Power Systems Host Bridge 4 (PHB4) Specification

OpenPOWER

Version 1.0
27 July 2018

© Copyright International Business Machines Corporation 2018

Printed in the United States of America July 2018

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

The OpenPOWER word mark and the OpenPOWER Logo mark, and related marks, are trademarks and service marks licensed by OpenPOWER.

Other company, product, and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

This document is intended for the development of technology products compatible with Power Architecture®. You may use this document, for any purpose (commercial or personal) and make modifications and distribute; however, modifications to this document may violate Power Architecture and should be carefully considered. Any distribution of this document or its derivative works shall include this Notice page including but not limited to the IBM warranty disclaimer and IBM liability limitation. No other licenses (including patent licenses), expressed or implied, by estoppel or otherwise, to any intellectual property rights are granted by this document.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN “AS IS” BASIS. IBM makes no representations or warranties, either express or implied, including but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement, or that any practice or implementation of the IBM documentation will not infringe any third party patents, copyrights, trade secrets, or other rights. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

IBM Systems
294 Route 100, Building SOM4
Somers, NY 10589-3216

The IBM home page can be found at **ibm.com®.**

Version 1.0
27 July 2018

Contents

List of Figures

List of Tables

Revision Log

Each release of this document supersedes all previously released versions. The revision log lists all significant changes made to the document since its initial release. In the rest of the document, change bars in the margin indicate that the adjacent text was modified from the previous release of this document.

Revision Date	Description
27 July 2018	Version 1.0. Initial release.

About this Document

This document describes the design and operation of the IBM® Power Systems™ Host Bridge 4 (PHB4) building block. It provides information on the requirements, function, operation, and usage.

Who Should Read this Document

This document is intended for designers, simulators, testers, and programmers who must understand the requirements, function, and software interfaces of the PHB4.

Related Documents

The documents available in OpenPOWER Connect, an online IBM technical library, are helpful in understanding the PHB4. Additional technical resources are available through the OpenPOWER Foundation (http://openpowerfoundation.org/technical/technical-resources/).

Terminology Used in this Document

See the *Glossary* on page 311 for the terms and acronyms used throughout this document.

Conventions Used in This Document

This section explains the numbering conventions and other conventions that are in this document.

Typographical Conventions

The following typographical conventions are used to define special terms and command syntax:

Convention	Description
Underline	Indicates that the definition of an acronym is displayed when the user hovers the cursor over the term.
Hyperlink	Web-based URLs are displayed in blue text to denote a virtual link to an external location. For example: http://www.ibm.com
Note: This is note text.	The note block denotes information that emphasizes a concept or provides critical information.

Representation of Numbers

Numbers are generally shown in decimal format, unless designated as follows:

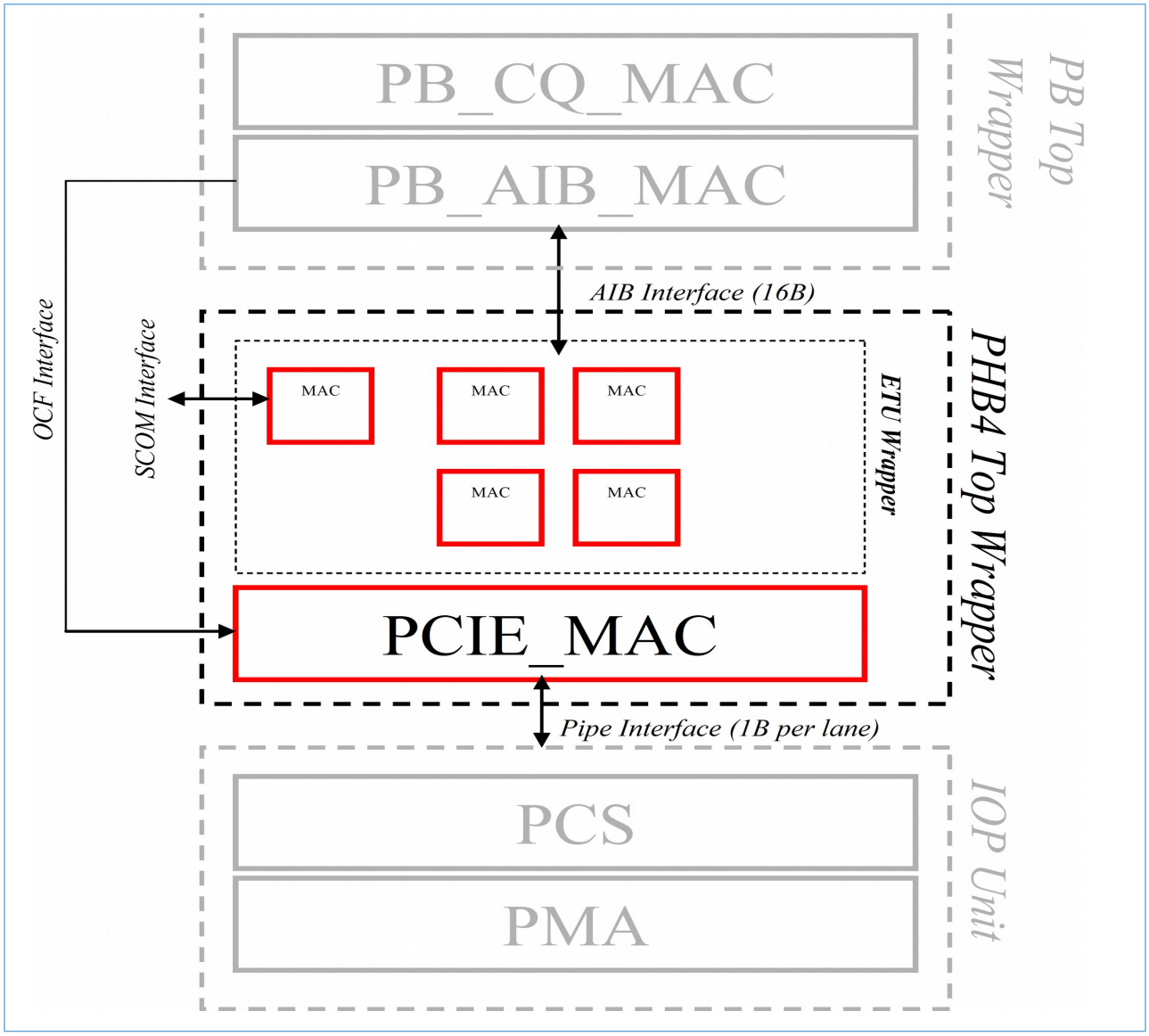
- Hexadecimal values are preceded by a 0x.
  For example: 0x0A00.
- Binary values in sentences are shown in single quotation marks.
  For example: ‘1010’.

1. Introduction

The IBM Power Systems Host Bridge 4 (PHB4) is a reusable building block that implements a single PCI-Express root complex port and will be used in the IBM® Power Systems™ POWER9™ microprocessor. The PHB4 provides a PCIe Gen 4 root complex port to connect to an adapter slot, a link to a PCIe switch, or a PCIe cable connection. The block is part of a larger unit called PCI Express controller (PEC) that also contains a PB block that connects to the Power Bus in the processor.

Figure 1-1 shows the top level interfaces for the PHB4.

Figure 1-1. PHB4 Top Level Interfaces



2. Requirements

This section describes the requirements for successful implementation of the PHB4. These requirements help to define the list of work items required for correct design implementation. Any changes or additions to these requirements will be done using a formal change control process and approval.

The requirements are separated into two sub-sections by the internal macros, ETU and PCIe, followed by the common requirements.

2.1 ETU Specific Requirements

This section covers the requirements specifically for the ETU LBS macros. The ETU macros implement most of the Power Systems architecture requirements. In this implementation the ETU is composed of several physical macros. The requirements will be described as if the ETU is a single macro.

Note there are two ETU top level macros/wrappers for each physical PCIe link width, a x08 (ETUX08) and a x16 (ETUX16).

2.1.1 Features

Table 2-1 lists the PHB4 ETU features.

Table 2-1. PHB4 ETU Features (Page 1 of 2)

Feature	PHB4 ETUX08	PHB4 ETUX16	Comments
AIB Bus, Datapath Width	16 bytes	16 bytes	
AIB Bus, BW in each direction	32GB/s	32GB/s	
BLIF interface width	16 bytes	16 bytes	
BLIF interface BW in each direction	32GB/s	32GB/s	
Target BW in each direction	14GB/s (87.5% eff)	28GB/s (87.5% eff)	N% of theoretical link BW. Note: PHB4, ETUX08 can only achieve 14GB/s even though the physical buses are capable of 32GB/s since it is attached to a PCIEX08 link.
ETU Base Frequency	2GHz	2GHz	Same as PCIe BCLK.
Technology	IBM, 14nm	IBM, 14nm	
IBM IODA architecture version supported	2+	2+	IODA2+ has significant changes for interrupts.
Number of physical address bits supported	56	56	This is also the maximum number of address bits sent across the AIB bus.
Partitionable Endpoints (PEs), number supported	256	512	This number affects the number of EEH error states and also the sizes of tables based on the IODA architecture.
LSI interrupts, number supported	8	8	
MSI interrupts, number supported	2K	4K	
IVC Cache Depth, number of entries	0	0	IODA2+ changes the interrupt structure for the source controller. There is no IVC cache in the PHB4.
Interrupt State Table (IST) Size, number of interrupts	2K	4K	The PHB4, holds the interrupt state bits and PE number for each interrupt source number in an internal SRAM.

Table 2-1. PHB4 ETU Features (Page 2 of 2)

Feature	PHB4 ETUX08	PHB4 ETUX16	Comments
LXIVT/MXIVT Tables in hardware	No	No	IODA2+ changes the interrupt structure for the source controller. These tables are no longer present in the PHB4.
Reject Bit Array (RBA) in hardware	No	No	IODA2+ changes the interrupt structure for the source controller. The RBA does not exist in the PHB4.
RID Translation CAM (RCAM), number of entries	64	128	
Migration Register Table (MRT), number of entries	8	16	The real number is actually one less than this number, but we set to an even number for simplicity.
TCE Validation Table (TVT), number of entries	512	1024	
TCE Cache structure	4-way set associative	4-way set associative	The PHB4 has an N-way set associative structure for timing efficiency and also allows for a larger number of TCEs cached.
TCE Cache, number of TCEs cached	512	1024	
TCE Page Sizes Supported (4 sizes)	4K, 64K, 2M, 1G	4K, 64K, 2M, 1G	TCE page sizes changed on POWER9 for the larger sizes.
MMIO BAR Table (MBT), number of entries	16	32	
MMIO Domain Table (MDT), number of entries	256	512	One entry per PE.
Peer-to-peer write capability	Yes	Yes	PHB to PHB communication through the Power bus interface.
CAPI Pseudo-VC support Non-Blocking Write (NBW), Credit Return Write (CRW)	No	Yes	No Pseudo-VC support in ETUX08.
Power Tunneled PCIe Atomics and ETU Decode Support	Yes	Yes	New Power Systems tunneled PCIe atomic transactions for POWER9/Nimbus.

2.1.2 Interfaces and Protocols

List of the interfaces:

- 1 x AIB (16-byte)
- 1 x BLIF (16-byte)
- 1 x SCOM

2.1.3 Functions

2.1.3.1 General functions

The general functions are as follows:

- 56-bit system addressing and 'no-translate' support.
  - PHB4 can support a full 56-bit system address for translate and no-translate modes.
  - PHB4 can also support 50-bit no-translate addressing as a special mode.
  - 32-bit no-translate support.
- TCE Cache
  - 4-way set associative.
  - ETUX08 will have 512 entries.
  - ETUX16 will have 1024 entries.
  - 'Match' logic must use full PCIe address and take the page size into account.
  - Binary-tree LRU to select between each way on replacement.
  - Firmware programmable set index/hash.
  - 'TCE Kill' function must handle conflicts with functional path and can be delayed.
  - 'Kill All' entries function can take several thousand cycles to complete because logic has to touch each SRAM entry one at a time and may have to wait on functional accesses in between updates.
  - 'TCE Kill' register will have a status indicator that indicates last cache kill/purge had completed.
  - TCE cache entry valid only when passes SEID checks for new address protection scheme when SEID checking is enabled.
- TCE and AIB Internal tags
  - ETUX08 will have 32 internal TCE tags. Maximum of 32 TCE fetches outstanding.
  - ETUX16 will have 64 internal TCE tags. Maximum of 64 TCE fetches outstanding.
  - Separate pool of AIB tags are removed for PHB4.
    - For PHB4, the TCE tag is used as the AIB tag and is only used for TCE reads/fetches.

2.1.3.2 Specific IBM methodology support functions

Specific IBM methodology support functions are as follows:

- Instantiated latches with SCAN chain IOs and LCB controls.
- Clock gating of instantiated latches. Clock gate unused logic during runtime.
  - ETU: clock gating will be designed into the HDL logic as it is written/changed.

2.2 PCIe Specific Requirements

This section covers the requirements specifically for the PCIe LBS macros (`pe_ph4_pciex08_mac` and `pe_ph4_pciex16_mac`). For brevity, these macros will be referred to as **PCIEX08** and **PCIEX16**.

2.2.1 Features

Table 2-2 Summary of PCIe features.

Table 2-2. PCIe features summary (Page 1 of 3)

Feature	PHB4 PCIEX08	PHB4 PCIEX16	Comment
BLIF Bus, Datapath Width	16 bytes	16 bytes	
BLIF Bus, BW in each direction	32GB/s	32GB/s	Note: PHB4, PCIEX08 can only achieve 16GB/s even though the physical buses are capable of 32GB/s since it is attached to a X08 link.
Outbound Completion Forwarding (OCF) Interface, Datapath Width	16 bytes	16 bytes	This bus is for DMA read completion header and payload.
Outbound Completion Forwarding (OCF) Interface, BW outbound	32GB/s	32GB/s	Note: PHB4, PCIEX08 can only achieve 16GB/s even though the physical buses are capable of 32GB/s since it is attached to a X08 link.
PCIE/BCLK Base Frequency	2GHz	2GHz	The 'top' of the PBL will run at constant 2GHz.
PCIE/PCLK Base Frequency	250MHz, 500MHz, 1GHz, 2GHz	250MHz, 500MHz, 1GHz, 2GHz	Gen 4 uses a divided clock to adjust for different link speeds. It is a 4:1 clock select versus a 3:1 in Gen 3.
PCIE BCLK to PCLK is asynchronous	No	No	All frequencies are in phase, no asynchronous boundaries.
Technology	MPZG, 14nm	MPZG, 14nm	
PCI-E Configuration	RC	RC	RC=Root Complex, EP=Endpoint.
PCI-E Core Stack Generation	Gen 4	Gen 4	
PCI-E Link Width	X8	X16	
PCI-E Max. Payload Size (Posted Writes)	512 bytes	512 bytes	For PHB4, the maximum posted payload size is limited to 512 bytes.
PCI-E Max. Payload Size (Completions)	256 bytes	256 bytes	The PHB limits the DMA read completion payload size to a maximum of 256 bytes in all configurations for buffer efficiency. 256 bytes is the upper limit by design, however firmware can program the maximum payload size to the smaller 128 bytes size as well. Thus, only two completion payload sizes are possible, 128 bytes and 256 bytes.
PCI-E Max. Outbound Write Request Size	128 bytes	128 bytes	MMIO/CI Store request size.
PCI-E Max. Outbound Read Request Size	32 bytes	32 bytes	MMIO/CI Load request size.
PCI-E Max. Outbound Number of Nonposted Requests	8	8	Max. number of PCIe read requests outstanding (CI Loads).
Max. Theoretical Link BW in each direction	16GB/s	32GB/s	No link overhead. Note: This is billions of bytes per second, not 2^{30} bytes per second.
Target BW in each direction	14GB/s (87.5% eff)	28GB/s (87.5% eff)	N% of theoretical link BW.
CAPI Pseudo-VC, Non-Blocking Write (NBW) Decode and Buffering Support	No	Yes	PCIEX08 does not support the CAPI Pseudo-VC logic and does not including additional buffering for NBWs.

Table 2-2. PCIe features summary (Page 2 of 3)

Feature	PHB4 PCIEX08	PHB4 PCIEX16	Comment
PCI-E Replay Buffer Size	8KB	16KB	Buffer implemented with an SRAM.
Max number of outstanding replay packets	64	128	Number based on total number of packets outstanding tracked in the TLDLP core.
PCI-E Ack Latency Assumption	500ns	500ns	500ns Ack latency and packet turnaround time is a conservative estimate for sizing purposes.
PCI-E RX, Posted Header Credits (RXPH)	64	128	Each credit is 16 bytes.
PCI-E RX, Posted Header Buffer Size (RXPH)	1KB	2KB	Buffer implemented with an SRAM.
PCI-E RX, Posted Data Credits (RXPD)	512	1024	Each credit is 16 bytes.
PCI-E RX, Posted Data Buffer Size (RXPD)	8KB	16KB	Buffer implemented with an SRAM.
PCI-E RX, Posted Header Buffer Size Non-Blocking Writes (NBWs) (RXPH, NBW)	0	2KB	NBWs share header and data credits with 'normal' writes. PCIEX08 does not support the CAPI Pseudo-VC logic and does not including additional buffering for NBWs.
PCI-E RX, Posted Data Buffer Size Non-Blocking Writes (NBWs) (RXPD, NBW)	0	16KB	NBW data buffering optimized for 128-byte payloads only. Buffer sized is number of headers times 128-byte. PCIEX08 does not support the CAPI Pseudo-VC logic and does not including additional buffering for NBWs.
PCI-E RX, Nonposted Header Credits (RXNPH)	64	128	Each credit is 16-bytes. Note for PHB4 the number of expected received posted and nonposted headers were doubled to accommodate smaller read request sizes like 128-byte or less. Some adapters only issue 128-byte reads. These adapters need more reads outstanding to maintain full BW.
PCI-E RX, Nonposted Header Buffer Size (RXNPH)	1KB	2KB	Buffer implemented with an SRAM.
PCI-E RX, Completion Header Credits	Infinite	Infinite	
PCI-E RX, Completion Data Credits	Infinite	Infinite	
PCI-E RX, Completion Data Buffer Size (RXCPLD)	256 bytes	256 bytes	Buffer implemented with an SSA array. Optimized to support max 8x32 byte inbound completions (CI Load response data).
PCI-E TX, Posted Header Slots (TXPH)	8	8	Number of posted write packets that can be stored pending transmission to the link.
PCI-E TX, Posted Header Buffer Size (TXPH)	128 bytes	128 bytes	Buffer implemented with an SSA array.
PCI-E TX, Posted Data Buffer Size (TXPD)	1KB	1KB	Buffer implemented with an SRAM.
PCI-E TX, Nonposted Header Slots (TXNPH)	8	8	Number of Nonposted Read packets that can be stored pending transmission to the link.

Table 2-2. PCIe features summary (Page 3 of 3)

Feature	PHB4 PCIEX08	PHB4 PCIEX16	Comment
PCI-E TX, Nonposted Header Buffer Size (TXNPH)	128 byte	128 byte	Buffer implemented with an SSA array.
PCI-E TX, Completion Slots (TXCPLH)	8	8	Number of Completion packets that can be stored pending transmission to the link. Note the PHB4 has the Outbound Completion Forward (OCF) interface and these are buffer slots to store packets from that interface.
PCI-E TX, Completion Data Buffer Size (TXCPLD)	512 bytes	512 bytes	Buffer implemented with an SRAM. The PHB4 has the Outbound Completion Forward (OCF) interface and these are buffer slots to store packets from that interface. This buffer is not store/forward and returns a credit on the OCF interface as soon as data moves to the PTL Note: The large outbound completion data buffer is implemented in the PB logic, close to the Power bus. The PB logic sends completions to the PHB4 via the OCF interface.

2.2.2 Interfaces and Protocols

- 1 x BLIF (16-byte) (both macros)
- 1 x OCF (16-byte) (both macros)
- 1 x REGBUS (both macros)
- 1 x PIPE x8 interface (PCIEX08)
- 1 x PIPE x16 interface (PCIEX16)

2.2.3 Functions

2.2.3.1 General functions:

- PCI Express Gen 4 Root Complex.
- PCI Device ID is: 0x044F.
  - Device ID field of PCI Configuration Space Header: 0xYYXX.
  - Byte 2 - XX - 4F.
  - Byte 3 - YY – 04.
- PCIe Gen 3: Atomic Operations.
  - PCIe Atomic Operations/Commands are not supported (either generation or reception).
  - PCIE logic will be configured to reject all inbound PCIe Atomics as Unsupported Request Response/Completions.
- PCIe Gen 3: TLP Processing Hints.
  - The ETU behavior is not affected by the TLP hint bits. These are merely meta-data bits passed with the transactions.

  - The ETU will pass the two hint bits from the PCIe TLP header (PH/TH) upbound via the AIB bus attribute field.
  - No steering tag support. The value is not stored or used in the ETU.
  - TLP hint bits apply to both DMA read and writes only.
  - TLP hint bits do not apply to CI Load/Stores or any other outbound requests.
- PCIe Gen 3: TLP Processing Hints.
  - PCIEX16 only, PCIEX08 macro does not use.
  - No steering tag support.
  - TLP hint bits apply to both DMA read and writes only.
  - TLP hint bits do not apply to CI Load/Stores or any other outbound requests.
- PCI Hot-Plug Functions.
  - REGB supports presence status capture and driving slot PERST# signal.
  - REGB does not drive power controls, power LED, or reference clock enable.
  - REGB also captures/drives cable IOs.
  - New Presence A/B status signals to support cables, switches, and slots.

2.2.3.2 Power Management Features

- The PCIe architected “low power states” are not supported. These modes will be disabled by default. These include L0s, L1, and L2 states.
- Dynamic link width change feature is supported.
  - Hardware support required to enable link width change interface.
  - Link width change from a wider to narrower link width and back again, up and down.
  - Simulation required to change link width in the middle of a simulation and continue running.
- Dynamic link speed change feature is supported.
  - Existing hardware supports changing the link speed.
  - Link speed change from high to low and back again, up and down.
  - Simulation required to change link speed in the middle of a simulation and continue running.

2.2.3.3 Specific IBM methodology support

- Instantiated latches w/SCAN chain IOs and LCB controls.
- Clock gating of instantiated latches. Clock gate unused logic during runtime.
  - PCIE macros: clock gating will be added via a tool that converts Verilog to VHDL. The clock gating logic is not designed into the Verilog source.

2.2.3.4 Other PCIe Features

- Support only one PCIe virtual lane and traffic class, VC0/TC0.
- No PCIe multi-root IOV support.
- No PCIe ATS services support.

2.3 PHB4 Common Requirements

2.3.1 Features

- Compliant with PCIe Gen 4 Architecture.
- Compliant with Power Systems RAS and error recovery strategies in the context of the current architecture.
- Compliant with IODA2+ Architecture.
- Tracing of internal facilities.
- No asynchronous clock boundaries within the ETU or PCIe macros.

