 | PIPE_TXDETECTRX_LB | PIPE receiver detect/loopback request. |
| 21 | PIPE_TXELECIDLE | PIPE transmit electrical idle request. |
| 20 | PIPE_TXCOMPLIANCE | PIPE transmit compliance request. |
| 19 | APP_INIT_RST | Application request to initiate training reset. |
| 18-16 | Reserved | Reads return 0 and writes have no effect. |
| 15-8 | RMLH_TS_LINK_NUM | Link number advertised/confirmed by link partner. |
| 7-5 | Reserved | Reads return 0 and writes have no effect. |
| 4 | XMLH_LINK_UP | LTSSM reports PHY link up. |
| 3 | RMLH_INSKIP_RCV | Receiver reports skip reception. |
| 2 | RMLH_TS1_RCVD | TS1 training sequence received (pulse). |
| 1 | RMLH_TS2_RCVD | TS2 training sequence received (pulse). |
| 0 | RMLH_RCVD_LANE_REV | Receiver detected lane reversal. |
| **End of Table 3-196** | | |

### 3.9.13 Gen2 Register (PL_GEN2)

The Gen2 Register (PL_GEN2) is shown in Figure 3-188 and described in Table 3-197. The offset is 180Ch.

**Figure 3-188    Gen2 Register (PL_GEN2)**

| 31 | | | | | | | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | DEEMPH | CFG_TX_CMPL | CFG_TX_SWING | DIR_SPD | LN_EN |
| R-0 | | | | | | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-2h |

| | 8 | 7 | | 0 |
|---|---|---|---|---|
| LN_EN | | NUM_FTS | | |
| R/W-2h | | R/W-Fh | | |

Legend: R = Read only; W = Write only; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

**Table 3-197    Gen2 Register (PL_GEN2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-21 | Reserved | Reads return 0 and writes have no effect. |
| 20 | DEEMPH | Set de-emphasis level for upstream ports (EP ports). |
| 19 | CFG_TX_CMPL | Configure TX compliance receive bit. When set to 1, signals LTSSM to transmit TS ordered sets with the compliance receive bit assert (equal to 1). |
| 18 | CFG_TX_SWING | Configure PHY TX Swing. Indicates the voltage level the PHY should drive.<br>0 = Full Swing<br>1 = Low Swing |
| 17 | DIR_SPD | Directed Speed Change.<br>0 = Indicates to the LTSSM not to initiate a speed change to Gen2 after the link is initialized at Gen1 speed.<br>1 = Indicates to the LTSSM to initiate a speed change to Gen2 after the link is initialized at Gen1 speed. |
| 16-8 | LN_EN | Lane Enable.<br>1h = ×1<br>2h = ×2<br>Others = Reserved. |
| 7-0 | NUM_FTS | Number of fast training sequences. |
| **End of Table 3-197** | | |

# Encoding of LTSSM State in DEBUG Registers

This appendix contains the following supplementary information:

# A.1 LTSSM States

The LTSSM state value that is read from DEBUG registers is an encoded value. The literal names of the LTSSM states corresponding to the encoded values are tabulated below.

**Table A-1      Encoding of LTSSM State in DEBUG registers**

| Code | LTSSM State |
|------|-------------|
| 0x00 | DETECT_QUIET |
| 0x01 | DETECT_ACT |
| 0x02 | POLL_ACTIVE |
| 0x03 | POLL_COMPLIANCE |
| 0x04 | POLL_CONFIG |
| 0x05 | PRE_DETECT_QUIET |
| 0x06 | DETECT_WAIT |
| 0x07 | CFG_LINKWD_START |
| 0x08 | CFG_LINKWD_ACEPT |
| 0x09 | CFG_LANENUM_WAIT |
| 0x0A | CFG_LANENUM_ACEPT |
| 0x0B | CFG_COMPLETE |
| 0x0C | CFG_IDLE |
| 0x0D | RCVRY_LOCK |
| 0x0E | RCVRY_SPEED |
| 0x0F | RCVRY_RCVRCFG |
| 0x10 | RCVRY_IDLE |
| 0x11 | L0 |
| 0x12 | L0s |
| 0x13 | L123_SEND_EIDLE |
| 0x14 | L1_IDLE |
| 0x15 | L2_IDLE |
| 0x16 | L2_WAKE |
| 0x17 | DISABLED_ENTRY |
| 0x18 | DISABLED_IDLE |
| 0x19 | DISABLED |
| 0x1A | LPBK_ENTRY |
| 0x1B | LPBK_ACTIVE |
| 0x1C | LPBK_EXIT |
| 0x1D | LPBK_EXIT_TIMEOUT |
| 0x1E | HOT_RESET_ENTRY |
| 0x1F | HOT_RESET |
| **End of Table A-1** | |

# IMPORTANT NOTICE

| Products | | Applications | |
|---|---|---|---|
| Audio | www.ti.com/audio | Automotive and Transportation | www.ti.com/automotive |
| Amplifiers | amplifier.ti.com | Communications and Telecom | www.ti.com/communications |
| Data Converters | dataconverter.ti.com | Computers and Peripherals | www.ti.com/computers |
| DLP® Products | www.dlp.com | Consumer Electronics | www.ti.com/consumer-apps |
| DSP | dsp.ti.com | Energy and Lighting | www.ti.com/energy |
| Clocks and Timers | www.ti.com/clocks | Industrial | www.ti.com/industrial |
| Interface | interface.ti.com | Medical | www.ti.com/medical |
| Logic | logic.ti.com | Security | www.ti.com/security |
| Power Mgmt | power.ti.com | Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Microcontrollers | microcontroller.ti.com | Video and Imaging | www.ti.com/video |
| RFID | www.ti-rfid.com | | |
| OMAP Applications Processors | www.ti.com/omap | **TI E2E Community** | e2e.ti.com |
| Wireless Connectivity | www.ti.com/wirelessconnectivity | | |