3. Functional Description

This section describes the lower-level functional logic such as commands and command flows, address translation, ordering rules, and so on.

3.1 PHB4 Command Details

This section details the AIB and PCIe commands for the PHB4. The PHB4 uses the base push protocol for the AIB bus.

Table 3-1. AIB Command Type Encoding

Encoding	Type Name	Notes
0x00	DMA Read Request	Size and attributes matches original PCIe TLP header. Data is not cached.
0x02	CI Load Request	For MMIO and to send interrupt related commands.
0x04	TCE Read Request	Always 128-byte and naturally aligned.
0x0E	CAPI Read	AIB command w/out data.
0x10	DMA Write Request	Write Done response is not required.
0x12	CI Store Request	For MMIO and to send interrupt related commands.
0x1C	Peer-to-Peer Write	
0x1E	CAPI Write	AIB command with data.
0x40	CI Load Response	CI load response upbound.
0x41	TCE Read Response	
0xBF	DMA Read Sync	Special command for DMA read synchronization.

1. All other valid AIB command types are ignored and silently dropped.
2. Invalid AIB command types will flag an error.

Table 3-2. PCIe TLP command summary

Class	Type Name	Notes
Completion	Completion without Data	For PCI CFG Writes (nonposted) or for error responses.
Completion	Completion with Data	CI load responses.
Nonposted	Configuration Read Request	Outbound only.
Nonposted	Configuration Write Request	Outbound only.
Posted	Message Request	Inbound only.
Nonposted	Memory Read Request	
Posted	Memory Write Request	

1. All other valid PCIe command types are ignored and dropped.
2. Invalid PCIe request command types will result in a completion response of Unsupported Request.

3.1.1 Other Command details

Table 3-3. AIB Command Field Definition

Byte	Bit	Field	Description
0	00:07	Type(0:7)	AIB Command Type, see *Table 3-1* on page 29.
1	08:15	Size(4:11)	Lower 8 bits of the size field. The upper 4 bits are kept in the attribute field when appropriate. See *Table 3-6* on page 32. Transaction size in bytes. A value of zero represents a size of 4096 bytes.
2	16:23	Tag(0:7)	A Requester drives a unique 8-bit value it wants returned with the response. A Responder must return it in any associated response. DMA reads and writes drive the original tag field from the PCIe TLP.
3	24	Spec_Req	Unused, PHB4 drives to zeros.
3	25:26	Channel(0:2)	1 of 4 virtual channel indication. Used by receiver to route command to proper physical resources.
3	27	Port_Val	Unused, PHB4 drives to zeros.
3	28:31	PE Number(0:3)	These are the most significant 4 bits of the PE number for the command. Refer to the table on page 32 for the least significant 8 bits. These bits are used for ordering and error handling purposes.
4	32	CA Bit	PHB4 asserts to '1' to command the PB logic to send a CA error response for a DMA read completion. This field is only valid for DMA read commands. The CA and UR bit should never be asserted to '1' at the same time. If so, the PB logic would flag an error.
4	33	UR Bit	PHB4 asserts to '1' to command the PB logic to send a UR error response for a DMA read completion. This field is only valid for DMA read commands. The CA and UR bit should never be asserted to '1' at the same time. If so, the PB logic would flag an error.
4	35:37	Traffic Class(2:0)	These are the original traffic class field bits from the PCIe TLP header. These bits must be returned in the completion header for the DMA read completion. This field is only valid for DMA read commands.
4	38:39	Address(06:07)	Reserved for address expansion, PHB4 drives to zeros.
5 to 11	40:95	Address(08:63)	Address field of the request. The current PHB4 design and Power bus only support 56 address bits.
12 to 15	96:127	Attribute(0:31)	AIB command attribute field. See *Table 3-6* on page 32.

Table 3-4. AIB RX (Downbound) Command Details

Type	Size	AIB TX Data	Notes
CI Load Request	1-8, 16, or 32 bytes	No	Address cannot cross an 8-byte boundary. For sizes above 8-byte, the address and length alignment are the same.
CI Store Request, Peer-to-Peer Write	1-8, 16, 32, 64, or 128 bytes	Yes	Address cannot cross an 8-byte boundary. For sizes above 8-byte, the address and length alignment are the same.
TCE Read Response	128 bytes	Yes	Always 128-byte, always aligned.

1. See *Table 3-6 AIB Attribute Field Summary* on page 32.
2. The DMA Read Responses are not sent over AIB. The PB logic will send DMA Read. Responses/Completions over the OCF interface directly to the PCIe macro block.

Table 3-5. AIB RX (Upbound) Command Details

Type	Size	AIB TX Data	Notes
DMA Read Request	1-4KB	No	AIB size and tag field matches original PCIe TLP header.
TCE Read Request	128 bytes	No	Always 128-byte, aligned.
DMA Write Request	1-128 bytes	Yes	ETU generated writes are PEST writes. Address must not cross a 128-byte boundary.
Interrupt Request	N/A	No	AIB Address bits 52:63 contain the interrupt source number, right-justified.
CI Load Response	1-8, 16, or 32 bytes	Yes	Response phase for a CI Load Request. PE Number/NodeID is always zeros for PHB4 register reads.

1. See the 'AIB Attribute Field Summary' *Table 3-6* on page 32.

Table 3-6. AIB Attribute Field Summary

Command	Attr(00:07) Byte 0	Attr(08:15) Byte 1	Attr(16:23) Byte 2	Attr(24:31) Byte 3
CI Load Request	00:03 – Reserved, all zeros 04:07 – PB BAR Decode Vector(0:3)	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros
CI Store Request, Peer-to-Peer Write	00:03 – Reserved, all zeros 04:07 – PB BAR Decode Vector(0:3)	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros
TCE Read Response	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros
DMA Read Request	00:03 – Most significant size field bits 04:05 – TLP processing hint bits 06:07 – TLP attribute bits	TLP RequesterID bits (00:07)	TLP RequesterID bits (08:15)	PE Number(04:11)
TCE Read Request	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros
DMA Write Request	00:03 – Reserved, all zeros 04:05 – TLP processing hint bits 06:07 – TLP attribute bits	Reserved, all zeros	16 – CAPI Bit 17 – Last DMA Write Fragment 18 – ETU sourced write 19 – ASN Bit 20:23 – Reserved, all zeros	PE Number(04:11)
DMA Read Request	00:03 – Most significant size field bits 04:05 – TLP processing hint bits 06:07 – TLP attribute bits	TLP RequesterID bits (00:07)	TLP RequesterID bits (08:15)	PE Number(04:11)
CI Load Response	Reserved, all zeros	Reserved, all zeros	Reserved, all zeros	PE Number(04:11)

1. The 'Most significant size field bits' for DMA reads are the upper 4 bits of the DMA request size in bytes. The least significant 8 bits of the size are contained in the AIB command bits(08:15).
2. TLP attribute bits(0:1) are: {relaxed-ordering, no-snoop}.
3. The 'ETU Sourced Write' attribute bit is set when the PHB4 issues a DMA write to update the PEST error log information in memory.
4. The 'ETU Sourced Write' attribute bit is also used in the PBCQ logic (above the PHB4) to help determine if the AIB command was sourced from a CAPI adapter or locally from the PHB/ETU.
5. TCE Read Request commands should not be ordered by the PBCQ logic for the PHB4 design.
6. The 'PE Number(4:11)' bits are the least significant 8 bits of the PE number for the command. The most significant bits of the PE number are contained in AIB command bits (28:31).
7. The 'PE Number' bits are also known as the 'NodeID' bits in past designs. They are typically used for ordering Power bus commands.
8. The 'PB BAR Decode Vector(0:3)' is a one-hot bit vector. It indicates which BAR match hit in the PB logic for the CI Load/Store operation. The PHB/ETU uses this information to route the CI Load/Store to the correct internal logic.

The PHB/ETU performs the followings special checks against the PB BAR Decode Vector(0:3), *Table 3-7*:

- Vector is checked for all CI Load/Store request commands and only for those commands. It is a don't care for other commands.

- Vector is one-hot, and always be non-zero and only have a single bit set.
- Interrupt **EOI** and **Trigger** commands can be any legal CI Load/Store length supported by the design.

  Note: Only a size of 8 bytes or less is valid for CI Loads. CI Stores ignore the size and the data is not used.

Table 3-7. PB Bar Decode Vector (0:3)

Bit	Name	Notes
0	PHB4 Register Space	Internal PHB4 register space.
1	MMIO Space	'Normal' MMIO sent to PCIe link (M32 or M64). Peer-to-Peer Write. 'CAPI' Writes from PB logic.
2	Interrupt Space (LSI/MSI)	**EOI** command (CI Load only). **Interrupt Trigger** command (CI Store only).
3	Reserved/unused	Reserved/unused.

Table 3-8. Inbound Mapping PCIe to AIB command mapping

PCIe Command	PCIe RequestSize	PCIe Payload Length	AIB Command	Notes
Nonposted Memory Read	1-4KB	N/A	DMA Read Request	PHB4 passes whole TLP header to AIB TX.
Posted Memory Write	1-4KB	16-4KB	DMA Write Request	PHB4 breaks up into multiple 128 bytes. AIB TX, DMA Write Requests.
Posted Memory Write	1-16 bytes	16 bytes	MSI Interrupt	PCIe address decodes to an MSI interrupt request.
Completion without Data	N/A	N/A	N/A	Response phase for PCI CFG Writes.

Table 3-9. Outbound Mapping AIB to PCIe command mapping

AIB Command	PCIe Command	PCIe RequestSize	PCIe Payload Length	Notes
CI Load Request	Nonposted Memory Read	1-128 bytes	N/A	BAR match: M32 or M64.
CI Load Request	Configuration Read Request	1-4 bytes	N/A	BAR match: Internal PHB4 Registers. Read of register CONFIG_DATA.
CI Store Request	Posted Memory Write	1-128 bytes	16-128 bytes	BAR match: M32 or M64.

Table 3-9. Outbound Mapping AIB to PCIe command mapping

AIB Command	PCIe Command	PCIe RequestSize	PCIe Payload Length	Notes
Peer-to-Peer Write Request	Posted Memory Write	1-128 bytes	16-128 bytes	BAR match: M32 or M64.
CI Store Request	Configuration Write Request	1-4 bytes	16 bytes	BAR match: Internal Registers. Write of register CONFIG_DATA.
DMA Read Response	Completion with Data	N/A	16-128 bytes	

3.2 Ordering Rules

This section details the PHB4 command ordering rules. The PHB4 must adhere to the PCIe ordering rules since all traffic flows to/from the PCIe link. These are the ordering rules with deadlock avoidance.

Table 3-10. PCIe Ordering Rules

Row Pass Column? (Y[1], N[2], Y/N[3])	Posted Memory Write Message Request	Nonposted Memory Read IO Read/Write Configuration Read/Write PCIe Atomic Request	Completion Completion without Data Completion with Data
Posted Memory Write Message Request	N	Y	Y
Nonposted Memory Read IO Read/Write Configuration Read/Write PCIe Atomic Request[4]	N	Y/N	Y/N
Completion Completion without Data Completion with Data	N	Y	Y/N

1. Y - The command must be allowed to pass to avoid deadlock. The command cannot be blocked indefinitely and cannot be blocked by the command in the specified column.
2. N - The command cannot pass the command in the specified column. Otherwise, data integrity might be compromised.
3. Y/N - The command is allowed to pass, but is not required to pass the command in the specified column. It has no strict ordering requirement.
4. The PCIe atomic requests are fetch and add, unconditional swap, and compare and swap.

Table 3-11. AIB RX Channel Mapping (downbound)

Channel 0 can pass ch 1,2,3	Channel 1 can pass ch 2,3	Channel 2 can pass ch 3	Channel 3 can pass 0,1,2
TCE Read Response		Peer-to-Peer Write[1]	CI Load Request
			CI Store Request

1. Peer-to-Peer Writes can pass all commands inside the ETU. However, they are ordered as posted TLPs once they are issued to the PCIe link and follow PCIe ordering rules for Posted Memory Write commands.

Table 3-12. AIB RX Channel Mapping (upbound)

Channel 0 can pass ch 1,2,3	Channel 1 can pass ch 2,3	Channel 2 can pass ch 3	Channel 3 can pass 0,1,2
DMA Write Request	CI Load Response	DMA Read Request	DMA Write Request (NBW)
CI Store (Interrupt Request)		TCE Read Request	
CAPI Write		CAPI Read	
		DMA Read Sync	

3.3 Interrupt Types

The PHB4 interrupt logic encompasses both functional MSI interrupts as well as PHB4 internally generated error interrupts. The PHB4 receives functional MSI interrupts from the PCIe link. Error interrupts are generated internally by the PHB4 as a result of an error being detected (INF and ER class).

A PCIe switch or bridge can generate an MSI interrupt with the intention of reporting an error. These are treated the same as functional MSI interrupts in the PHB4.

The PHB4 also supports legacy PCI, Level Sensitive (LSI) interrupts.

3.3.1 LSI Interrupts

LSI Interrupts are functional interrupts sourced from an endpoint device or a PCIe switch or bridge. All LSI interrupt from external device arrive as assert/deassert PCIe message TLPs. Internally, they are converted to 4 level state bits which represent the 'virtual wire' of the LSI interrupt.

LSI interrupts are 'Level Sensitive' interrupts that hold their level asserted until deasserted by a device, hence the name. The LSI interrupt pool is shared with PHB4 internally generated interrupts. The PHB4 will generate interrupts for INF and ER class errors (see *Error Interrupts* on page 36).

Table 3-13 on page 36 shows the interrupt level mapping.

Table 3-13. LSI Interrupt Level Mapping

Level	Type Name	Notes
0	INTA	LSI interrupt from PCI device.
1	INTB	LSI interrupt from PCI device.
2	INTC	LSI interrupt from PCI device.
3	INTD	LSI interrupt from PCI device.
4	Reserved	Not used or driven.
5	Reserved	Not used or driven.
6	PHB4 INF Error Interrupt	PHB4 generated interrupt for errors.
7	PHB4 ER Error Interrupt	PHB4 generated interrupt for errors.

3.3.2 MSI Interrupts

MSI interrupts are functional interrupts sourced from an endpoint device or a PCIe switch or bridge. All MSI interrupts are received as Posted Memory Write Requests on the PCIe link of size less than 16 byte. The MSI interrupt is decoded via special address bits in a PCIe Posted Memory Write Request, see *PHB4 Configuration Register* on page 73.

The PHB4 requires that an MSI interrupt request from any source be aligned on a 16 bytes address boundary. The PHB4 will always use "byte 0" of the MSI data payload as the interrupt number and therefore requires the alignment restriction. The restriction will be enforced by the software programming of the MSI, PCI CFG space registers in the endpoint, switch, or bridge.

An MSI interrupt must be ordered behind all prior DMA Write requests. The PHB4 and all logic above the PHB4 must guarantee that this order is maintained otherwise data integrity can be compromised.

3.3.3 Error Interrupts

The PHB4 sends an error interrupt requests over the AIB interface for INF and ER class errors only. Fatal error class errors do not trigger any interrupt, they set their relevant error bit in the LEM FIR register.

The PHB4 sends an interrupt request command for error interrupts sent over the AIB bus. The PHB4 remembers a second error interrupt similar to the required bit for MSI interrupts. LSI level 6/7 is used for error interrupts. This interrupt level retains the state as follows: the second PE has requested an interrupt to be sent when there has already been one interrupt outstanding. Refer to *Error Reporting* on page 48 for more information on error interrupts and error reporting.

3.4 Interrupt Logic

The PHB is considered an Interrupt Virtualization Source Engine (IVSE) for interrupts as defined by IBM architecture. It is the master source of all physical interrupts for its PCIe link to the system.

The PHB4 accepts CI Loads and Stores to process interrupts. These are special commands that are identified by the CI command type on the AIB interface in combination with a match to the interrupt 'BAR' space in the PB logic, see *Section 3.1.1 Other Command details* on page 30. This 'BAR' space is also referred to as the 'ESB Page' in the IBM interrupt architecture.

Note only the sizes of 8 bytes or less are valid for these CI Load commands. If the command size is greater than 8 bytes, the command will return an all ones response for CI Loads. For CI Stores the command size is ignored as long as it is a legal size. The CI Store data is simply dropped for these interrupt commands and is not used. The CI Store commands will be honored and will have the requested side effect based on the CI Store address.

3.5 Transaction Logic

3.5.1 MMIO Transactions

The PHB4 supports two unique MMIO address spaces:

- **M32** - MMIO commands that hit this space are sent as normal PCIe reads/writes.
  - The PCIe TLP header is 3DW, memory read or memory write command.
  - AIB address is truncated to a 32 bit PCI address.
- **M64** - MMIO commands that hit this space are sent as normal PCIe reads/writes.
  - The PCIe TLP header is 4DW, memory read or memory write command.
  - AIB address is not truncated, the address is sent as part of the 64 bit PCI address.

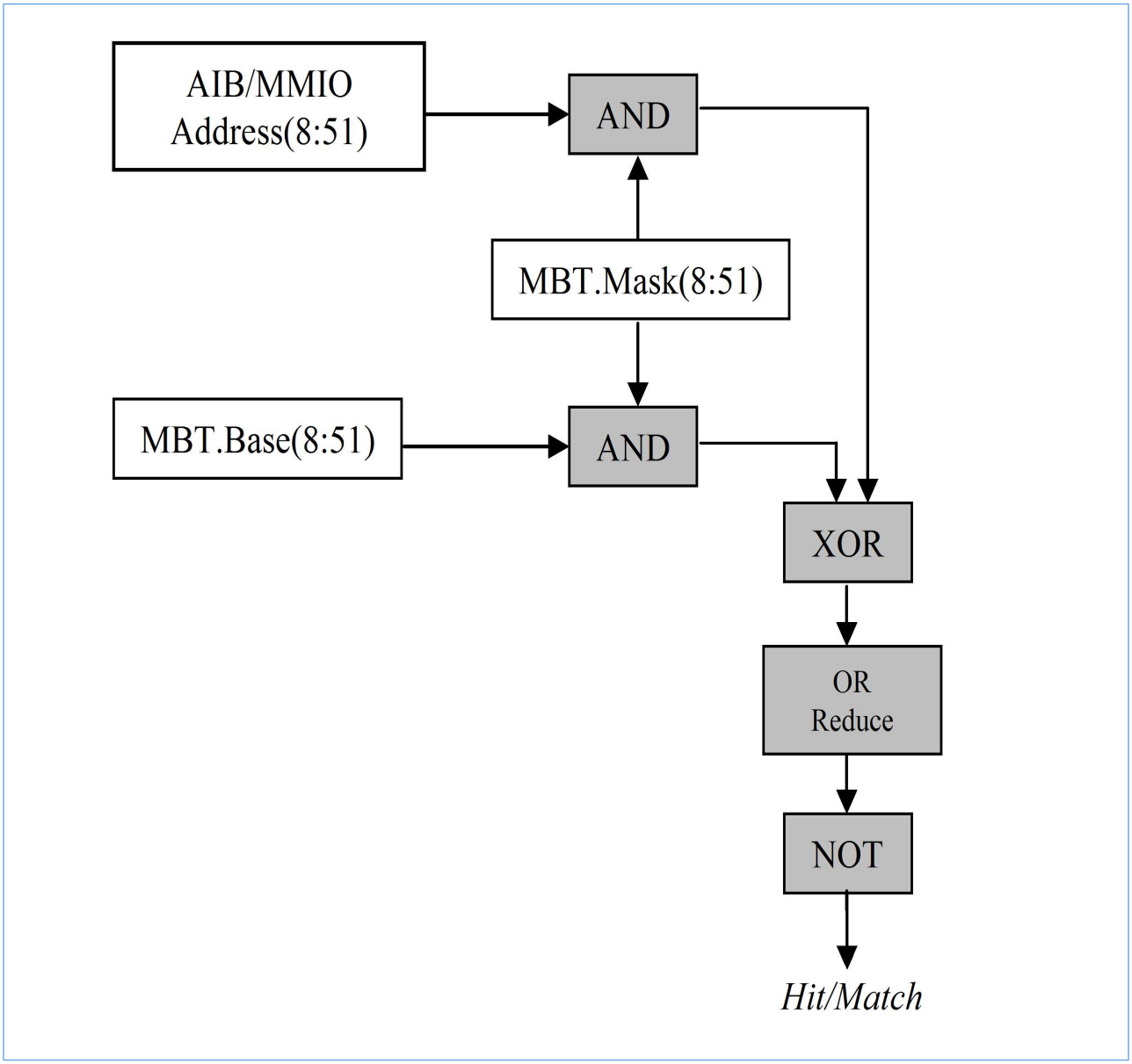
To process an MMIO transaction, the PHB4 first checks the AIB attributes for the command and the PB BAR Decode Vector(0:3). Refer to *Table 3-6 AIB Attribute Field Summary* on page 32. The decode vector must indicate the command is for the MMIO space to be processed as an MMIO.

The PHB4 then checks the CI Load/Store addresses for a match against its internal BARs to determine the space type (M32 or M64) and if the MMIO is allowed to target the PCIe link. The CI Load/Store transaction must match at least one BAR to be sent to the PCIe link. Note the 'CI Load/Store' and 'AIB/MMIO' addresses mean the same thing and are used interchangeably.

The PHB4 has a single pool of MMIO BARs that can be assigned independently to either the M32 or M64 space. The BARs are kept in the MMIO BAR Table (MBT) in the hardware. Refer to *Section 4-20 MBT Data (Part 0)* on page 101.

Figure 3-1 shows the logic flow of how the AIB address is compared against an individual BAR entry in the MBT. The `MBT.Mask` value is applied to the AIB address and `MBT.Base` value. The results are compared and if they match the transaction will have 'Hit' for that particular BAR entry. Note each BAR entry of the MBT is compared simultaneously and it is possible to hit multiple entries. The first entry that hits starting at entry 0 is the entry that will be used for the MMIO address translation and PE# decoding steps.

Figure 3-1. MMIO Hit/Match and Compare Logic Flow

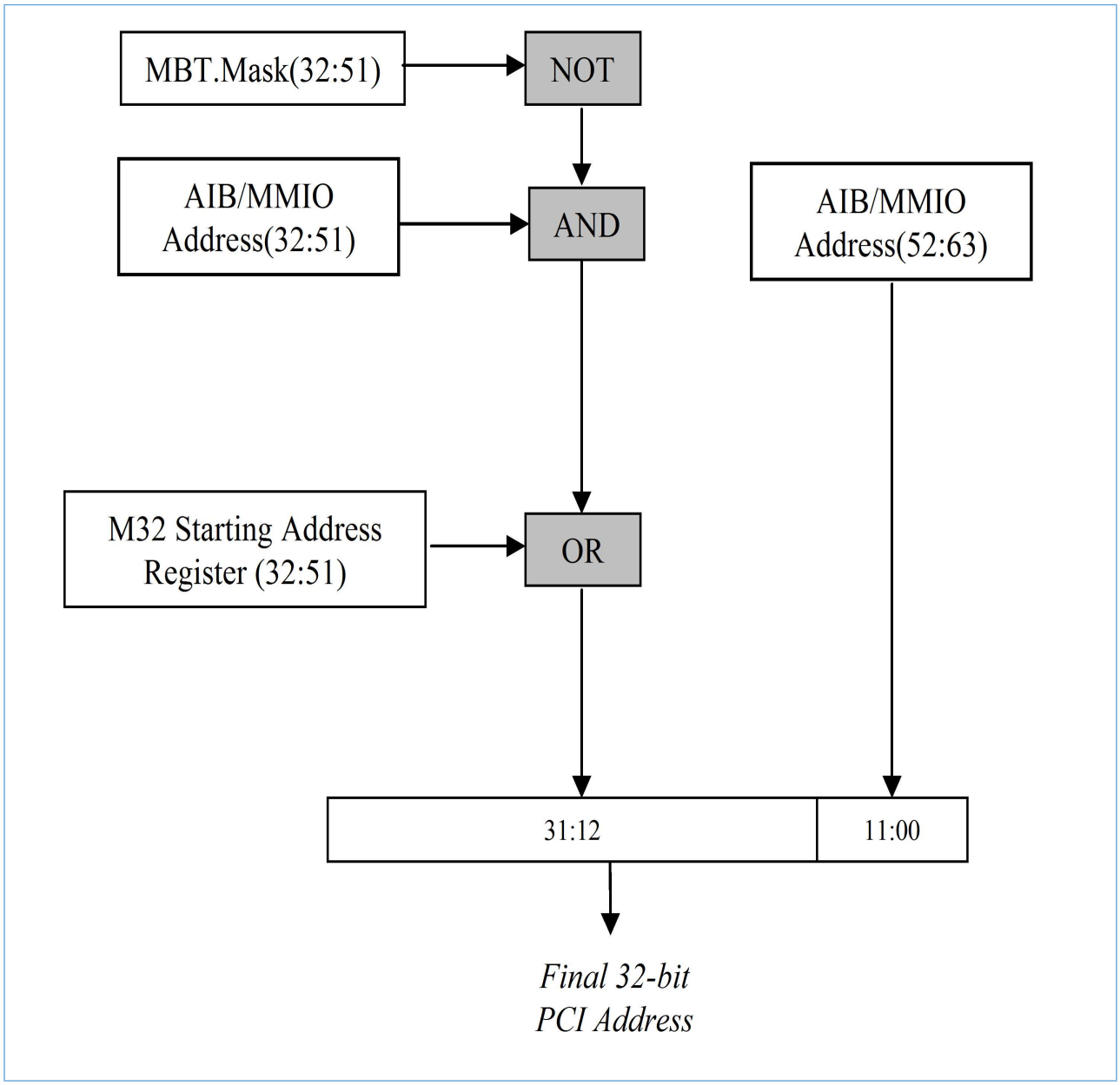


3.5.1.1 MMIO Address Translation

The AIB address is changed based on the MMIO type being M32 or M64. The changed address is the actual address sent on the PCIe link. The type is based on the MBT entry setting for the BAR that matched for the transaction. Refer to *Section 4-20 MBT Data (Part 0)* on page 101.

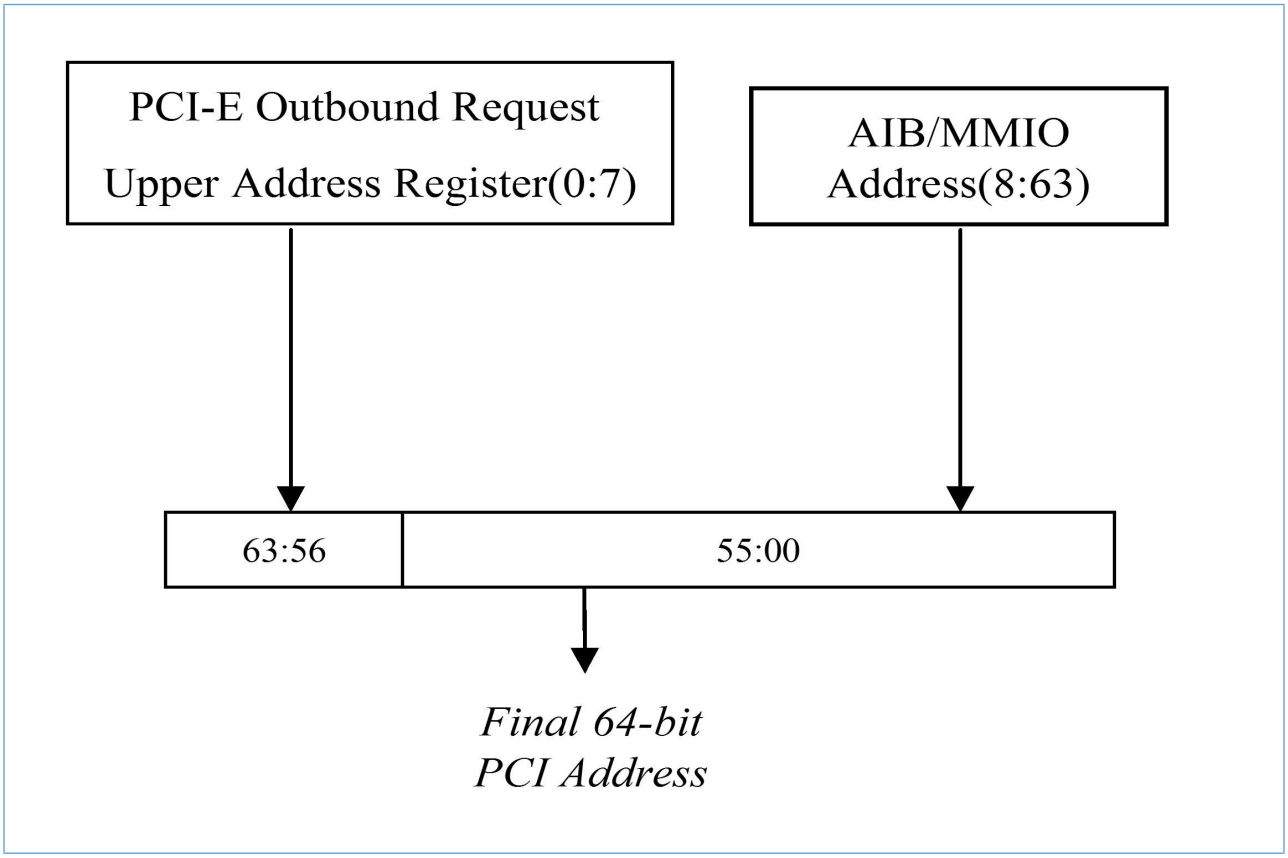
For the M32 case, the AIB address is truncated to a 32-bit address to send in the PCIe command TLP, refer to*Figure 3-2*. The least significant 32 bits of the AIB address is used. The upper bits are ignored. The inverted `MBT.Mask` (32:51) bits are applied to the AIB address bits. That result is logically ORed with the M32 Starting Address Register value (see *M32 Starting Address Register* on page 75). The least significant 12 bits are passed as-is.

Figure 3-2. M32 Address translation



For the M64 case, all of the AIB address is preserved. The final address sent in the PCIe TLP is a 64-bit address, refer to *Section 3-3* on page 40. The AIB address bits (8:63) are sent as-is in the least significant 56 bits. The most significant 8 bits are replaced with the value from the *PCIe Outbound Request Upper Address Register* on page 82.

Figure 3-3. M64 Address Translation



3.5.1.2 MMIO to PE Number Mapping

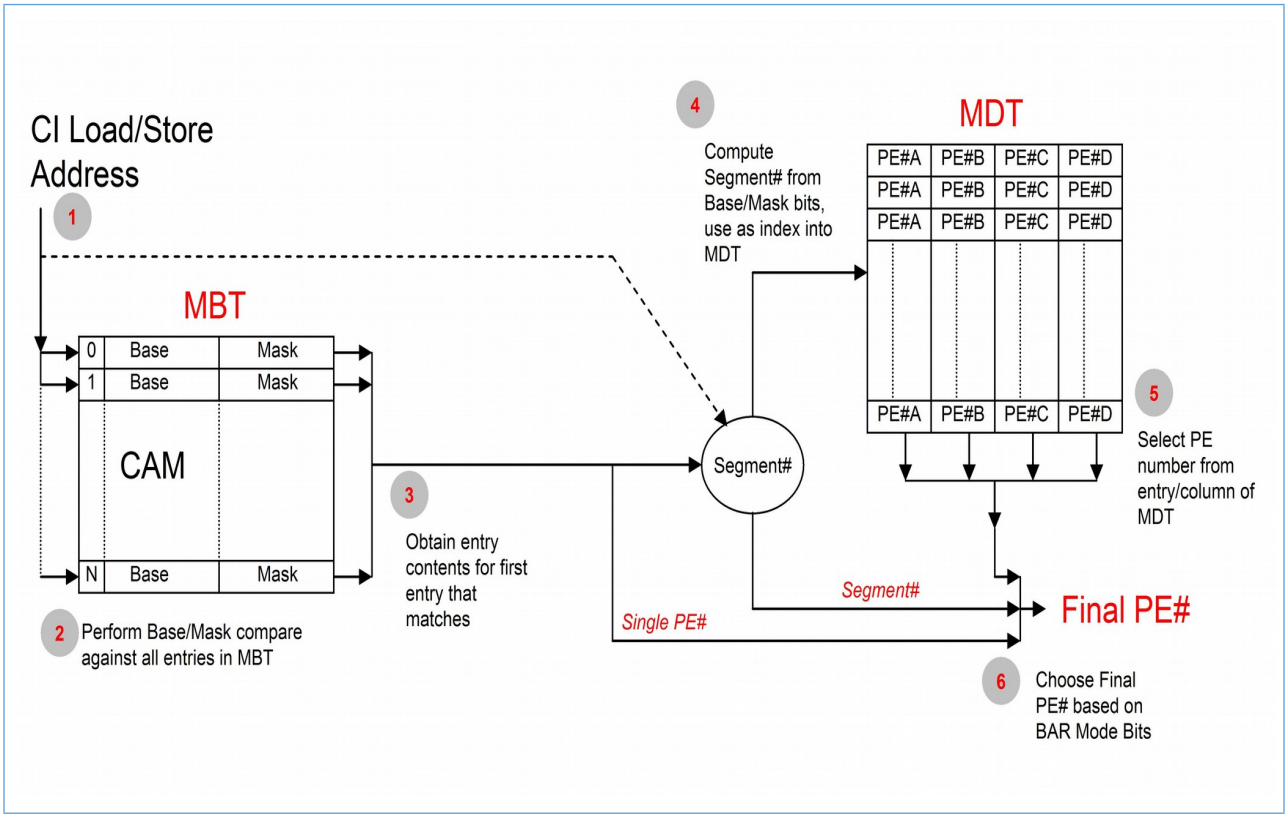
One of the primary outcomes of the MMIO processing flow is to determine the Partionable Endpoint (PE) number that is associated with the MMIO Address. Firmware assigns specific MMIO address regions to a particular PE number. This number is used by the EEH error checking logic. The logic checks if an MMIO transaction's PE number is currently in the 'stopped state' or to which PE number to assert into the stopped state on an error.

If an MMIO transaction's PE number is currently in the stopped state the logic will do the following with the transaction:

- MMIO Stores will be dropped and no transaction will be sent on the PCIe link.
- MMIO Loads will return an "all ones data" error response for the transaction, no transaction will be sent on the PCIe link.
- Refer to *Section 3.7 Error Logic* on page 45 for more details.

Figure 3-4 on page 41 shows the logic flow to determine the PE# for the MMIO address (CI Load/Store address).

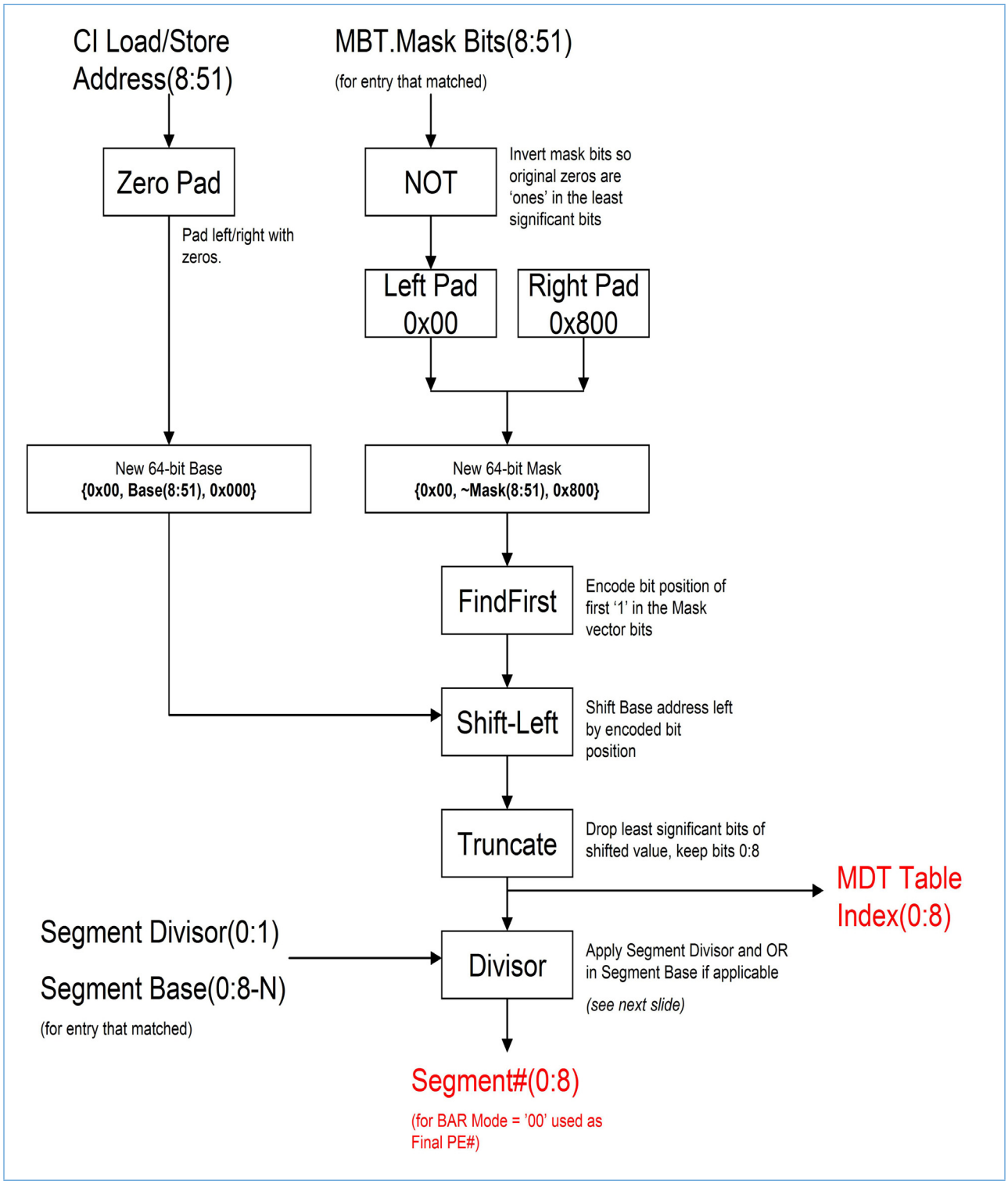
Figure 3-4. MMIO to PE decode and mapping flow



The steps of the MMIO to PE decode flow are described as follows:

1. Use the AIB/MMIO address bits 8:51 to compare against the MBT entries.
2. Compare against each MBT entry in parallel, see *Figure 3-1* on page 38.
3. Obtain MBT entry contents for first entry that 'hit'.
4. Compute the Segment# and MDT Table Index using MBT entry contents. The MDT Table Index number is used as the index into the MMIO Domain Table (MDT).
   - Refer to the Segment# computation flow in *Figure 3-5* on page 42.
   - Refer to the MDT content definition in *Table 4-20* on page 101.
5. Select the individual PE# from the MDT entry contents.
   - Refer to the MBT entry definition, Part 0: MDT Column Number(0:1), *Table 4-21* on page 101.
6. Choose PE# based on MBT 'BAR Mode' setting.
   - Refer to the MBT entry definition, Part 0: BAR Mode Bits(0:1), *Table 4-21* on page 101.
   - Firmware must choose this setting based on its requirements and its discretion.
   - The result will be the final PE# used for the MMIO Address.

Figure 3-5. MMIO Segment# Computation Flow



Note: For 'BAR Mode = 00', the Segment# is used as the PE# and the 'Segment Divisor(0:1)' field in the MBT entry selects how the Segment# is modified. Refer to *Table 3-14* and *Table 3-15*.

Table 3-14. Segment Divisor Table (vA4.1)

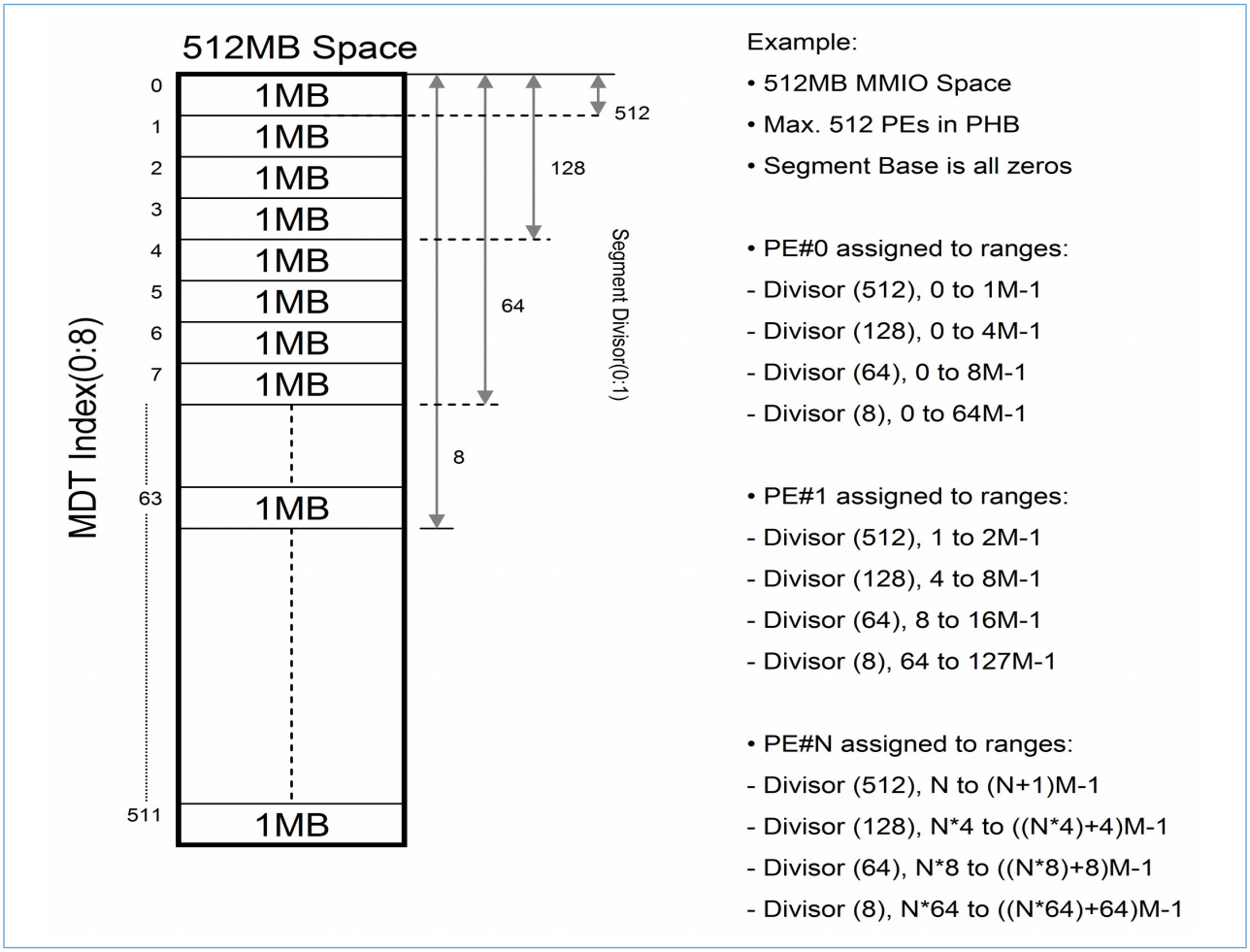
Segment Divisor(0:1) (from MBT entry)	Segment# (Final PE#) (0:8)	Comment
00 - (Max PEs)	MDT Table Index(0:8)	Same value as the MDT Table Index used to index the MDT. The number of segments is the maximum number of PEs supported. Segment base bits are not used.
01 - 128 equal segments	{Segment Base(0:1), MDT Table Index(2:8)}	N=7.
10 - 64 equal segments	{Segment Base(0:2), MDT Table Index(3:8)}	N=6.
11 - 8 equal segments	{Segment Base(0:5), MDT Table Index(6:8)}	N=3.

Table 3-15. Segment Divisor Table (vA4.2)

Segment Divisor(0:1) (from MBT entry)	Segment# (Final PE#) (0:8)	Comment
00 - (Max PEs)	MDT Table Index(0:8)	Same value as the MDT Table Index used to index the MDT. The number of segments is the maximum number of PEs supported. Segment Base bits are not used.
01 - 128 equal segments	{Segment Base(0:1), MDT Table Index(0:6)}	N=7.
10 - 64 equal segments	{Segment Base(0:2), MDT Table Index(0:5)}	N=6.
11 - 8 equal segments	{Segment Base(0:5), MDT Table Index(0:2)}	N=3.

Figure 3-6 on page 44 shows a MMIO segment example.

Figure 3-6. MMIO Segment Mode Example



3.6 Nontranslate/Untranslated Addresses

The TVE entry contents select between translate and nontranslate modes via the IO Page Size(0:4) field, refer to *Table 4-15* on page 95.

There are two sub-modes to the nontranslate mode:

- 50-bit (has range checking and only passes 50 bits of the PCI address(49:0) as the DMA address).
- 56-bit (no range checking and passes 56 bits of the PCI address(55:0) as the DMA address).

The sub-mode is selected via TVE entry bit 56, refer to *Table 4-15* on page 95. For 50-bit non-translate mode only, the Non-Translate Prefix Register is used, refer to *Section 4.4.2.19 Non-Translate Prefix Register* on page 82.

For 50-bit nontranslate mode only, in addition to the range checking in the TVE there is an additional check that the original PCI address bits (55:50) are all zeros. If the PCI address bits are non-zero it will be considered a range violation in this mode. Refer to the error bit in *Section 4.4.4.32 RXE_ARB Error Status Register* on page 145.

3.7 Error Logic

This section summarizes the major aspects of error logic in the PHB4. The following topics summarize the major aspects of the error logic in the PHB4 will be covered:

- Classification
- Detection
- Reporting
- Handling
- Recovery
- Injection

3.7.1 Error Classification

Errors in the PHB4 are categorized into three major classes:

- Informative (INF)
- Endpoint Recoverable (ER)
- Fatal/PHB4

Each class reports, handles, and recovers uniquely. The class is defined based on the settings in the error control registers, written in the initialization sequence.

3.7.1.1 INF Class

The INF error class indicates that an unexpected event occurred, but the hardware is fully operational. The event does not have any functional side effects. An example is a correctable ECC error. Errors in this class are low severity and do not put the PHB4 into an error state. These errors are masked and not enabled to cause actions like fencing, EEH freeze, and other actions. This error class is analogous to the PCIe architected error: ERR_COR (correctable error).

Further information about the INF class errors follows:

- Errors are for informational purposes only, they do not cause any side effects in the hardware.
- No fence should be raised after the error.
- No endpoints should be frozen after the error.
- These errors do not require a PHB4 reset or for the PCIe link to be dropped.
- The error bit will be logged in the error status register.
- These errors can or can not be masked for first error capture.
- LEM interrupts should be masked for these errors.
- These errors can or can not require an adapter reset, based on a predetermined threshold.
- An interrupt attention will be sent to the firmware when one of these errors occurs. The interrupt will be sent via the LSI mechanism; the interrupt is level 6.

3.7.1.2 ER Class

The ER error class includes errors that require the partitionable endpoint (PE) to be set to the frozen error state. All MMIO Loads/Stores to the frozen endpoint and DMA read/writes from the frozen endpoint are rejected. MMIO Loads return all ‘1’s for data and MMIO stores are discarded. DMA reads return either complete abort or unsupported request configuration status. DMA writes are discarded.

Further information about ER class errors follows:

- No fence should be raised after the error.
- One or more endpoints should be frozen after the error.
- These errors do not require a PHB4 reset or for the PCIe link to be dropped.
- The error bit will be logged in the error status register.
- These errors should be enabled for first error capture.
- These errors should be enabled to signal a PHB4 error interrupt. The LSI level 7 is used for ER.
- The PHB4 sets the stop state bits in the PHB4 internal PESTA and PESTB IODA tables.
- The PHB4 will update the appropriate entry in the PEST in system memory with error information.
- LEM interrupts should be masked for these errors.
- These errors can or can not require an adapter reset.
- The PHB4 logic is still fully functional, all transactions from non-frozen PEs are unaffected.

The ER class has three subclasses, as follows:

- ER(SINGLE) - Errors that affect one device or endpoint.
- ER(PELTV) - Errors that affect multiple endpoints. This class is only used for PCIe error messages received from the PCIe link.
- ER(ALL) - Errors that affect all devices or the device is not determinant.

In general, the ER(SINGLE) errors are the architected errors or errors that can be isolated to one PE. The ER(ALL) errors are errors that cause multiple devices to go into the error state or when the specific device cannot be determined. An example of an ER(SINGLE) error is a DMA request that points to an invalid TVT entry. An example of an ER(ALL) error is when the PCIe UTL core asserts its primary interrupt signal, AL_PINT. The reason and source of the interrupt cannot be immediately determined and might affect multiple devices. The AL_PINT signal is not a true interrupt. It is an error indicator.

This error class is analogous to the PCIe architected error ERR_NONFATAL. The PCIe architecture definition considers this error class an uncorrectable error.

3.7.1.3 Fatal/PHB4 Class

Fatal class errors indicate the PHB4 logic can be corrupt and cannot recover on its own or continue operation. These errors are rare and indicate a possible physical hardware problem. These error will raise a fence in the PHB4 that prevent new transactions from be sent or received on the AIB interface. The SCOM interface is expected to be functional to access internal registers, if possible. These errors require the PHB4 to be reset to be recovered.

Information about the Fatal/PHB4 class errors follows:

- A fence should be raised after the error. This will effectively disable the AIB bus to/from the PHB4.

- All register accesses must be performed via the SCOM bus after a fence is raised.
- One or more devices should be frozen after the error.
- These errors require a PHB4 reset after firmware gathers all error state and data.
- The PCIe link will go down.
- The error bit will be logged in the error status register.
- These errors should be enabled for first error capture.
- PHB4 error interrupts should be disabled for these errors since the AIB bus is fenced and the interrupt would never be presented.
- LEM interrupts should be enabled for these errors.
- These errors will require an adapter reset.

This error class is analogous to the PCI Express architected error: ERR_FATAL. PCI Express architecture definition considers this error class as an uncorrectable error.

3.7.2 Error Detection

3.7.2.1 LEM and Error Trap Registers

The following registers in the PHB4 register map contain the current error status information for all detected errors in the PHB4 ETU and PCIe blocks:

- *LEM FIR Accumulator Register* on page 120
- *PHB4 Error Status Register* on page 130
- *TXE Error Status Register* on page 136
- *RXE_ARB Error Status Register* on page 145
- *RXE_MRG Error Status Register* on page 153
- *RXE_TCE Error Status Register* on page 159

The LEM registers are spatially located together in the register address map starting at offset 0x0C00. The error trap registers are grouped similarly. The error trap register groups repeat for each unique error trap: PHB4, Outbound, InboundA, and InboundB. The error status bits in the error trap registers are fed from internal logic sources. These bits are ORed together to individual LEM FIR bits. The LEM FIR is a summary of all the errors in the PHB4, and it is the first register read to determine the error class to recover.

The bits in the LEM FIR and error status registers indicate their error class. The LEM FIR also indicates which error trap bits feed its individual bits or if a bit is sent directly to an LEM FIR bit.

3.7.2.2 PCIe Errors (detected by PCIe cores)

All of these signals are captured in the *PHB4 Error Status Register* on page 130.

3.7.2.3 Timeouts

The PHB4 contains several timers that time an array of different transactions and conditions. They typically fall into one of two types:

- Response transaction timers for DMA Reads and MMIO Loads.
- Forward progress timers for interfaces like the AIB and BLIF interfaces.

The Timeout Control Registers 1/2 are used to control timeouts in the PHB4. The values in these registers are specified in the initialization sequence. The values are always subject to change based on empirical lab experiments.

3.7.3 Error Reporting

When an error occurs it is captured in status registers and notifies firmware that the error has occurred. The only means to alert firmware that an error occurred in the PHB4 is by an interrupt attention command. All the error classes spawn an interrupt attention command, unless there is a special exception case where the error is configured to not generate an attention command.

3.7.3.1 Error Interrupts

INF class errors generate an interrupt attention command on the AIB bus of the PHB4. ER class errors also generate an interrupt attention command on the AIB bus of the PHB4. Fatal class errors use the chip-level interrupt generator to generate an interrupt attention command.

The chip-level interrupt is generated by controlling the LEM action register values. These registers are set up during the initialization sequence. LEM Action Registers 0 and 1 are used as a pair to specify a 2-bit value for each corresponding error in the FIR Accumulator as follows:

01: Fatal class interrupt

The PHB4 logic will generate an Interrupt command on the AIB bus as part of the error logging sequence for ER and INF class errors. The Interrupt level used is mapped as follows:

- LSI Interrupt with Interrupt level 6 for INF class errors.
- LSI Interrupt with Interrupt level 7 for ER class errors.

3.7.4 Error Handling

The PHB4 logic has implicit behavior when an error occurs that can be different depending on which error has occurred. There are specific actions that always take place based on the error's class. There are also other operations that can occur, such as dropping a command or returning all ‘1’s data with a bad response code.

3.7.4.1 INF Class Error Handling

No special handling is performed for the INF class errors. The logic behaves as if they never occurred at all.

3.7.4.2 ER Class Error Handling

The ER class errors set PE stopped state bits. The PE is considered frozen. For PEs that are in the stopped state:

- DMA Read/Writes, MSI interrupts, and MMIO Load Responses are rejected for the transaction that caused the ER error and for all transactions from the PE after the error.
- DMA Reads are silently dropped and not issued for the device in error state, a complete abort/unsupported response is returned for the transaction.
- DMA Writes are silently dropped and not issued for the PE in the error state.

- MMIO Load Responses return all ones data with a Command Reject response code for the PE in the stopped state.

3.7.4.3 Fatal/PHB4 Class Error Handling

Fatal errors act like INF errors in that the logic does not really do anything special except raise a fence on the AIB interface. The fence shuts off the flow of data in the event of a severe error. This helps prevent bad data from propagating, limits the number of additional errors that can occur, and prevents other parts of the chip from stalling or backing up.

The AIB fence is a 'solid' fence in that no packets of any kind get past it. The AIB fence affects traffic in both directions. Here are some additional notes about fencing:

- AIB TX Fence (Upward bound)
  - All packets queued to be sent are silently dropped and discarded if the fence is active.
  - If fence goes active in the middle of a transfer, the transfer will complete normally.
- AIB RX Fence (Downward bound)
  - If fence is active all commands are silently dropped and discarded.
  - If fence goes active during or before the command phase of a transaction the command will be silently dropped and discarded.
  - If fence goes active during the data phase of a transaction:
    - MMIO Writes: PHB4 will send data to PCIe link and mark the data as a poisoned TLP.
    - DMA Read Response: PHB4 will send a Completion with Completer Abort status to the PCIe link.
  - Active PCI CFG accesses are dropped and discarded when the fence goes active. This is to avoid deadlock issues that can lock out the SCOM bus. The fence condition must be cleared before any new CFG accesses are allowed.

3.7.5 Error Recovery

Error recovery consists of a sequence of operations that restores the PHB4 to the functional state that existed before the error occurred. Each of the three error classes described in *Section 3.7.1* on page 45 has its own sequence of operations for recovery. The classes are:

- Informative (INF)
- Programmable Endpoint Specific (ER)
- Fatal/PHB4

3.7.5.1 Recovery Sequences

INF Class

An INF class error is configured to not put the PHB4 into a device error/freeze state or to cause a fence. They are for informational purposes only and have no mainline logic side effects.

- Use the AIB interface for all steps unless stated otherwise.
- The notation value read indicates use the same register in a prior step.

Table 3-16. INF Recovery sequence (Page 1 of 3)

Steps	Access	Address	Write data	Comments
Recov_1	Read	x0138	Not applicable	Lock0 Register.
• This register will be read until the value returned is 0 (poll). • The zero value will notify the reader that the lock has been granted. **Note:** Any thread wishing to access the PHB4 registers must first obtain the lock to enforce mutual exclusion with other threads. This rule must be enforced by firmware. The PHB4 hardware does not enforce this rule.				
Recov_2	Read	x101C	Not applicable	PCI - IO Base/Limit and Secondary Status Register (size= 4-byte).
Recov_3	Write	x101C	0x000000FF	PCI - IO Base/Limit and Secondary Status Register (size=4-byte).
• Note firmware can read this register to determine if a completion received either a Completer Abort (CA) or Unsupported Request (UR) return code. • Bit 28: Received CA. • Bit 29: Received UR. • Capture status, clear all error bits.				
Recov_4	Read	x1050	Not applicable	EC - Device Control / Status Register (size = 4-byte).
Recov_5	Write	x1050	0x40000F00	EC - Device Control / Status Register (size = 4-byte).
• Enable correctable, non-fatal, fatal, and unsupported request error reporting. • Disable relaxed-ordering. • Enable a max payload size of 4KB. • Capture status, clear all error bits.				
Recov_6	Read	x111c	Not applicable	AER - Header Log Register #1 (size = 4-byte).
Recov_7	Read	x1120	Not applicable	AER - Header Log Register #2 (size = 4-byte).
Recov_8	Read	x1124	Not applicable	AER - Header Log Register #3 (size = 4-byte).
Recov_9	Read	x112c	Not applicable	AER - Header Log Register #4 (size = 4-byte).
• Capture AER header log, cannot clear since registers are read-only.				
Recov_10	Read	x1104	Not applicable	AER - Uncorrectable Error Status Register (size = 4-byte).
Recov_11	Write	x1104	0xFFFFFFFF	AER - Uncorrectable Error Status Register (size = 4-byte).
• Capture status, clear all error bits.				
Recov_12	Read	x1110	Not applicable	AER - Correctable Error Status Register (size = 4-byte).
Recov_13	Write	x1110	0xFFFFFFFF	AER - Correctable Error Status Register (size = 4-byte).
• Capture status, clear all error bits.				
Recov_14	Read	x1130	Not applicable	AER - Root Error Status Register (size= 4-byte).
Recov_15	Write	x1130	0xFFFFFFFF	AER - Root Error Status Register (size = 4-byte).
• Capture status, clear all error bits.				
Recov_16	Read	x1900	Not applicable	PBL - Error Status Register.
Recov_17	Write	x1900	<value read>	PBL - Error Status Register.
Recov_18	Read	x1908	Not applicable	PBL - First Error Status Register.
Recov_19	Write	x1908	0x00000000_00000000	PBL - First Error Status Register.

Table 3-16. INF Recovery sequence (Page 2 of 3)

Steps	Access	Address	Write data	Comments
Recov_20	Read	x1940	Not applicable	PBL - Error Log Register 0.
Recov_21	Write	x1940	0x00000000_00000000	PBL - Error Log Register 0.
Recov_22	Read	x1948	Not applicable	PBL - Error Log Register 1.
Recov_23	Write	x1948	0x00000000_00000000	PBL - Error Log Register 1.
Recov_24	Read	x1C00	Not applicable	REGB Error Status Register.
Recov_25	Write	x1C00	<value read>	REGB Error Status Register.
Recov_26	Read	x1C08	Not applicable	REGB First Error Status Register.
Recov_27	Write	x1C08	0x00000000_00000000	REGB First Error Status Register.
Recov_28	Read	x1C40	Not applicable	REGB Error Log Register 0.
Recov_29	Write	x1C40	0x00000000_00000000	REGB Error Log Register 0.
Recov_30	Read	x1C48	Not applicable	REGB Error Log Register 1.
Recov_31	Write	x1C48	0x00000000_00000000	REGB Error Log Register 1.
• Capture PCIe macro specific error information in each register. • Clear set bits in each register with the value read from the register. These registers a write '1' to clear. • Capture all error trap register contents and logging values. • Clear error status bits read previously and clear all error log register contents.				
Recov_32	Read	x0D00	Not applicable	TXE Error Status Register.
Recov_33	Write	x0D00	<value read>	TXE Error Status Register.
Recov_34	Read	x0D08	Not applicable	TXE First Error Status Register.
Recov_35	Write	x0D08	0x00000000_00000000	TXE First Error Status Register.
Recov_36	Read	x0D40	Not applicable	TXE Error Log Register 0.
Recov_37	Write	x0D40	0x00000000_00000000	TXE Error Log Register 0.
Recov_38	Read	x0D48	Not applicable	TXE Error Log Register 1.
Recov_39	Write	x0D48	0x00000000_00000000	TXE Error Log Register 1.
Recov_40	Read	x0D80	Not applicable	RXE_ARB Error Status Register.
Recov_41	Write	x0D80	<value read>	RXE_ARB Error Status Register.
Recov_42	Read	x0D88	Not applicable	RXE_ARB First Error Status Register.
Recov_43	Write	x0D88	0x00000000_00000000	RXE_ARB First Error Status Register.
Recov_4	Read	x0DC0	Not applicable	RXE_ARB Error Log Register 0.
Recov_45	Write	x0DC0	0x00000000_00000000	RXE_ARB Error Log Register 0.
Recov_46	Read	x0DC8	Not applicable	RXE_ARB Error Log Register 1.
Recov_47	Write	x0DC8	0x00000000_00000000	RXE_ARB Error Log Register 1.
Recov_48	Read	x0E00	Not applicable	RXE_MRG Error Status Register.
Recov_49	Write	x0E00	<value read>	RXE_MRG Error Status Register.
Recov_50	Read	x0E08	Not applicable	RXE_MRG First Error Status Register.
Recov_51	Write	x0E08	0x00000000_00000000	RXE_MRG First Error Status Register.
Recov_52	Read	x0E40	Not applicable	RXE_MRG Error Log Register 0.

Table 3-16. INF Recovery sequence (Page 3 of 3)

Steps	Access	Address	Write data	Comments
Recov_53	Write	x0E40	0x00000000_00000000	RXE_MRG Error Log Register 0.
Recov_54	Read	x0E48	Not applicable	RXE_MRG Error Log Register 1.
Recov_55	Write	x0E48	0x00000000_00000000	RXE_MRG Error Log Register 1.
Recov_56	Read	x0E80	Not applicable	RXE_TCE Error Status Register.
Recov_57	Write	x0E80	<value read>	RXE_TCE Error Status Register.
Recov_58	Read	x0E88	Not applicable	RXE_TCE First Error Status Register.
Recov_59	Write	x0E88	0x00000000_00000000	RXE_TCE First Error Status Register.
Recov_60	Read	x0EC0	Not applicable	RXE_TCE Error Log Register 0.
Recov_61	Write	x0EC0	0x00000000_00000000	RXE_TCE Error Log Register 0.
Recov_62	Read	x0EC8	Not applicable	RXE_TCE Error Log Register 1.
Recov_63	Write	x0EC8	0x00000000_00000000	RXE_TCE Error Log Register 1.
Recov_64	Read	x0C80	Not applicable	PHB4 Error Status Register.
Recov_65	Write	x0C80	<value read>	PHB4 Error Status Register.
Recov_66	Read	x0C88	Not applicable	PHB4 First Error Status Register.
Recov_67	Write	x0C88	0x00000000_00000000	PHB4 First Error Status Register.
Recov_68	Read	x0CC0	Not applicable	PHB4 Error Log Register 0.
Recov_69	Write	x0CC0	0x00000000_00000000	PHB4 Error Log Register 0.
Recov_70	Read	x0CC8	Not applicable	PHB4 Error Log Register 1.
Recov_71	Write	x0CC8	0x00000000_00000000	PHB4 Error Log Register 1.
Recov_72	Read	x0C00	Not applicable	LEM FIR Accumulator Register.
Recov_73	Write	x0C08	~<value read>	LEM FIR AND Mask Register.
Recov_74	Write	x0C40	0x00000000_00000000	LEM WOF Register.
• Capture all error trap register content and logging values. • Clear error status bits read previously and clear all error log register contents. • Write bitwise inverse of FIR bits to LEM FIR AND Mask Register to clear set bits in the FIR.				
Recov_75	Write	x0138	<init sequence value >	LEM Error Mask Register.
• Update mask value to value from init sequence.				
Recov_76	Write	x0138	0x00000000_00000000	Lock0 Register.
• Firmware clears its lock once it is finished with its operations.				
Recov_77	nop	Not applicable	Not applicable	INF Recovery Sequence Complete.

ER Class

ER class errors set the MMIO and DMA error state bits in the local IODA State Tables PESTA and PESTB. If the error bit is classified as ER (SINGLE), it affects only one device. The PHB4 sends a write transaction to update the entry matching the PE number that reported the error in the PE State Table in system memory with error logging information. This is followed by an interrupt attention. If the error bit is classified as ER (ALL) all devices are affected. The PHB4 sends a write transaction to update entry 0 of the PE State Table in system memory with error logging information. This is followed by an interrupt attention.

Note: Use the AIB interface for all steps unless stated otherwise.

Table 3-17. Recovery sequence

Step	Access	Address	Write data	Comment
Recov_1	nop	Not applicable	Not applicable	Follow all INF recovery steps.
• Note firmware should not release the Lock0 Register in the last step of the INF recovery sequence. • The Lock0 will be held until the end of the sequence.				
Recov_2	Write	x0220	x8014_0000_0000_0000	IODA Table Select Register.
Recov_3	Read	X0228 (4/8, x0)	Not applicable	IODA Table Data Register.
• Set up Table Select Register to Read/Write the PE Error Vector Table (PEEV) in Auto-Increment mode. • Read Data 0 Register N times to fetch all bits of the vector, 64 bits per read. • Each bit represents its corresponding PE number; bit[0] = PE0, bit[1] = PE1, and so on. • For each bit set in the vector, that PE is in the stopped state. The PE State Table in system memory can now be read for error details.				
Recov_4	Write	x0138	0x00000000_00000000	Lock0 Register.
• Firmware will clear its lock once it is finished with its operations.				
Recov_5	nop	Not applicable	Not applicable	ER Recovery Sequence Complete.

Fatal/PHB4 Class

A fatal error has been identified. These errors put the PHB4 into an error state and also raise the PHB4 fence so that no new DMA reads/writes or MMIO load responses leaves the PHB4. These errors are rare.

Because the fence has been raised, the DMA reads and writes are blocked. This eventually blocks PCIe completions. Therefore, PCI CFG space reads or any reads to an adapter do not complete including PCI CFG space reads of the local CFG space in the PHB4. No CFG accesses will be issued until the fence condition has been cleared; that is, after the PHB4 has been reset and initialized.

Note: For the Fatal class only because the AIB bus is expected to be fenced, use the ASB interface for all steps unless stated otherwise.

Table 3-18. Fatal Recovery sequence

Step	Access	Address	Write data	Comment
Recov_1	nop	Not applicable	Not applicable	Follow all INF recovery steps.
• No change from INF recovery sequence.				
Recov_2	nop	Not applicable	Not applicable	Firmware waits 1 second before proceeding.
• This will allow potential DMA read responses in flight to be flushed out.				
Recov_3	nop	Not applicable	Not applicable	Fatal Recovery Sequence Complete.
• Proceed to reset the PHB4 and start the PHB4 initialization sequence.				

3.7.6 Error Injection

The PHB4 hardware supports error injection for specific error bits via its internal registers. Refer to the register sections for further details.

It is also possible to inject errors by modifying the RTC, TVT, and TCE entries for a specific PE, or by sending a transaction that causes a conflict with the current RTC, TVT, and TCE entry settings that cause a specific error to occur.

4. Software Interfaces

4.1 Register Definitions

Table 4-1 lists the acronyms used to indicate the type of access available for a specific register.

Table 4-1. Register Access Type Legend

Type	Description
RO	Read-Only. Reads to the register are allowed and returns the current value in the register. A write operation is not allowed and does not affect the register value. If the write operation requires a status, the status always indicates a successful write.
RW	Read-Write. The value in the register can be either read or written. There are some exceptions, but generally the register holds the last value written.
RW1C	Read-Write 1 to Clear. Reads to the register are allowed and returns the current value in the register. A write operation with a value of ‘1’ clears the register bit to ‘0’'.
RWOR	Read-Only, Write 1 to Set. Reads to the register are allowed and returns the current value in the register. A write operation with a value of ‘1’ sets the register bit to ‘1’.
WO	Write-Only. The value in the register can only be written, never read. A read of the register always returns all zeros.

Table 4-2 lists suffixes used with the acronyms described in *Table 4-1*.

Table 4-2. Register Access Type Suffixes

Suffix	Description
H	Hardware update. If appended to a Register Access Type name, the value in the register can be updated by the hardware. Examples: RW1CH, RW1CHS.
S	Sticky. If appended to a Register Access Type name, the value in the register is retained after reset (any reset). Examples: RW1CH, RW1CHS.

Table 4-3 shows the RST register value legend.

Table 4-3. RST Register Value Legend

Term	Description
0	All bits are set to ‘0’.
0xN	4-bits of hex value ‘N’ are applied to all existing bits in a ‘right-justified’ manner. Can be multiple digits for ‘N’.

Section 4-4 on page 56 describes what the term “Reserved” means when used with certain access types.

Table 4-4. Register "Reserved" Name Legend

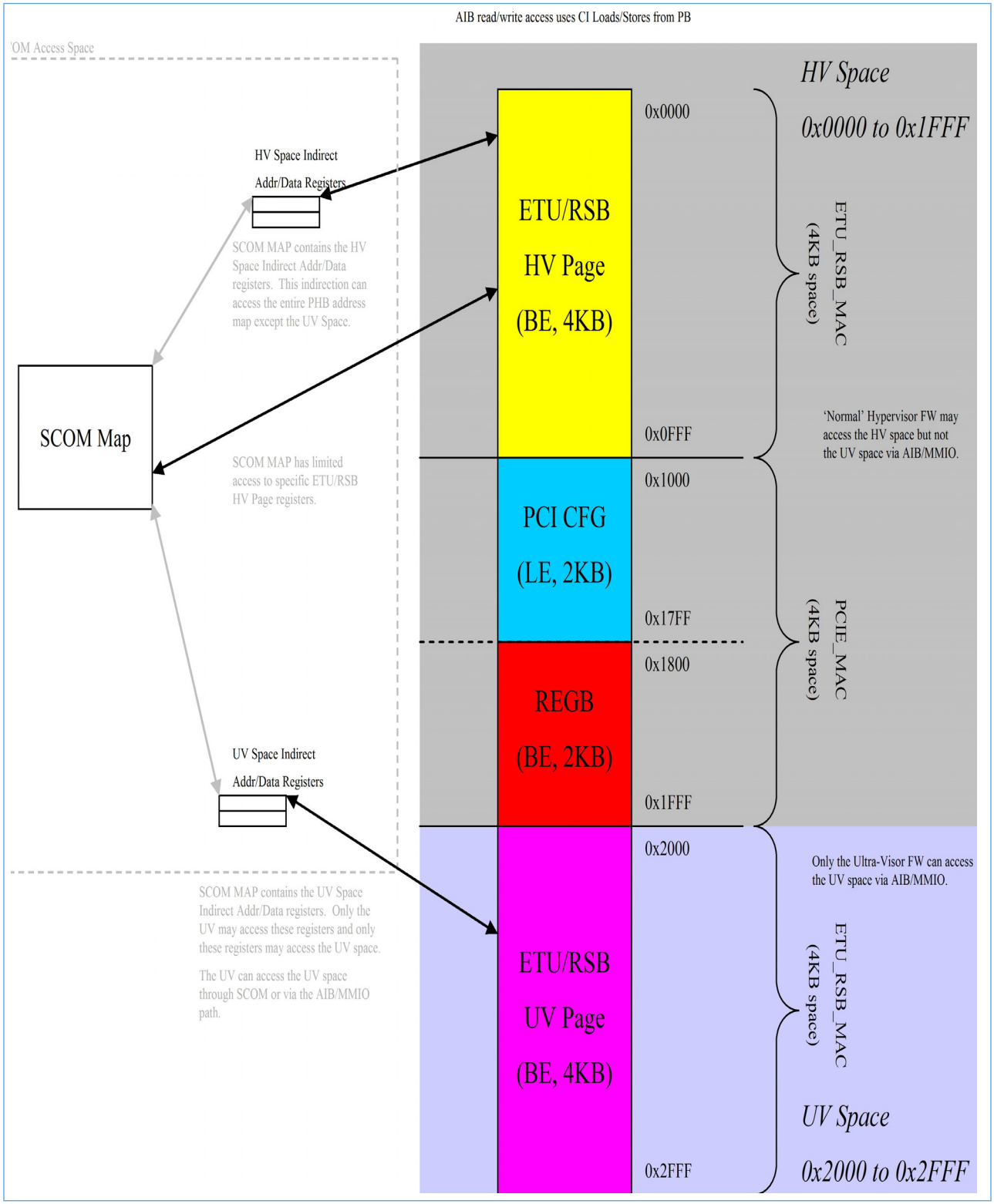
Name	Type	Description
Reserved	RO	Not physically built.
Reserved	RW	Register physically built but not assigned to a function.

4.2 PHB4 Unified Address Space

The PHB4 has a unified address map that can be accessed via SCOM or via MMIO operations on the AIB bus. The first two 4KB pages (0x0000 to 0x1FFF) are named the Hypervisor Space (HV space). The third 4K page (0x2000 to 0x2FFF) is named the Ultra-Visor Space (UV space). The UV space is a protected space only accessible by the Ultra-Visor. The protection is enforced outside of the PHB4. There is no hardware protection in the PHB4, refer to *Figure 4-1* on page 57.

The PHB4 configuration registers are accessed via MMIO Loads/Stores from either the AIB or via read and write operation from the SCOM bus. These registers cannot be accessed via the PCIe link. The AIB and SCOM bus can access the same registers within the PHB4 register space. The AIB bus is the primary means of accessing the PHB4 registers. The SCOM bus is an alternate means of accessing the registers used by the system diagnostic logic and is guaranteed access to the register space even if the AIB bus is blocked or has an unrecoverable error. MMIO register accesses via the AIB bus are guaranteed to be ordered with other MMIOs. SCOM accesses have no ordering requirements or restrictions; they are completely unordered in relation to AIB bus operations. The AIB bus MMIO Load/Store operations use a system address to access the PHB4 internal registers. The SCOM bus uses a different address map, thus a different address for its operations.

Figure 4-1. PHB4 Register map diagram

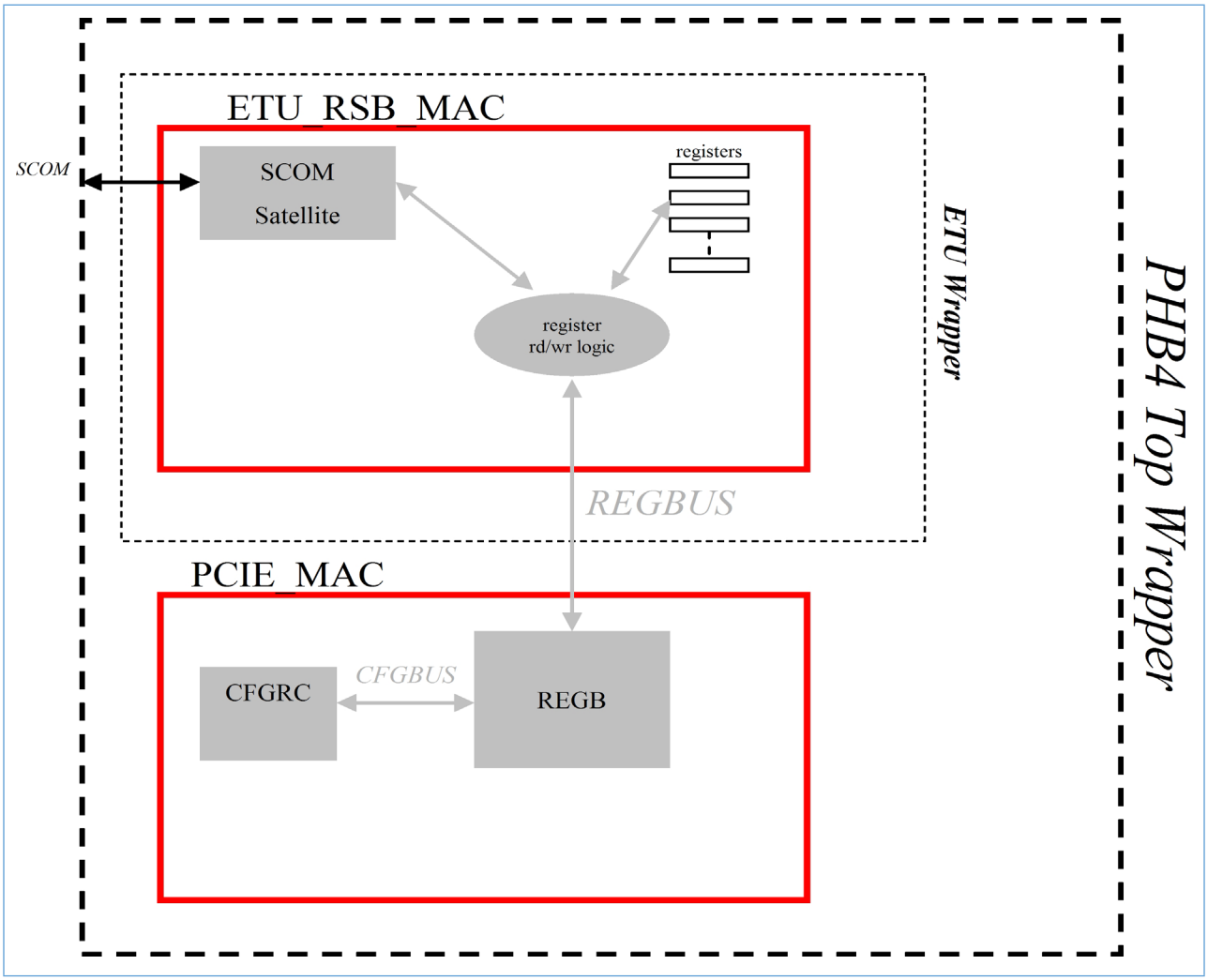


4.3 SCOM Registers (ETU)

The ETU macro has a single SCOM access port and single SCOM satellite. This satellite is capable of accessing the entire PHB4 register address map using the indirect access registers described in this section. The actual SCOM satellite physically resides in the ETU/RSB macro, but it has internal ports to access the registers of all the other macros in the PHB4 including the PCIe/REGB and PCIe/CFGRC space. Refer to *Figure 4-2*.

The SCOM registers in the SCOM address map themselves are only accessible via the SCOM port, they are not accessible via MMIO. MMIO operations can only access the PHB4 register address map through the AIB interface.

Figure 4-2. SCOM Port Register access diagram



4.3.1 SCOM Register Address Map

Table 4-5 details the SCOM register address map.

Table 4-5. SCOM Register address map

SCOM Offset	Description	Page
0x00 (00)	SCOM - HV Indirect Address Register	60
0x01 (01)	SCOM - HV Indirect Data Register	61
0x02 (02)	SCOM - UV Indirect Address Register (vA4.1) Reserved, invalid access (vA4.2)	
0x03 (03)	SCOM - UV Indirect Data Register (vA4.1) Reserved, invalid access (vA4.2)	
0x04 (04)	Reserved, invalid access (vA4.1) SCOM – UV - Secure Address Exclude CMP/MSK Register (vA4.2)	295
0x05 (05)	Reserved, invalid access	
0x06 (06)	Reserved, invalid access (vA4.1) SCOM – UV - Secure Address Include CMP Register (vA4.2)	297
0x07 (07)	Reserved, invalid access (vA4.1) SCOM – UV - Secure Address Include MSK Register (vA4.2)	297
0x08 (08)	PHB/ETU: LEM FIR Accumulator Register	120
0x09 (09)	PHB/ETU: LEM FIR AND Mask Register	125
0x0A (10)	PHB/ETU: LEM FIR OR Mask Register	128
0x0B (11)	PHB/ETU: LEM Error Mask Register	127
0x0C (12)	PHB/ETU: LEM Error AND Mask Register	127
0x0D (13)	PHB/ETU: LEM Error OR Mask Register	128
0x0E (14)	PHB/ETU: LEM Action 0 Register	128
0x0F (15)	PHB/ETU: LEM Action 1 Register	129
0x10 (16)	PHB/ETU: LEM WOF Register	129
0x11 (17)	Reserved, invalid access	
0x12 (18)	Reserved, invalid access	
0x13 (19)	Reserved, invalid access	
0x14 (20)	Reserved, invalid access	
0x15 (21)	Reserved, invalid access	
0x16 (22)	Reserved, invalid access	
0x17 (23)	PHB/ETU: Performance Monitor Configuration Register	169
0x18 (24)	PHB/ETU: Performance Monitor Counter Register 0	173
0x19 (25)	PHB/ETU: Performance Monitor Counter Register 1	173
0x1A (26)	PHB/ETU: Performance Monitor Counter Register 2	174
0x1B (27)	PHB/ETU: Performance Monitor Counter Register 3	174
0x1C (28) to0xFF	Reserved, invalid access	

4.3.2 SCOM Registers

4.3.2.1 SCOM - HV Indirect Address Register

SCOM space register. This register can be used to indirectly address the 'Hypervisor Accessible' register space in the PHB4. HV can access the PHB4 address range from 0x0000 to 0x1FFF.

Mnemonic SCOM_HVIAR

Address Offset 0x00

Bit	Field Mnemonic	Type	Reset Value	Description
0	Valid Bit	RW	0	Valid Bit must be set to '1', else read/write access will not occur when the indirect data register is read/written. When '0', any read/write access to the indirect data register will be rejected and will flag an error bit.
1	Size Bit	RW	0	0' = 8 byte register access '1' = 4 byte register access The 4 byte register access is required when accessing the PCI CFG space registers.
2	Auto-Increment Bit	RW	0	Set this bit to automatically increment the Indirect Address field in this register by 4 byte/8 byte after the Indirect Data Register is accessed. Default is to not automatically increment the address. The increment by 4 byte/8 byte amount is based in the Size Bit setting.
03:50	Reserved	RO	0	Reserved.
51:61	Indirect Address(00:10)	RW	0	Indirect Address Bits(00:10). This is the 4 byte aligned address for the register operation. If the access is for 8 byte then bit 61 should be set to '0'.
62:63	Indirect Address(11:12)	RO	0	Indirect Address Bits(11:12). 4 bytes aligned minimum offset bits are always zeros.

4.3.2.2 SCOM HV Indirect Data Register

SCOM space register. This register is the companion to the indirect address register. The indirect register access, read or write, is triggered by a read or write to this register respectively.

A read of this register will generate an internal read of the register at the address in the indirect address register. The data read from the internal register is returned with the read operation. Writes to this register will write to the register at the address of the indirect address register. Reads or writes to this register will increment the indirect address after each access to this register if the Auto Increment bit is set.

Mnemonic SCOM_HVIDR

Address Offset 0x01

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Indirect Data(00:63)	RW	0	Data to Write/Read Result Data.

4.3.2.3 SCOM Direct Mapped Registers

The following registers are directly mapped into the SCOM address space and do not require access via the indirect address/data registers. It is still possible to access these registers via the indirect access registers, it is just not required. These registers are given direct access for convenience and reduced access delay.

- PHB/ETU: LEM FIR Accumulator Register, page 120.
- PHB/ETU: LEM FIR AND Mask Register, page 125.
- PHB/ETU: LEM FIR OR Mask Register, page 126.
- PHB/ETU: LEM Error Mask Register, page 127.
- PHB/ETU: LEM Error AND Mask Register, page 127.
- PHB/ETU: LEM Error OR Mask Register, page 128.
- PHB/ETU: LEM Action 0 Register, page 128.
- PHB/ETU: LEM Action 1 Register, page 129.
- PHB/ETU: LEM WOF Register, page 129.
- PHB/ETU: Performance Monitor Configuration Register, page 169.
- PHB/ETU: Performance Monitor Counter Register 0, page 173.
- PHB/ETU: Performance Monitor Counter Register 1, page 173.
- PHB/ETU: Performance Monitor Counter Register 2, page 174.
- PHB/ETU: Performance Monitor Counter Register 3, page 174.

4.4 ETU/RSB HV Registers

This section describes the internal configuration registers specific to a PHB4, that are used for initialization, control, and debug of the PHB4 function. These registers are accessed through the RSB block of the ETU_RSB_MAC macro and mapped directly into the chip address space. Refer to the address map in *Figure 4-1* on page 57. The hypervisor accessible register set will also be described.

The PCI Configuration Operations to the PCIe link are controlled via the indirect CONFIG_ADDRESS and CONFIG_DATA registers. Accesses to CONFIG_DATA will generate PCI configuration cycles to the PCIe link. It does not have access to the local root complex PCI CFG space of the PHB4. The local space is directly mapped into the PHB4 address space, refer to *PHB4 Unified Address Space* on page 56. All PCI Configuration Space registers are 4-byte aligned and are little-endian byte ordered.

The local, direct mapped/root complex registers only allow 4-byte length and aligned access from SCOM or AIB/MMIO interfaces. 'Remote' PCI registers are accessed via PCI CFG operations sent on the PCIe link and software/firmware must use the CONFIG_DATA register to access them. The CONFIG_DATA access will allow 1, 2, 3, or 4-byte lengths. The length and address alignment should not cross a 4-byte boundary.

MMIO Load/Store accesses to PHB4 registers that do not exist are handled by ignoring MMIO Stores and returning data of all 1’s for MMIO Loads. All PHB4 configuration register accesses via the AIB or SCOM bus must have a length of 8 bytes and be aligned on an 8 byte address boundary. The only exception to this rule is the CONFIG_DATA register which is 4-bytes in length and must be accessed with 1, 2, 3, or 4-byte lengths. SCOM access to the CONFIG_DATA can only be 4 bytes. The CONFIG_DATA access cannot cross a 4 byte address plus length boundary. All PHB4 registers with the exception of CONFIG_DATA are big-endian byte ordered.

4.4.1 ETU/RSB HV Register Address Map

Table 4-6. ETU/RSB HV Register Address Map (Page 1 of 5)

Offset	Description	Page
Fundamental Registers Set A (FA)		
0x0000 to 0x00F8	Reserved (unused, will return all 1’s when read)	
0x0100	LSI SourceID Register	67
0x0108	Reserved (unused, will return all 1’s when read)	
0x0110	DMA Channel Status Register	68
0x0118	Reserved (unused, will return all 1’s when read)	
0x0120	Processor Load/Store Status Register	68
0x0128	Reserved (unused, will return all 1’s when read)	
0x0130	CONFIG_DATA	69
0x0138	Lock0 Register	70
0x0140	CONFIG_ADDRESS	71
0x0148	Lock1 Register	72
0x0150 to 0x0158	Reserved (unused, will return all 1’s when read)	

Table 4-6. ETU/RSB HV Register Address Map (Page 2 of 5)

Offset	Description	Page
0x0160	PHB4 Configuration Register	73
0x0168	RTT Base Address Register	74
0x0170 to 0x0180	Reserved (unused, will return all 1's when read)	
0x0188	PELT-V Base Address Register	74
0x0190 to 0x0198	Reserved (unused, will return all 1's when read)	
0x01A0	M32 Starting Address Register	75
0x01A8	PEST Base Address Register	75
0x01B0	CRW Base Address Register	76
0x01B8	CRW Gathering Control Register	77
0x01C0	ASN Compare/Mask Register	78
0x01C8	CAPI Compare/Mask Register	79
0x01D0	M64 Outbound Address AND/OR Mask Register (vA4.2)	80
0x01D8 to 0x01E8	Reserved (unused, will return all 1's when read)	
0x01F0	PCIe Outbound Request Upper Address Register	80
0x01F8	Non-Translate Prefix Register	82
0x0200	DMA Read Sync Register	83
0x0208	RTC Invalidate Register	84
0x0210	TCE Kill Register	85
0x0218	Speculation Control Register	86
0x0220	IODA Table Address Register	87
0x0228	IODA Table Data Register	89
0x0230 to 0x0248	Reserved (unused, will return all 1's when read)	
0x0250	PHB4 General Capabilities Register	103
0x0258	PHB4 TCE Capabilities Register	104
0x0260	PHB4 Interrupt Capabilities Register	105
0x0268	PHB4 EEH Capabilities Register	105
0x0270 to 0x02A8	Reserved (unused, will return all 1's when read)	
0x02B0	PAPR Error Injection Control Register	106
0x02B8	PAPR Error Injection Address Register	108
0x02C0	PAPR Error Injection Address Mask Register	108
0x02C8	ETU Error Summary Status Register	109
0x02D0 to 0x02F8	Reserved (unused, will return all 1's when read)	
0x0300	Interrupt Notify Base Address Register	110

Table 4-6. ETU/RSB HV Register Address Map (Page 3 of 5)

Offset	Description	Page
0x0308	Interrupt Notify Base Index Register	110
0x0310 to 0x07F8	Reserved (unused, will return all 1's when read)	
Fundamental Register Set B (FB)		
0x0800	PHB4 - Version Register	111
0x0808	Reserved (unused, will return all 1's when read)	
0x0810	PHB4 - Control Register	111
0x0818 to 0x0858	Reserved (unused, will return all 1's when read)	
0x0860	PHB4 - AIB Fence Control Register	114
0x0868	PHB4 - TCE Tag Enable Register	115
0x0870	PHB4 - TCE Tag Watermark Register	116
0x0878	PHB4 - Timeout Control Register 1	117
0x0880	PHB4 - Timeout Control Register 2	118
0x0888	PHB4 - Quiesce DMA Register	119
0x0890 to 0x0900	Reserved (unused, will return all 1's when read)	
0x0908	PHB4 - TCE Tag Status Register	120
0x0910 to 0x0BF8	Reserved (unused, will return all 1's when read)	
Error Registers		
0x0C00	LEM FIR Accumulator Register	120
0x0C08	LEM FIR AND Mask Register	125
0x0C10	LEM FIR OR Mask Register	126
0x0C18	LEM Error Mask Register	127
0x0C20	LEM Error AND Mask Register	127
0x0C28	LEM Error OR Mask Register	128
0x0C30	LEM Action 0 Register	128
0x0C38	LEM Action 1 Register	129
0x0C40	LEM WOF Register	129
0x0C48 to 0x0C78	Reserved (unused, will return all 1's when read)	
0x0C80	PHB4 Error Status Register	130
0x0C88	PHB4 First Error Status Register	132
0x0C90	PHB4 Error Injection Register	132
0x0C98	PHB4 Error LEM Report Enable Register	133
0x0CA0	PHB4 Error System Interrupt Enable Register	133
0x0CA8	PHB4 Error EEH Freeze Enable Register	133

Table 4-6. ETU/RSB HV Register Address Map (Page 4 of 5)

Offset	Description	Page
0x0CB0	PHB4 Error AIB Fence Enable Register	134
0x0CB8	Reserved (unused, will return all 1's when read)	
0x0CC0	PHB4 Error Log Register 0	134
0x0CC8	PHB4 Error Log Register 1	134
0x0CD0	PHB4 Error Status Mask Register	135
0x0CD8	PHB4 First Error Status Mask Register	135
0x0CE0 to 0x0CF8	Reserved (unused, will return all 1's when read)	
0x0D00	TXE Error Status Register	136
0x0D08	TXE First Error Status Register	140
0x0D10	TXE Error Injection Register	140
0x0D18	TXE Error LEM Report Enable Register	142
0x0D20	TXE Error System Interrupt Enable Register	142
0x0D28	TXE Error EEH Freeze Enable Register	142
0x0D30	TXE Error AIB Fence Enable Register	143
0x0D38	Reserved (unused, will return all 1's when read)	
0x0D40	TXE Error Log Register 0	143
0x0D48	TXE Error Log Register 1	143
0x0D50	TXE Error Status Mask Register	144
0x0D58	TXE First Error Status Mask Register	144
0x0D60 to 0x0D78	Reserved (unused, will return all 1's when read)	
0x0D80	RXE_ARB Error Status Register	145
0x0D88	RXE_ARB First Error Status Register	148
0x0D90	RXE_ARB Error Injection Register	149
0x0D98	RXE_ARB Error LEM Report Enable Register	150
0x0DA0	RXE_ARB Error System Interrupt Enable Register	150
0x0DA8	RXE_ARB Error EEH Freeze Enable Register	150
0x0DB0	RXE_ARB Error AIB Fence Enable Register	151
0x0DB8	Reserved (unused, will return all 1's when read)	
0x0DC0	RXE_ARB Error Log Register 0	151
0x0DC8	RXE_ARB Error Log Register 1	151
0x0DD0	RXE_ARB Error Status Mask Register	152
0x0DD8	RXE_ARB First Error Status Mask Register	152
0x0DE0 to 0x0DF8	Reserved (unused, will return all 1's when read)	
0x0E00	RXE_MRG Error Status Register	153

Table 4-6. ETU/RSB HV Register Address Map (Page 5 of 5)

Offset	Description	Page
0x0E08	RXE_MRG First Error Status Register	156
0x0E10	RXE_MRG Error Injection Register	156
0x0E18	RXE_MRG Error LEM Report Enable Register	156
0x0E20	RXE_MRG Error System Interrupt Enable Register	157
0x0E28	RXE_MRG Error EEH Freeze Enable Register	157
0x0E30	RXE_MRG Error AIB Fence Enable Register	157
0x0E38	Reserved (unused, will return all 1's when read)	
0x0E40	RXE_MRG Error Log Register 0	158
0x0E48	RXE_MRG Error Log Register 1	158
0x0E50	RXE_MRG Error Status Mask Register	158
0x0E58	RXE_MRG First Error Status Mask Register	159
0x0E60 to 0x0E78	Reserved (unused, will return all 1's when read)	
0x0E80	RXE_TCE Error Status Register	159
0x0E88	RXE_TCE First Error Status Register	161
0x0E90	RXE_TCE Error Injection Register	162
0x0E98	RXE_TCE Error LEM Report Enable Register	163
0x0EA0	RXE_TCE Error System Interrupt Enable Register	163
0x0EA8	RXE_TCE Error EEH Freeze Enable Register	163
0x0EB0	RXE_TCE Error AIB Fence Enable Register	164
0x0EB8	Reserved (unused, will return all 1's when read)	
0x0EC0	RXE_TCE Error Log Register 0	164
0x0EC8	RXE_TCE Error Log Register 1	164
0x0ED0	RXE_TCE Error Status Mask Register	165
0x0ED8	RXE_TCE First Error Status Mask Register	165
0x0EE0 to 0x0F78	Reserved (unused, will return all 1's when read)	
Debug Registers		
0x0F80	PHB4 - Trace Control Register	166
0x0F88	Performance Monitor Configuration Register	169
0x0F90	Performance Monitor Counter Register 0	173
0x0F98	Performance Monitor Counter Register 1	173
0x0FA0	Performance Monitor Counter Register 2	174
0x0FA8	Performance Monitor Counter Register 3	174
0x0FB0 to 0x0FF8	Reserved (unused, will return all 1's when read)	

4.4.2 Fundamental Register Set A

4.4.2.1 LSI SourceID Register

This register sets the base source ID for LSI interrupts. The LSI and MSI interrupts share the same source ID pool of source numbers. Therefore, the total number of MSI interrupts is always the maximum minus eight because there are eight LSI interrupts. The LSI Source ID bit numbering in this register supports 4K interrupts.

Mnemonic LSISRCID

Address Offset 0x0100

Bit	Field Mnemonic	Type	Reset Value	Description
0:3	Reserved	RO	0	Reserved.
04:12	LSI Source ID(00:08)	RW	0x1FF	This is the base source ID for LSI interrupts. The default is to use a base of all ones so the LSI interrupt uses the upper eight interrupt source numbers in the total range.
13:15	LSI Source ID(09:11)	R0	0	The PHB4 supports eight LSI interrupts. The interrupt levels 0 - 7 are as follows. These represent the XIVEs in the LXIVT. Level Description 0: PCI INTA 1: PCI INTB 2: PCI INTC 3: PCI INTD 4: Unused 5: Unused 6: Error interrupts (INF) 7: Error interrupts (ER)
16:63	Reserved	RO	0	Reserved.

4.4.2.2 DMA Channel Status Register

This register contains status information for the DMA Channel (upbound toward the processor).

Mnemonic DMACSR

Address Offset 0x0110

Bit	Field Mnemonic	Type	Reset Value	Description
0:26	Reserved Legacy Bits	RO	0	Reserved.
27	Any Error Detected	ROH	0	One of the PHB4's error status register bits is set (PHB4/Outwardbound/INA0/INA1).
28	Any First Error Detected	ROH	0	One of the PHB's first error status register bits is set (PHB4/Outwardbound/INA0/INA1).
29	Any Endpoint is MMIO or DMA Frozen	ROH	0	Any endpoint in the PESTA or PESTB is in the freeze state.
30:63	Reserved	RO	0	Reserved.

4.4.2.3 Processor Load/Store Status Register

This register contains status information for Processor Load/Store requests (MMIO downward bound from the processor).

Mnemonic PLSSR

Address Offset 0x0120

Bit	Field Mnemonic	Type	Reset Value	Description
0:26	Reserved Legacy Bits	RO	0	Reserved.
27	Any Error Detected	ROH	0	One of the PHB4's error status register bits is set (PHB4/Outbound/INA0/INA1).
28	Any First Error Detected	ROH	0	One of the PHB3's first error status register bits is set (PHB4/Outbound/INA0/INA1).
29	Any Endpoint is MMIO or DMA Frozen	ROH	0	Any endpoint in the PESTA or PESTB is in the freeze state.
30:63	Reserved	RO	0	Reserved.

4.4.2.4 CONFIG_DATA Register

This register defines the data to write or the read result data from the PCI Configuration Space registers. It acts as a dedicated I/O port to these registers. Used in conjunction with the CONFIG_ADDRESS Register, it performs configuration cycle accesses to local, Type 0, or Type 1 registers. Reads or writes to this register cause a PCI configuration cycle access to occur to the root complex (local) or a Type 0/1 access to the PCI configuration space. Type 0 and type 1 accesses are sent as configuration cycles on the PCIe link.

Additional information about this register:

- Software must first set up the CONFIG_ADDRESS Register. It can then perform a read or write to the CONFIG_DATA Register to perform a configuration cycle read or write operation respectively.
- Accesses to the CONFIG_DATA Register when the CONFIG_ADDRESS enable bit is zero returns data of all ones on reads. Writes to this register are ignored if the enable bit is zero.
- If the PHB4 enters the Fenced state while a CFG access is active, the current access is terminated. This is to remove a possible deadlock condition with the ASB interface. If that current access was a read, an all ones response is returned.
- Accesses to the CONFIG_DATA Register when the PHB4 is fenced returns all ones on reads. Writes to this register are ignored if the PHB4 is fenced.
- The CONFIG_DATA Register is 4 bytes in length and must be accessed with lengths of 1, 2, 3, or 4 bytes. The length of the request, when added to the “start address” of the first valid byte, must not cross the naturally aligned 4-byte boundary. Accesses of any invalid size sets MMIO Error Status bit 25 (CFG Size Error).
- The CONFIG_DATA Register follows little-endian byte order. Byte swapping commands (load/store/reverse) must be used in the firmware code when accessing this register if the system is running with the processor in big-endian mode.
- If this register is addressed using 0x134 - 0x137, the address is aliased to 0x130 - 0x133 respectively. However, PHB4 error bit 44 (Register Address Error) is set.
- PHB4 logic uses the access to this register to calculate the PCIe byte enables. This register should be accessed on the same byte address that matches the first valid byte address of the PCIe configuration register being addressed. For example, to read the CFG Vendor ID register (cfg space address 0x00), this register is accessed with a 2-byte read to address 0x0130. To read the CFG Device ID Register (cfg space address 0x02), this register is accessed with a 2-byte read to address 0x0132. All the valid access combinations are:
  - To address 0x0130, for lengths of 1, 2, 3, 4 bytes
  - To address 0x0131 for lengths of 1, 2, 3 bytes
  - To address 0x0132, for lengths of 1, 2 bytes
  - To address 0x0133, for length of 1 byte

Mnemonic CONFIG_DATA

Address Offset 0x0130

Bit	Field Mnemonic	Type	Reset Value	Description
0:7	PCI Data(7:0)	RWH	0	The enable bit must be set in the CONFIG_ADDRESS Register for access to this register to be accepted. If the enable bit is zero, all writes are ignored and reads returns all zones. No configuration cycles occur. Reads to this register cause a configuration cycle read access to occur and the data from the configuration cycle access is returned. Writes to this register cause a configuration cycle write access to occur.
8:15	PCI Data(15:8)	RWH	0	Same as PCI Data(7:0).
16:23	PCI Data(23:16)	RWH	0	Same as PCI Data(7:0).
24:31	PCI Data(31:24)	RWH	0	Same as PCI Data(7:0).
32:63	Reserved	RO	0	Reserved.

4.4.2.5 Lock0 Register

This register acts as a semaphore so that mutual exclusion is maintained on a set of registers. Enforcement of mutual exclusion is the responsibility of software. The hardware does not guarantee or enforce mutual exclusion. This register simply implements an accessible state bit in hardware via MMIO accesses. Although this lock is intended to be used for a specific critical region, software can use it for any purpose.

Mnemonic LOCK0

Address Offset 0x0138

Bit	Field Mnemonic	Type	Reset Value	Description
0	Lock Bit	RW	0	To request the lock, the system software must issue a read to this register. If the read value is ‘0’, the lock is granted and the state changes to ‘1’. If the read value is ‘1’, the lock request is denied because it was already locked by a prior reader. The owner of the lock must eventually release the lock by writing a value of ‘0’ to this bit so that it can be granted to the next reader. The owner can also rewrite a value of ‘1’ to this bit to update the data bits and still retain the lock. Read value: • 0 - Lock is obtained by the reader, granted. • 1 - Lock is denied. It is already locked by another reader.
1:63	Data Bits	RW	0	These bits are scratch-pad register bits that can be used for any purpose. Hardware does not modify these bits in any way. They are read/written only via software. These bits can be used with the lock as a mailbox mechanism for interprocess communication.

4.4.2.6 CONFIG_ADDRESS Register

This register defines the address used to access the PCI Configuration Space registers. Used in conjunction with the CONFIG_DATA register, it performs configuration cycle accesses to local, Type 0, or Type 1 registers. Reads or writes to this register cause no side effects.

The enhanced PCIe configuration access mechanism, defined in the PCIe specification, defines a flat memory-mapped address space for accessing the PCI configuration space registers. This method can be ignored. The CONFIG_ADDRESS and CONFIG_DATA indirect access method replaces this mechanism.

Additional information about this register is as follows:

- The enable bit must be set for any configuration cycle to occur. If an access to the CONFIG DATA Register is started with the Enable bit = 0, then an error will be flagged.
- This register cannot be used to access local root complex PCI Configuration Space.
- Type 0 configuration cycle accesses are sent to PCIe link if the Bus Number field matches the secondary bus number configured in the local configuration space of the root complex.
- Type 1 configuration cycle accesses are sent to the PCIe link if the Bus Number is greater than the secondary bus number configured in the local configuration space for root complex.
- Configuration accesses attempted with a Bus Number less than the primary Bus Number or greater than the subordinate Bus Number are treated as errors and no configuration cycle access occurs. These bus numbers are considered out-of-range for the configured PCI bus.

Mnemonic CONFIG_ADDRESS

Address Offset 0x0140

Bit	Field Mnemonic	Type	Reset Value	Description
0	Enable	RWH	0	This bit must be set for any configuration cycle to occur.
01:03	CFGOP Completion Status Code(2:0)	ROH	0	PCIe Completion Status code bits CS(2:0). These bits are updated by the hardware and reflect the status of the last CFGOP. CS(2:0) (should match PCIe specification encoding): • '000' – good, successful completion. • '001' – UR, unsupported request. • '010' – CFG retry status. • '100' – CA, completer abort. • Other encodes are reserved and are not defined.
04:11	Bus Number(0:7)	RW	0	Bus Number to use for the configuration cycle access.
12:16	Device Number(0:4)	RW	0	Device Number to use for the configuration cycle access.
17:19	Function Number(0:2)	RW	0	Function Number to use for the configuration cycle access.
20:29	Register Number(00:09)	RW	0	Register Number to use for the configuration cycle access. This allows access to a 4K PCI Configuration Space.
30:31	Register Number(10:11)	RO	0	Don't Cares. The value written will be dropped, and zeros are always returned for a read. The actual address offset and length of the access to the CONFIG_DATA register are used to calculate the CFG byte enables.
32:38	Reserved	RO	0	Reserved.
39:47	PE Number(0:8)	RW	0	PE number to use for this CFG access.
48:63	Reserved	RO	0	Reserved.

4.4.2.7 Lock1 Register

This register acts as a semaphore so that mutual exclusion is maintained on a set of registers. Enforcement of mutual exclusion is the responsibility of software. The hardware does not guarantee or enforce mutual exclusion. This register simply implements an accessible state bit in hardware via MMIO accesses. Although this lock is intended to be used for a specific critical region, software can use it for any purpose.

Mnemonic LOCK1

Address Offset 0x0148

Bit	Field Mnemonic	Type	Reset Value	Description
0	Lock Bit	RW	0	To request the lock, the system software must issue a read to this register. If the read value is ‘0’ the lock is granted and the state will change to ‘1’. If the read value is ‘1’ then the lock request is denied since it was already locked by a prior reader. The owner of the lock must eventually release the lock by writing a value of ‘0’ to this bit so that it can be granted to the next reader. The owner can also rewrite a value of ‘1’ to this bit to update the Data Bits and still retain the lock. Read value: • 0 - Lock is obtained by the reader, granted. • 1 - Lock is denied. It is already locked by another reader.
01:63	Data Bits	RW	0	These are scratch-pad register bits. They can be used for any purpose. Hardware does not modify these bits in any way. They are read/written only via software. These can be used with the lock as a mailbox mechanism for interprocess communication.

4.4.2.8 PHB4 Configuration Register

This register controls specific internal PHB4 functions.

Mnemonic PHB_CR

Address Offset 0x0160

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	Reserved	RO	0	Reserved.
8	32-bit MSI Support Enable	RW	0	This bit when set to ‘1’ enables 32-bit inbound memory write operations with specific address requirements to be interpreted as MSI interrupts. When disabled, all 32-bit inbound memory write operations are treated as normal DMA operations. These are the PCI address bits of interest for a 32-bit MSI interrupt. PCI Addr(31:16) = 0xffff (high order 16 bits must be all ones).
09:13	Reserved	RO	0	Reserved.
14	64-bit MSI Support Enable	RW	1	This bit when set to ‘1’ enables 64-bit inbound memory write operations with specific address requirements to be interpreted as MSI interrupts. When disabled, all 64-bit inbound memory write operations are treated as normal DMA operations. These are the PCI address bits of interest for a 64-bit MSI interrupt. PCI Addr(61:60) = b”01”.
15	Reserved	RO	0	Reserved.
16	Reserved	RW	0	Reserved, but implemented for future use.
17:63	Reserved	RO	0	Reserved.

4.4.2.9 RTT Base Address Register

This register specifies the base system address for the RID Translation Table (RTT) in memory.

Note: The PE number is right justified within the 2-byte entry.

Mnemonic RTT_BAR

Address Offset 0x0168

Bit	Field Mnemonic	Type	Reset Value	Description
0	RTT BAR Enable	RW	0	Must be set to '1' to use this BAR. If an operation attempts to use this BAR and it is not enabled, an error is flagged.
1:13	Reserved	RO	0	Reserved.
14:46	Base Address(14:46)	RW	0	This is the 50-bit system base address. This address is aligned on a 64 K × 2 B = 128 KB boundary (bits 47:63 of the address are implied zeros). This supports all 16 bits of the RequesterID, two bytes in each RTE entry.
47:63	Reserved	RO	0	Reserved.

4.4.2.10 PELT-V Base Address Register

This register specifies the base system address for the PE lookup table vector (PELT-V) in memory.

Mnemonic PELTV_BAR

Address Offset 0x0188

Bit	Field Mnemonic	Type	Reset Value	Description
0	PELT-V BAR Enable	RW	0	Must be set to '1' to use this BAR. If an operation attempts to use this BAR and it is not enabled, an error is flagged.
1:13	Reserved	RO	0	Reserved.
14:50	Base Address(14:50)	RW	0	This is the 50-bit system base address. **Note:** This address is aligned on a 256 × 32 bytes = 8 KB boundary (bits 51:63 of the address are implied zeros). This supports all 256 PEs, 32 bytes (256 bits) in each PELT-V entry.
51:63	Reserved	RO	0	Reserved.

4.4.2.11 M32 Starting Address Register

This register is used for MMIO address translation. This register is used as the new base for the PCI address. It is combined with the original offset from the system address to create a new PCI address. This new address is sent to the PCIe link.

Mnemonic M32_SAR

Address Offset 0x01A0

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	Reserved	RO	0	Reserved.
32:51	M32 Starting Address(32:51)	RW	0	This is the base address to be combined with the offset from the system address. Depending on the size of the region, the appropriate rightmost bits of this register need to be set to zeros since the value in this register is logically ORed with the offset. Refer to *Figure 3-3 M64 Address Translation* on page 40.
52:63	Reserved	RO	0	Reserved.

4.4.2.12 PEST Base Address Register

This register specifies the base system address for the PE State Table (PEST) in memory.

Mnemonic PEST_BAR

Address Offset 0x01A8

Bit	Field Mnemonic	Type	Reset Value	Description
0	PEST BAR Enable	RW	0	This bit must be set to ‘1’ to use this BAR. If an operation attempts to use this BAR and it is not enabled, an error is flagged.
1:07	Reserved	RO	0	Reserved.
08:51	Base Address(08:51)	RW	0	This is the 56-bit system base address. This address is aligned on a 256 x1 6 byte = 4KB boundary (bits 52:63 of the address are implied zeros). This supports 256 endpoints, 16 bytes in each PEST entry. For 512 PEs, bits 51:63 are implied zeros and bit 51 from the Base Address is not used.
52:63	Reserved	RO	0	Reserved.

4.4.2.13 CRW Base Address Register

This register specifies the base system address for the Credit Return Write (CRW) commands. The CRW commands are sent for credit returns in support of the CAPI, Pseudo-VC mechanism.

Mnemonic CRW_BAR

Address Offset 0x01B0

Bit	Field Mnemonic	Type	Reset Value	Description
00:61	CRW Base Address(00:61)	RW	0	This value is used as is for the PCIe request address in the CRW, posted write command. This value should match the expected value for a CRW in the CAPI endpoint. Firmware must ensure this value is correct. The hardware does not check if this value is correct. It just blindly uses the current value in this register as the PCIe address for the CRW command.
62	Reserved	RW	0	Reserved.
63	CRW Function Enable	RW	0	Set to '1' to enable the PHB4 to send CRW commands. CRWs will never be sent when this bit is '0'.

4.4.2.14 CRW Gathering Control Register

This register controls the 'gathering' function for the Credit Return Write (CRW) commands. The CRW commands are sent for credit returns in support of the CAPI, Pseudo-VC mechanism.

Mnemonic CRW_GATHER

Address Offset 0x01B8

Bit	Field Mnemonic	Type	Reset Value	Description
0	CRW Gather Enable	RW	0	Must be set to '1' to enable CRW gathering function. If '0', CRWs will be sent as soon as possible. The logic will not attempt to gather CRWs. New 'normal' DMA write commands will pause until a pending CRW has been sent if one is pending.
01:12	Reserved	RO	0	Reserved.
13:15	CRW Gather Num(0:2	RW	0	This field specifies the number of CRW commands to gather before sending to the CAPI endpoint as a single CRW command. New 'normal' DMA write commands will pause until a pending CRW (gathered number) has been sent if one is pending. '000' - reserved, behavior not defined. '001' - gather 1 CRW (this is the same as if gathering was not enabled). '010' - gather 2 CRWs. '011' - gather 3 CRWs. '100' - gather 4 CRWs. '101' - reserved, behavior not defined. '110' - reserved, behavior not defined. '111' - reserved, behavior not defined.
16:17	Reserved	RO	0	Reserved.
18:23	CRW Gather Watch-dog Timer Value(0:5)	RW	0	This field specifies the 'watchdog timer' value for the gathering function. The watchdog timer determines how long to wait before attempting to gather the required number of CRWs as specified in the CRW Gather Num(0:2) field. If the timer elapses, the function will attempt to send the current number of CRWs it has accumulated up to that point. A value of 0x20 will disable the timer, thus it will wait forever to accumulate CRWs until it accumulates the required number. The configuration value of the timeout will always represent a minimum/maximum timeout range: • Minimum timeout value is: $2^{(\text{timer value})}$ x 32ns. • Maximum timeout value is: $2^{(\text{timer value})}$ x 48ns.
24:63	Reserved	RO	0	Reserved.

4.4.2.15 ASN Compare/Mask Register

This register supports the CAPI AS_Notify and CAPI Atomic tunneling commands over the PCIe link. These will be referred to as 'ASN' commands. The PHB/Root Complex must decode an ASN versus a normal write using address bits within the posted write command. This register specifies the compare and mask values to apply to inbound posted write addresses.

This is a firmware programmable register to match expected 16 bit value for ASN command.

- 16 bit compare value, CMP(63:48), 16 bit mask value, MSK(63:48).
- Compare and mask values applied to upper 16 bits of TLP header address (addr(63:48)).
- First mask off address bits that do not contribute to compare, then compare.
- The final result will indicate a packet is a ASN='1' versus a normal write='0'.

Example:

```
CMP(63:48) = 0xAB00
MSK(63:48) = 0xFF00
Addr(63:48) = 0xAB13
and_mask(15:0) = (0xAB13 and 0xFF00) = 0xAB00
xor(15:0) = (0xAB00 xor 0xAB00) = 0x0000
ASN = not or_reduce(xor(15:0)) = '1' (packet is an ASN)
```

Compare and mask values chosen must match expected ASN bits driven by the CAPI device.
The PHB4 will assert the 'ASN Bit' for DMA writes that match this register. See the *Table 3-6 AIB Attribute Field Summary* on page 32.

Mnemonic ASN_CMPM

Address Offset 0x01C0

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	CMP(63:48)	RW	0	Compare Value.
16:31	MSK(63:48)	RW	0	Mask Value.
32:62	Reserved	RO	0	Reserved.
63	ASN Enable	RW	0	Set to '1' to enable ASN decoding. If '0', ASN will not be decoded.

4.4.2.16 CAPI Compare/Mask Register

'CAPI' specific DMA Write commands are identified by their most significant PCI address bits. Firmware must configure this register to match the correct address bits for CAPI specific commands.

This is a firmware programmable register to match expected 16 bit value for CAPI command:

- 16 bit compare value, CMP(63:48), 16 bit mask value, MSK(63:48).
- Compare and mask values applied to upper 16 bits of TLP header address (addr(63:48)).
- First mask off address bits that do not contribute to compare, then compare.
- The final result will indicate a packet is a CAPI='1' versus a normal write='0'.

Example:

```
CMP(63:48) = 0xAB00
MSK(63:48) = 0xFF00
Addr(63:48) = 0xAB13
and_mask(15:0) = (0xAB13 and 0xFF00) = 0xAB00
xor(15:0) = (0xAB00 xor 0xAB00) = 0x0000
CAPI = not or_reduce(xor(15:0)) = '1' (packet is CAPI)
```

Compare and mask values chosen must match expected CAPI bits driven by the CAPI device.

The PHB4 will assert the 'CAPI Bit' for DMA writes that match this register. See the *Table 3-6 AIB Attribute Field Summary* on page 32.

The CAPI command is not processed in the ETU, it is flagged as a CAPI command via the AIB attribute. The ETU processes the command as a normal DMA write command. The PB logic above the ETU will process the CAPI command and issue the correct Powerbus command sequences based on the CAPI architecture.

Mnemonic CAPI_CMPM

Address Offset 0x01C8

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	CMP(63:48)	RW	0	Compare Value.
16:31	MSK(63:48)	RW	0	Mask Value.
32:62	Reserved	RO	0	Reserved.
63	CAPI Enable	RW	0	Set to '1' to enable CAPI decoding. If '0', CAPI will not be decoded.

4.4.2.17 M64 Outbound Address AND/OR Mask Register (vA4.2)

This is a firmware programmable register to modify upper bits of all M64, MMIO addresses.

- AND/OR control and mask bits, defaults to current behavior.
- Global for all M64 **MMIO** commands.
- M64 address modified after MBT/MDT lookup.

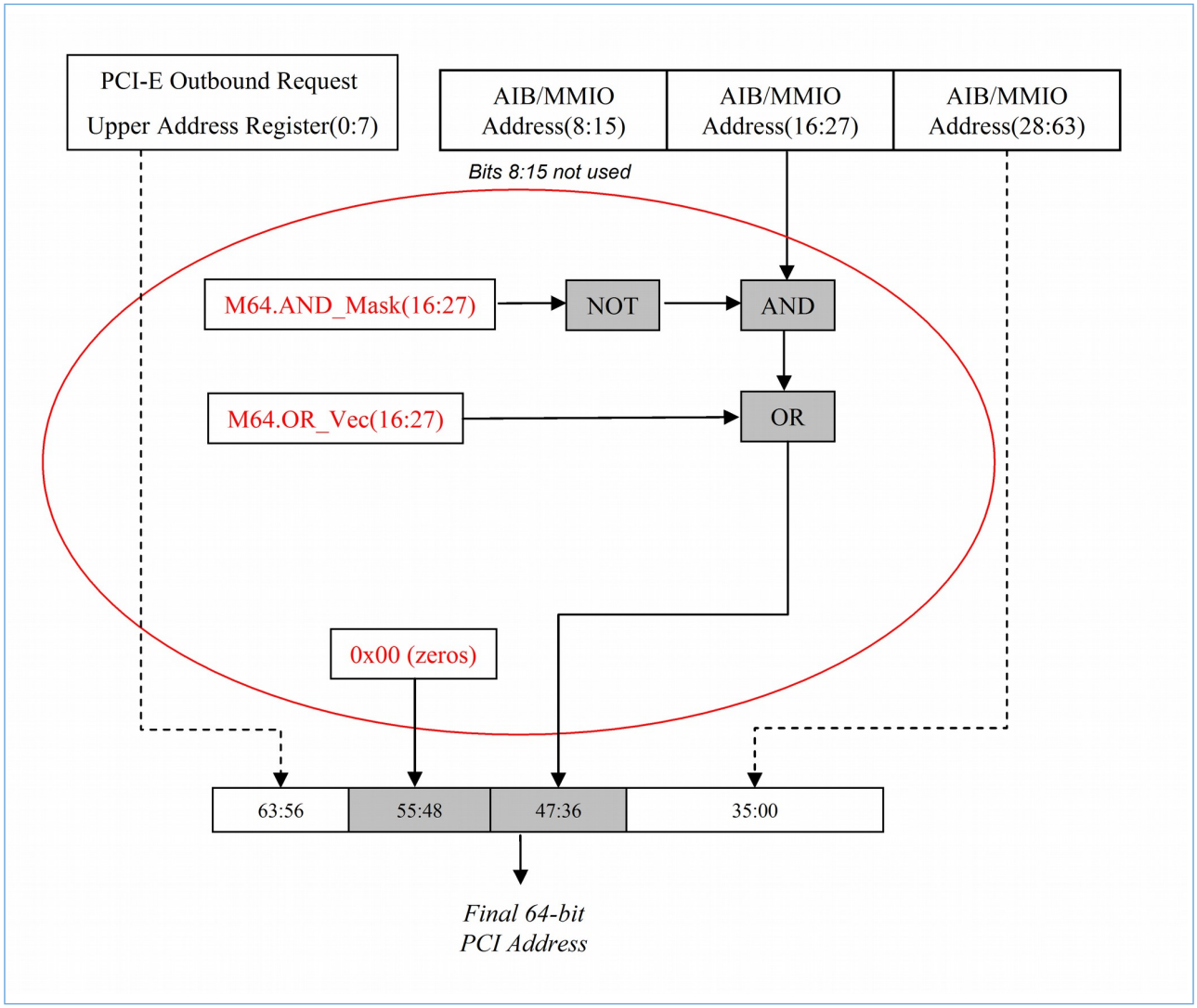
Mnemonic M64_AOMASK

Address Offset 0x01D0

Bit	Field Mnemonic	Type	Reset Value	Description
0	Enable	RW	0	Enable Bit. If set to '1', will apply the AND and OR mask values in this register to all M64, Outbound MMIO Addresses before the MMIO request is sent to the PCIe link. This does not affect the MBT/MDT lookup path in any way. These masks are applied after the MBT/MDT steps and right before the request is sent to the PCIe link. Refer to *Figure 4-3* on page 81. If '0' (default), M64 addresses will not be modified by the AND or the OR masks in this register.
01:07	Reserved	RO	0	Reserved.
08:15	AND Mask Bits (08:15)	RO	0xFF	These are the AND mask bits that correspond to PCIe Address(55:48) bits. This vector is hard-coded with each bit set to '1' so it will zero out the original bits from the AIB/MMIO address (08:15) bits for all MMIO requests that match in the MBT as M64 space. M32 space MMIO is never modified.
16:27	AND Mask Bits (16:27)	RW	0	These are the AND mask bits that correspond to PCIe Address(47:36) bits. Bits set to '1' in this vector will zero out the original bits from the AIB/MMIO address(16:27) bits for all MMIO requests that match in the MBT as M64 space. M32 space MMIO is never modified. This field defaults to all zeros which means by default the original MMIO bits pass through unmodified.
28:47	Reserved	RO	0	Reserved.
48:59	OR Vector Bits (16:27)	RW	0	These are the OR vector bits that correspond to PCIe Address(47:36) bits. This vector will be bitwise ORed with the address(16:27) bits after they have been masked with the AND Mask Bits(16:27) and once again only for MMIO requests that match in the MBT as M64 space. M32 space MMIO is never modified. This field defaults to all zeros which means by default the original MMIO bits pass through unmodified.
60:63	Reserved	RO	0	Reserved.

Refer to *Figure 4-3* on page 81 for additional details.

Figure 4-3. M64 - Outbound Address AND/OR Mask Logic Diagram



4.4.2.18 PCIe Outbound Request Upper Address Register

This register defines the upper address bits to substitute into the outbound request TLP address.

The value in this register is only used when the MMIO address matches the M64 BAR. It only applies to intended 64-bit MMIO addressed requests. Refer to *Figure 3-3 M64 Address Translation* on page 40.

Mnemonic PEOBRQ_UAR

Address Offset 0x01F0

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	Upper Address Bits (63:56)	RW	0	This field defines the address bits (63:56) that will be sent with outbound PCIe requests. This allows software to move the MMIO space anywhere within the 64 bit PCI address space. The PHB4 only receives a 56-bit address from the Power bus. These are bits 08:63 or in PCI numbering, bits 55:0.
08:63	Reserved	RO	0	Reserved.

4.4.2.19 Non-Translate Prefix Register

This register is used in the non-translate/50-bit mode only.

The value in this register is only used when the TVE entry is set up for non-translate/50-bit mode. Refer to *Table 4-15 TVT Data* on page 95.

The Prefix Address Bits are global for all **DMA** commands for the TVE entry when configured in non-translate/50-bit mode.

Mnemonic NXLATE_PREFIX

Address Offset 0x01F8

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	Reserved	RO	0	Reserved.
08:13	Prefix Address Bits(55:50)	RW	0	This field defines the address bits (55:50) that will be sent to the Power bus for inbound DMAs when in the non-translate/50-bit mode. These are address bits 08:13 or in PCI numbering, bits 55:50.
14:63	Reserved	RO	0	Reserved.

4.4.2.20 DMA Read Sync Register

This register is used to guarantee all prior DMA reads have completed after this register is written. This is typically used during purge cache operations to ensure all old DMA Read operations have finished using an old page in memory. Firmware must ensure all accesses to an old page have ceased before reallocating a page to another process or thread.

- Bit 1 will initially return a state of zero after bit 0 is written.
- When bit 0 is written, the hardware generates an internal sync token that is ordered behind all prior DMA Reads in the pipe. When this token reaches the end of the internal pipeline, a snapshot of all current DMA Read tags is stored.
- As each DMA Read operation is completed, the snapshot bit is reset.
- When all snapshot bits are clear, bit 1 is set to 1.
- Once the register is written and the sync sequence started (bit 1 is zero), subsequent writes to this register will have no effect. Only one sync sequence is permitted at any time. After bit 1 is set back to 1, another new sync sequence can be initiated.

Mnemonic DMARD_SYNC

Address Offset 0x0200

Bit	Field Mnemonic	Type	Reset Value	Description
0	Start Sync Bit	WO	0	Writing a 0 to this bit has no effect. Writing a 1 starts the DMA Read synchronization process.
1	Sync Complete Bit	ROH	1	b'0': Indicates that some DMA Read operations that were active when the Start Sync bit was written are still pending. b'1': Indicates that all DMA read operations that were active when the Start Sync bit was written have been completed.
02:63	Reserved	RO	0	Reserved.

4.4.2.21 RTC Invalidate Register

This register provides a mechanism to support Firmware Managed RTC Coherency. Software has the ability to invalidate any potential RID to PE translation that can be cached in the PHB4. Invalidates to non-cached entries have no affect. Writing this register has a side effect of generating an invalidate to the RTC.

This register has a side effect when written to. The act of writing this register will invalidate an entry in the RTC if any entry matches the RequesterID.

Reading this register causes no side effects and will return the last value written.

Mnemonic RTC_INVLDT

Address Offset 0x0208

Bit	Field Mnemonic	Type	Reset Value	Description
O	Invalidate All	RW	0	Setting this bit will invalidate all entries in the RTC regardless of the setting in the rest of this register. The default value of zero will invalidate the entry specified by the RequesterID.
01:15	Reserved	RO	0	Reserved.
16:31	RequesterID(0:15)	RW	0	16-bit PCI RequesterID to invalidate in the RTC.
32:63	Reserved	RO	0	Reserved.

4.4.2.22 TCE Kill Register

This register provides a mechanism to support Firmware Managed TCE Coherency. Software has the ability to invalidate any potential TCE system address that can be cached in the PHB4. Invalidates to non-cached entries have no affect. Writing this register has a side effect of generating an invalidate/kill to the TCE Cache.

This register has a side effect when written to. Writing this register with the operation bits (not = 000b) will cause one or more entries in the TCE Cache to be invalidated, if the specified invalidate values match.

Reading this register causes no side effects and will return the last value written.

Note: It can take multiple cycles to invalidate one or more entries in the TCE cache. The PHB4 has a small internal queue of kill requests to absorb kill requests that are pending completion. If the queue is full then it will not clear the operation bits 0:2 until a free slot is available in the internal queue.

Software and firmware can read the 'TCE Kill Status' bit in the PHB4 - Quiesce DMA Register to test if there are kill operations pending, see *Table 4.4.3.8 PHB4 - Quiesce DMA Register* on page 119.

Mnemonic TCE_KILL

Address Offset 0x0210

Bit	Field Mnemonic	Type	Reset Value	Description
0:2	Invalidate Operation	RWH	0	Defines the operation to be executed. Valid values are: • 1xxb - invalidate entire TCE cache (all entries). • 01xb - invalidate TCEs that match the specified PE#. • 001b - invalidate the TCE for the specified PCIe address and PE#. The hardware will clear these bits to zeros when the invalidate/kill operation is added to the internal kill queue.
3	Page Select Bit	RW	0	When set to '1' this enables the 'Page Select(0:1)' bits below. Firmware must set this to '1' to invalidate a TCE page size larger than 4KB. When '0' the page size defaults to 4KB.
04:51	Invalidate Address(04:51)	RW	0	PCIe address for the specified PE# for which the corresponding direct TCE is to be invalidated, I/O page aligned. This field is required for Invalidate Operation 0b001, and not required for 0b01x or 0b1xx. Invalidate Address(04:51) = PCIe Address(59:12) (original PCIe TLP address). The low order bits of this field are overloaded for page sizes larger than 4KB. This is only valid when the Page Select Bit is a '1'. Invalidate Address(50:51) = Page Select(0:1). 00 – Reserved/4K (if 4K then must set Page Select Bit = '0'). 01 – 64KB. 10 – 2MB. 11 – 1GB.
52:54	Reserved	RW	0	Reserved, but implemented register(s) for future use.

Bit	Field Mnemonic	Type	Reset Value	Description
55:63	Invalidate PE Number(0:8)	RW	0	PE# of the cache entry or entries to invalidate. Required for Invalidate Operations 0b01x and 0b001, and not required for 0b1xx. This PE number supports 512 max PEs. For 256 PEs, bit 55 is a don't care and is not used. PHB4X08 – PE Number(1:8) (256 PEs). PHB4X16 – PE Number(0:8) (512 PEs).

4.4.2.23 Speculation Control Register

This register controls TCE address speculation and RTC table speculation.

A TCE speculative request will only occur if there is a miss in the TCE cache. A speculative request can replace a valid TCE entry in the cache based on the LRU algorithm.

An RTC speculative request will only occur if there is a miss in the RTC cache. A speculative request can replace a valid RTC entry in the cache based on the LRU algorithm.

Mnemonic SPEC_CTL

Address Offset 0x0218

Bit	Field Mnemonic	Type	Reset Value	Description
0	DMA Read TCE Address Speculation Enable	RW	0	Set to ‘1’ to allow inbound TCE address speculation for inbound DMA reads. This function will get an advanced look at the incoming PCI address and will start the TCE address translation and fetch before the request is actually presented on the BLIF interface from the PCIe/PBL core. The goal is to hide as much of the TCE fetch latency as possible to improve overall performance.
1	DMA Write TCE Address Speculation Enable	RW	0	Set to ‘1’ to allow inbound TCE address speculation for inbound DMA writes. This function will get an advanced look at the incoming PCI address and will start the TCE address translation and fetch before the request is actually presented on the BLIF interface from the PCIe/PBL core. The goal is to hide as much of the TCE fetch latency as possible to improve overall performance.
2	DMA Read RTC Speculation Enable	RW	0	Set to ‘1’ to allow inbound RTC speculation for inbound DMA reads. This is similar to TCE speculation except it is for the RTC cache. The RequesterID value from the incoming TLP is used to read the RTT table in memory for the RID to PE translation.
3	DMA Write RTC Speculation Enable	RW	0	Set to ‘1’ to allow inbound RTC speculation for inbound DMA writes. This is similar to TCE speculation except it is for the RTC cache. The RequesterID value from the incoming TLP is used to read the RTT table in memory for the RID to PE translation.
04:05	Reserved	RW	0	Reserved, but implemented register(s) for future use.
6	LRU Speculation Update Disable	RW	0	When set to '1', disables LRU updates on speculation requests only.
7	Reserved	RW	0	Reserved, but implemented register(s) for future use.
08:63	Reserved	RO	0	Reserved.

4.4.2.24 IODA Table Address Register

This register sets the address for accessing one of the architected IODA tables in the PHB4.

Mnemonic IODA_ADDR

Address Offset 0x0220

Bit	Field Mnemonic	Type	Reset Value	Description
0	Auto Increment	RW	0	Set this bit to automatically increment the Table Address field after the IODA Table Data Register 0 is accessed. Default is to not automatically increment the address.
01:10	Reserved	RO	0	Reserved.
11:15	Table Select(0:4)	RW	0	This encoding selects the table to access. See *Table 4-7 IODA Table Summary* on page 88. If an encoding is used that is not listed in the table, a read operation will return random data and a write will be ignored.
16:27	Reserved	RO	0	Reserved.
28:31	MIST Partial Write Vector(0:3)	RW	0	This field is only used to modify the write behavior to the MIST table. It does not affect read behavior or data. It does not affect any other IODA table and is a do not care for other tables. This field allows for a 'partial' write to specific interrupt source 'indexes' in the 64-bit data word of the MIST table entry. See also *Table 4-9 Mist Data* on page 90. Each MIST table entry contains information for four different interrupt sources. Each source has 16 bits of data. This vector allows firmware to only modify less than 4 sources at a time. A typical case is firmware only wanting to modify one source at a time. All 16 bits of the selected interrupt source will be updated when enabled. This feature allows firmware to update one source while not affecting others. It avoids the window of having the hardware updating interrupt source state that firmware does not intend to modify while potentially being overwritten by the firmware update. 0000 – write entire 64-bits, all 4 interrupt sources, of IODA Table Data Register (default). 1111 – same as '0000' case, write entire 64-bit value (no partial). Any other combination except above will be a partial write. Examples: • 1000 – only write source index 0, merge current hardware updates for other sources. • 1101 – write source indexes 0,1,3, merge hardware updates for index 2.
32:53	Reserved	RO	0	Reserved.
54:63	Table Address(0:9)	RW	0	This value defines the starting address for read or write accesses to the selected table. Note this address is incremented after every read or write of the IODA Table Data Register 0 as long as the Auto Increment bit is set. This address will also wrap around if the table selected has an address size smaller than this field.

Table 4-7 is a summary of the IODA table.

Table 4-7. IODA Table Summary (Page 1 of 2)

Table Name	Table Selection (0:4)	Address Bits	Size (# of Entries)	Page
Reserved, unused	00000			
LSI Interrupt State Table (LIST)[1]	00001	3	8	90
MSI Interrupt State Table (MIST)	00010	10	512 (x08) 1024 (x16)	90
Reserved, unused	00011			
Reserved, unused	00100			
RID Translation CAM (RCAM)[1]	00101	7	64 (x08) 128 (x16)	91
Migration Register Table (MRT)	00110	4	8 (x08) 16 (x16)	92
PE State Table A (PESTA)	00111	9	256 (x08) 512 (x16)	92
PE State Table B (PESTB)	01000	9	256 (x08) 512 (x16)	93
TCE Validation Table (TVT)	01001	9	256 (x08) 512 (x16)	95
TCE Cache RAM (TCR)[1]	01010	10	512 (x08) 1024 (x16)	97
TCE Data RAM (TDR)[1]	01011	10	512 (x08) 1024 (x16)	97
Reserved, unused	01100			
Reserved, unused	01101			
Reserved, unused	0110			
TCE Hash Table (THASH)[1]	01111	7	32 (x08) 64 (x16) (each entry is 16B)	100 100
MMIO BAR Table (MBT)	10000	6	16 (x08) 32 (x16) (each entry is 16B)	101 101
MMIO Domain Table (MDT)	10001	9	256 (x08) 512 (x16)	102
Reserved, unused	10010			

Table 4-7. IODA Table Summary (Page 2 of 2)

Table Name	Table Selection (0:4)	Address Bits	Size (# of Entries)	Page
Reserved, unused	10011			
PE Error Vector (PEEV)[1]	10100	3	4 (x08) 8 (x16)	102

1. The LIST, RCAM, TCR, TDR, and THASH are accessible for debug purposes only and are not required to be accessed during PHB4 initialization. In addition, the PEEV is only needed for ER error recovery and is not required during initialization.

4.4.2.25 IODA Table Data Register

This register contains the data for writing to or reading from an IODA table in the PHB4.

A read of this register will generate a read of the table selected and address set in the IODA Table Address Register. The data read from the table is returned with the read operation. Writes to this register will write to the address location pointed to by the IODA Table Address Register. Reads or writes to this register will increment the IODA Table Address Register Table Address by one after each access to this register if the Auto Increment bit is set.

- Partial read/writes to this register are not permitted. All data must be aligned.
- Reads/writes to this register cause the Table Address to be incremented by 1 if the Auto Increment bit is set in that register.
- *Table 4-8* on page 90 through *Table 4-23* on page 102 are the unique definitions for each table's data.

Mnemonic IODA_DATA

Address Offset 0x0228

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	IODA Table Data 0(00:63)	RW	0	Data to write/read result data.

Table 4-8, each LIST entry contains meta data for 1 interrupt source.

Table 4-8. List Data

Bits	Fields	Definitions
00:05	Unused	Reserved, all zeros.
6	'P' bit	'P' state bit for the interrupt. This is also known as the 'Presented' bit. Software and firmware can write this bit to any state 0 or 1 if needed.
7	'Q' bit	'Q' state bit for the interrupt. This is also known as the 'Queued' bit. The hardware will never set this bit to '1'. There is no 'queuing' or 'stacking' of LSI interrupts since they are represented as levels only. Software and firmware can write this bit to any state 0 or 1 if needed. In the LSI case, firmware can write the state PQ=01 (off) to disable the interrupt. If software or firmware sets the state PQ=11 then the internal state machine will behave similar to MSIs on how they treat the Q bit. It is generally not advised to set the state PQ=11 for LSIs.
08:13	Unused	Reserved, all zeros.
14	LSI Level Current State	Current state of the LSI level for this interrupt source. This is a read-only status bit and the hardware updates this based on the current state of the LSI level: • '0' = inactive. • '1' = active.
15	'0'	Reserved, zero. This just pads out the last bit of the second byte for readability.
16:63	Unused	Reserved, all zeros.

Table 4-9, each MIST entry contains meta data for 4 interrupt sources.

Table 4-9. Mist Data (Page 1 of 2)

Interrupt Source Index	Bits	Fields	Definitions
(i^4)+0	0	'P' bit	'P' state bit for the interrupt. This is also known as the 'Presented' bit. Firmware can write this bit to any state 0 or 1 if needed.
(i^4)+0	1	'Q' bit	'Q' state bit for the interrupt (always '0' for LSIs). This is also known as the 'Queued' bit. It indicates that the interrupt source has been 'stacked' with multiple MSI triggers. Firmware can write this bit to any state 0 or 1 if needed.
(i^4)+0	02:03	Unused	Reserved, all zeros. For future PE number expansion.
(i^4)+0	04:15	PE Number(0:11)	PE Number for the current interrupt source. Firmware must set this to the PE assigned for this interrupt. MSI interrupt from PEs that do not match the PE but attempt to use this source will be rejected.

1. The index 'i' is the range of the number of interrupts divided by 4: *i = 0 to N/4-1, where N=max number of interrupt sources*
2. A given PHB4 implementation will use the rightmost bits of the PE number based on the total number of PEs supported in that implementation. Any leftmost/most significant bits remaining of the PE number will be reserved and all zeros.
3. See the MIST Partial Write Vector field(0:3) in the *IODA Table Address Register* on page 87.

Table 4-9. Mist Data (Page 2 of 2)

Interrupt Source Index	Bits	Fields	Definitions
(i^4)+1	16:31	Same definition as above 2 bytes	Same definition as above 2 bytes.
(i^4)+2	32:47	Same definition as above 2 bytes	Same definition as above 2 bytes.
(i^4)+3	48:63	Same definition as above 2 bytes	Same definition as above 2 bytes.

1. The index 'i' is the range of the number of interrupts divided by 4: *i = 0 to N/4-1, where N=max number of interrupt sources*
2. A given PHB4 implementation will use the rightmost bits of the PE number based on the total number of PEs supported in that implementation. Any leftmost/most significant bits remaining of the PE number will be reserved and all zeros.
3. See the MIST Partial Write Vector field(0:3) in the *IODA Table Address Register* on page 87.

Table 4-10. RCAM Data

Bits	Fields	Definitions
0	Valid State Bit	This is a state bit for the RCAM entry. This bit is set when the RCAM entry is allocated. This bit will get cleared after the entry is killed via an RTC invalidate.
1	Pending State Bit	This is a state bit for the RCAM entry. This bit is set when the RCAM entry is allocated indicating that a RTT fetch is pending. This bit is cleared when the RTT data is returned for the fetch.
2	Kill State Bit	This is a state bit for the RCAM entry. This bit is set when an RTC invalidate is received for the entry. This bit will clear when the entry goes invalid (all of the state bits will be zero). This bit will remain asserted if the Pending State Bit is asserted. This allows the entry to remain valid until its RTT fetch response is received. Once the fetch data has been received, the killed entry will go invalid and all state bits will clear.
03:15	Unused	Reserved, all zeros.
16:31	RID(0:15)	This is the match RequesterID (RID) bits for the entry.
32:38	Unused	Reserved, all zeros.
39:47	PE Number(0:8)	This is the cached PE Number for the RID fetched from the RTT table in memory.
48:63	Unused	Reserved, all zeros.

Table 4-11. Migration Register Table (MRT) Data

Bits	Fields	Definitions
0	Valid	0: Invalid. Attempts to use this register will put the PE in the EEH stopped state. 1: Valid. Entry is allowed to be used.
01:07	Unused	Reserved, all zeros.
08:51	Target RPN	The Page Mapping and Control field is inherited from the TCE for the source page. If the Page Mapping and Control field is something other than page fault, then these bits contain the RPN used for the target page translation.
52:56	Unused	Reserved, all zeros.
57	Read Target	0: DMA Read operations are made to the source page. 1: DMA Read operations are made to the target page.
58	Unused	Reserved, all zeros.
59:63	Target Page Size(0:4)	This is the target page size which can be greater than or equal to the page size defined in the TVT. Note that if this size is less than the TVT page size then an error will be flagged. I/O Page Size field values supported by the PHB4: • Value of 1: use 11+1 = 12 bits (4KB I/O Page Size). • Value of 5: use 11+5 = 16 bits (64KB I/O Page Size). • Value of 10: use 11+10 = 21 bits (2MB I/O Page Size). • Value of 19: use 11+19 = 30 bits (1GB I/O Page Size). • All other values default to a 4KB I/O Page Size.
Note: Entry #0 in this table, MRT(0), is not usable or valid. This entry represents the no migrate case. Thus, only 7 of 8 entries are valid and written. This fact is reflected in the initialization sequence.		

Table 4-12. PestA Data

Bits	Fields	Definitions
0	MMIO Stopped State	MMIO Stopped State. If set, this bit indicates that MMIO operations are frozen/stopped for this PE. This bit is set in hardware when an MMIO EEH event is detected. It can also be set by firmware to force the frozen state for this PE. 1: Frozen/Stopped. 0: No error.
01:63	Unused	Reserved, all zeros.

Table 4-13. PestB Data

Bits	Fields	Definitions
0	DMA Stopped State	DMA Stopped State. If set, this bit indicates that DMA operations are frozen/stopped for this PE. This bit is set in hardware when an DMA EEH event is detected. It can also be set by firmware to force the frozen state for this PE. 1: Frozen/Stopped 0: No error
01:63	Unused	Reserved, all zeros.

Table 4-14. PestB Data Address Field(ER Error) (Page 1 of 2)

LEM Bit(s)	Error Name(s)	PESTB Address
0-2	AIB Command Invalid AIB Address Decode Error - Multi AIB Address Decode Error - None	Lower 48 bits of address from the GX bus. These are right justified in the PESTB Address field. The MSB bits are all zeros.
12	PCIe CFG/IO Write Error	PCI CFG Write: • The contents of the CONFIG_ADDRESS register (32 bits). IO Writes: • The 32 bit PCI address sent to the PCIe link. All addresses are right justified in the PESTB field. The upper bits are all zeros.
19	DMA Response Timeout	All zeros. The PHB4 does not remember the address for DMA read requests that have been issued upbound. Software must rely on the PHB4 tracing and other chip tracing functions to determine the address on a failure.
21-22	CFG Retry Timeout Error CFG Access Error	The contents of the CONFIG_ADDRESS register (32 bits). All addresses are right justified in the PESTB field. The upper bits are all zeros.
23	AIB Response Data Error	All zeros. The PHB4 does not remember the address for DMA read requests that have been issued upbound. Software must rely on the PHB4 tracing and other chip tracing functions to determine the address on a failure.
27	PAPR Outbound Injection Error Triggered	Lower 48 bits of address from the GX bus. These are right justified in the PESTB Address field. The MSB bits are all zeros.
33-36	IODA MSI PE Mismatch Error IODA MVT Error IODA TVT Error IODA TVT Address Truncation Error	PESTB(3:63) ≤ PCI Address(60:0) All 'IODA' errors are detected during the request phase of a DMA read/write or MSI interrupt request. Therefore, the lower 61 bits of the PCI Address is available for logging.

Table 4-14. PestB Data Address Field(ER Error) (Page 2 of 2)

LEM Bit(s)	Error Name(s)	PESTB Address
37	IODA Page Access Error	DMA Request phase: • 48 bit TCE address that hit in the TCE cache. • This is the error detected when a new requests hits in the cache and • its page access conflicts with the access type for the cached TCE. TCE Response from memory: • For 32-bit DMAs, the LSB 28 bits of the PCI address. • For 64-bit DMAs, the LSB 48 bits of the PCI address.
38	PAPR CFG Injection Error Triggered	PESTB(4:31) ≤ CONFIG_ADDRESS Reg(4:31). This error is detected during the request phase of a CFG read/write. Therefore, the contents of the CFG address register (0x140) is available for logging.
39	PAPR Inbound Injection Error Triggered	PESTB(3:63) ≤ PCI Address(60:0). This error is detected during the request phase of a DMA read/write or MSI interrupt request. Therefore, the lower 61 bits of the PCI Address is available for logging.
43	BLIF Inbound Completion Done Error	PCI CFG Read Response: • The contents of the CONFIG_ADDRESS register (32 bits). MMIO Load Response: • The PCI address sent to the PCIe link. All addresses are right justified in the PESTB field. The upper bits are all zeros. This address is only valid if outbound PCIe reads are sent one at a time. This is controlled by a PHB4 configuration register. See the *PHB4 - Control Register* on page 111, 0x810, bit 13.
45	TCE Request Timeout Error	All zeros. The PHB4 does not remember the address for TCE read requests that have been issued upbound. Software must rely on the PHB4 tracing and other chip tracing functions to determine the address on a failure.
58	AIB Dat_Err Indication	Lower 48 bits of address from the GX bus. This address is only valid if the error is detected on an MMIO Write command.
60-61	CFG_EC08_FATAL_ERROR CFG_EC08_NONFATAL_ERROR	All zeros. The PHB4 does not receive an address for inbound messages from the PCIe link.

Table 4-15. TVT Data (Page 1 of 3)

Bits	Field	Definition
00:47	Translation Table Address, TTA(04:51) or Address range for no translate	When the I/O Page Size field is non-zero and the TVE is valid (TCE Table Size is non-0) this field in the TTA for the given TVE Index. The TTA(04:51) bits are aligned to the same bit position as they would be in the 64 bit system address of the TCE table. That is, bit 4 is bit 4 of the 64 bit address, bit 51 is bit 51 of the system address. Note the PHB4 only supports a 56-bit system address. Therefore, the system address bits range from (08:63). TCE Tables must be aligned on a boundary which is an integer multiple of their size. Depending on the size of the table and the TCE size, some of the low-order bits are unused. Likewise, only enough bits need to be implemented to match the largest real address in the platform. Minimum alignment is at least 4KB. The TVE can be configured in two different no-translate modes, selectable via TVE[56]: • 50-bit non-translate mode • 56-bit non-translate mode In 50-bit non-translate mode, the fields of the TVE are used to perform an address range check. In this mode TCE Table Size(0) must be a '1' (TVE[51] = 1) since the TCE Table size must be non-zero for the TVE to be valid and the TCE Table size bits are overloaded to be used for the address range check. If: PCI Addr(49:24) ≥ TVE[52:53]+TVE[0:23] and PCI Addr(49:24) < TVE[54:55]+TVE[24:47], where the + sign denotes concatenation, not addition, use the PCI Address(49:0), untranslated, as the DMA system address. A 6-bit prefix is concatenated on the most significant bits of the DMA address before it is passed to the Power bus. Refer to *Section 4.4.2.19 Non-Translate Prefix Register* on page 82.
48:50	Number of TCE Table Levels	This field indicates the number of indirect TCE table levels for operations using this TVE, which is the total number of TCE table levels (including the last level) minus 1. When this field is 0, there are no indirect levels. The following values are defined by the IODA2 architecture: • 000 - Only one level (direct level). • 001 - 1 indirect level, 1 direct level. • 010 - 2 indirect levels, 1 direct level. • 011 - 3 indirect levels, 1 direct level. • 100 - 4 indirect levels, 1 direct level. • 101-111 Reserved.

Table 4-15. TVT Data (Page 2 of 3)

Bits	Field	Definition
51:55	TCE Table Size(0:4) or Valid field and additional address range bits for no-translate	A value of 0 in this field indicates that the TVE is invalid. If the I/O Page Size field in this register is nonzero (translate case) and the TVE is valid (that is, this field is nonzero), the value of this field defines the number of DMA I/O bus address bits that are used for the TCE index field or fields. The hardware uses the value of this field, along with the Number of TCE Table Levels and I/O Page Size fields of this TVE, to validate the range of the DMA I/O address (that is, validates that the appropriate number of high-order bits in the DMA I/O bus address are 0), and to prevent an IOA function from accessing outside of its address range. Value of 0: Invalid TVE. Value of 1: 9 TCE Index bits, 4 KB table size. Value of 2: 10 TCE Index bits, 8 KB table size. Value of 3: 11 TCE Index bits, 16 KB table size. Value of 4: 12 TCE Index bits, 32 KB table size. Value of 5: 13 TCE Index bits, 64 KB table size. Value of 6: 14 TCE Index bits, 128 KB table size. Value of 7: 15 TCE Index bits, 256 KB table size. Value of 8: 16 TCE Index bits, 512 KB table size. Value of 9: 17 TCE Index bits, 1 MB table size. Value of 10: 18 TCE Index bits, 2 MB table size. Value of 11: 19 TCE Index bits, 4 MB table size. Value of 12: 20 TCE Index bits, 8 MB table size. Value of 13: 21 TCE Index bits, 16 MB table size. Value of 14: 22 TCE Index bits, 32 MB table size. Value of 15: 23 TCE Index bits, 64 MB table size. Value of 16: 24 TCE Index bits, 128 MB table size. Value of 17: 25 TCE Index bits, 256 MB table size. Value of 18: 26 TCE Index bits, 512 MB table size. Value of 19: 27 TCE Index bits, 1 GB table size. Value of 20: 28 TCE Index bits, 2 GB table size. Value of 21: 29 TCE Index bits, 4 GB table size. Value of 22: 30 TCE Index bits, 8 GB table size. Value of 23: 31 TCE Index bits, 16 GB table size. Value of 24: 32 TCE Index bits, 32 GB table size. Value of 25: 33 TCE Index bits, 64 GB table size. Value of 26: 34 TCE Index bits, 128 GB table size. Value of 27: 35 TCE Index bits, 256 GB table size. Value of 28: 36 TCE Index bits, 512 GB table size. Value of 29: 37 TCE Index bits, 1 TB table size. Value of 30: 38 TCE Index bits, 2 TB table size. Value of 31: 39 TCE Index bits, 4 TB table size. When the I/O Page Size field is zero (no translate case) and the TVE is valid (that is, this field is non-zero), then this indicates that the TTA(04:51) bits along with bits from this field are to be used to validate the PCI Express address (see the definition column for the “Address range for no translate” field row of this table).
56	50-bit Non-Translate Mode Enable	Set to '1' to enable 50-bit non-translate mode when the TVE is configured for non-translate mode. The default '0' is 56-bit non-translate mode. In this mode there is no range checking and all PCI address bits 55:0 are passed through as-is to the Power bus logic.

Table 4-15. TVT Data (Page 3 of 3)

Bits	Field	Definition
57:58	unused	Reserved, all zeros.
59:63	I/O Page Size(0:4) Non-Translate Mode is enabled when this field is all zeros.	The number of low-order PCI address bits to be used as page offset is this field plus 11. A value of 0 in this field indicates that the address should not be translated. Note: The value of the TVE is static relative to DMA operations for the IOA that will use the TVE. That is, the IOA must be disabled when the TVE is changed. Therefore, usage of the TTA as a sort of pseudo-TCE is not reasonable, except in the static case. For the Translation case, I/O Page Size values supported by the PHB4: • Value of 1: use 11+1 = 12 bits (4KB I/O Page Size). • Value of 5: use 11+5 = 16 bits (64KB I/O Page Size). • Value of 10: use 11+10 = 21 bits (2MB I/O Page Size). • Value of 19: use 11+19 = 30 bits (1GB I/O Page Size). • All other values default to a 4KB I/O Page Size.

The TCR Data (*Table 4-16* on page 97) represents data that can be read/written via the IODA Table Data Register. Each TCR entry in the table represents meta data for a single set plus way index in the cache.

The TCR table address is composed as {set index(0:log2(M)-1), way(0:1)}:
- The number of ways is 4 in the PHB4 implementation. It is a 4-way set-associative cache.
- The set index size M is the total number of cache entries divided by the number of ways, 4 in this case.
- PHB4X08, M=128.
- PHB4X16, M=256.

Each TCR entry has a matching/companion TDR entry and the entry number matches one for one.

Table 4-16. TCR Data (Page 1 of 2)

Bits	Fields	Definitions
0	Valid State Bit	This is a state bit for the cache entry. This bit is set when the TCAM entry is allocated. This bit will get cleared after the entry is killed via a TCE Kill command.
1	Pending State Bit	This is a state bit for the cache entry. This bit is set when the entry is allocated indicating that a TCE fetch is pending. This bit is cleared when the TCE data is returned for the fetch.
2	Kill State Bit	This is a state bit for the cache entry. This bit is set when a TCE Kill is received for the entry. This bit will clear when the entry goes invalid (all of the state bits will be zero). This bit will remain asserted if the Pending State Bit is asserted. This allows the entry to remain valid until its TCE fetch response is received. Once the fetch data has been received, the killed entry will go invalid and all state bits will clear.
3	Unused	Reserved, all zeros.

Table 4-16. TCR Data (Page 2 of 2)

Bits	Fields	Definitions
04:51	Compare Address(04:51)	This is the PCI address stored in the cache and it is compared against the PCI address of the DMA command on a cache lookup. The bits 04:51 correspond to PCI address bits 59:12. The address is 4KB aligned (least significant 12 bits are not compared) since the minimum page size is 4KB. Note that the Page Size(0:1) can vary for each cache entry based on the configuration of the TVT entry that corresponds to the DMA that installed the cache entry. The Page Size(0:1) is stored in the cache entry.
52:53	Unused	Reserved, all zeros.
54:55	Page Size(0:1)	Page Size(0:1) encoding for this cache entry for the supported page sizes in the current design. 00 – 4KB. 01 – 64KB. 10 – 2MB. 11 – 1GB. This field is used to mask off the appropriate least significant bits of the Compare Address(04:51) so that the correct bits are compared. Also, the page size stored in the cache entry must be equal to the page size from the TVT entry for the DMA command otherwise there will not be a match/hit.
56:60	unused	Reserved, all zeros.
61:63	LRU State(0:2)	LRU state bits. This field is only valid for way#0 in all sets and are unused in the other ways.

The TDR Data (*Table 4-17* on page 99) represents data that can be read/written via the IODA Table Data Register. Each TDR entry in the table represents meta data for a single set plus way index in the cache.

The TDR table address is composed as {set index(0:log2(M)-1), way(0:1)}:

- The number of ways is 4 in the PHB4 implementation. It is a 4-way set-associative cache.
- The set index size M is the total number of cache entries divided by the number of ways, 4 in this case.
- PHB4X08, M=128.
- PHB4X16, M=256.

Each entry in the TDR stores the RPN(08:51), Migration Pointer(0:3) and Page Control(0:1) bits of its corresponding TCE from system memory.

The format of this table matches the bit positions in the actual 8-byte TCE in memory for the RPN, Migration Pointer, and Page Control fields.

Each TDR entry has a matching/companion TCR entry and the entry number matches one for one.

Note in the PHB4 implementation the Migration Pointer(0:3) increased to the IODA architecture limit of 4 bits. This was needed since the PHB4X16 will now support 16 Migration Register Table (MRT) entries. In the PHB4X08 case, there are only 8 MRT entries, but the Migration Pointer(0:3) field size is still fixed at 4 bits. In this case, the MRT entry numbers 8 to 15 are aliased with numbers 0 to 7.

Table 4-17. TDR Data

Bits	Fields	Definitions
00:07	PE Number(00:07)	Most significant PE number bits for the cache entry. The PHB4 implementation can not use all the bits depending on the number of PEs implemented in the design. Any unused bits will be don't care.
08:51	Real Page Number RPN(08:51)	TCE State Information – RPN(08:51). This field is sourced from the original TCE in system memory and is stored in the TDR. It contains the real page number bits used for address translation. The current size is limited for a 56-bit system address size.
52:55	Migration Pointer(0:3)	TCE State Information - Migration Pointer(0:3). This field is sourced from the original TCE in system memory and is stored in the TDR. Used during a migration operation, the meaning is as follows: • 0000 - A migration is not in process for this page. • Not 0000 - A migration is in progress, and the value of this field points to which MRT entry to use for this operation.
56:59	PE Number(08:11)	Least significant PE number bits for the cache entry. These bits are used in all PHB4 implementations. The PE Number(0:11) bits are combined to compare against the PE Number of the DMA command (only the rightmost, implemented bits of the 0:11 field). The number stored in the cache entry must be equal to the number for the DMA command else there will not be a match/hit.
60:61	Reserved	Reserved, all zeros.
62:63	Page Control(0:1)	TCE State Information - Page Control(0:1). These are the 'Page Control' bits as defined in the TCE definition from the even TCE in memory, stored in the entry. Bit definition: • 00 - page fault. • 01 - read-only. • 10 - write-only. • 11 - read/write.

The THASH information is used for debug purposes only. The contents represent the hardware memory for the TCE request and response, stored in an internal SRAM in the hardware. This data is accessed on the TCE response, based on the TCE tag. The contents are required to process the TCE response correctly in the hardware. The THASH Data can be read/written via the IODA Table Data Register.

Each entry number in the THASH tables (*Table 4-18* and *Table 4-19* on page 100) is the TCE tag.

Each IODA Table Address Register can access one half of the entry. It takes two accesses to get all the data for the entry. Example:

```
entry#0, part 0 = table address = 0
entry#0, part 1 = table address = 1
entry#1, part 0 = table address = 2
entry#1, part 1 = table address = 3
entry#2, part 0 = table address = 4
entry#2, part 1 = table address = 5
and so forth
```

Table 4-18. THASH Data (Part 0)

Bits	Fields	Definitions
00:02	req_type(0:2)	Internal request type field.
3	dmard	1=DMA read, 0=DMA write.
04:51	pci_addr(04:51)	Original PCI address bits 4:51.
52:54	levels(0:2)	Number of TCE translation levels.
55:63	Reserved	Reserved, all zeros.

Table 4-19. THASH Data (Part 1)

Bits	Fields	Definitions
00:11	penum(0:11)	PE number for the TCE request. Value is right justified within the 12 bits.
12:13	page_size(0:1)	TCE page size encoding for the request.
14:18	tbl_size(0:4)	TCE table size encoding for the request.
19	tclb_alloc	A cache line buffer entry has been allocated for the TCE request if set to '1'.
20:31	setidx(0:11)	TCE cache set index for the request.
32:47	seid(0:15)	SEID value from the TVE entry for the request.
48:49	way(0:1)	TCE cache way number allocated for the request.
50:51	tclb_ent(0:1)	Internal cache line buffer entry for the TCE request.
52:55	addr_idx(57:60)	Least significant address bits used to select the correct 8-byte TCE entry of the 128-byte cache line.
56:63	Reserved	Reserved, all zeros.

The MBT Data (*Table 4-20* on page 101 and *Table 4-21* on page 101) represents data that can be read/written via the IODA Table Data Register. Each entry in the MBT table is represented by the contents of the two tables above. The entry number is the BAR number.

Each IODA Table Address Register (table address) can access one half of the entry. It takes two accesses to get all the data for the entry. Example:

```
entry#0, part 0 = table address = 0
entry#0, part 1 = table address = 1
entry#1, part 0 = table address = 2
entry#1, part 1 = table address = 3
entry#2, part 0 = table address = 4
entry#2, part 1 = table address = 5
and so forth…
```

Table 4-20. MBT Data (Part 0)

Bits	Field	Definitions
0	BAR Enable Bit (shared bit with Part 1)	Enables this BAR region. This bit must be set to a '1' to obtain a valid match for this BAR. This bit always defaults to '0' after reset. NOTE: This bit is shared between Part 0/1 halves. Firmware can read/write this bit from either half of the entry, whichever is most convenient.
1	BAR Space Type Bit	Chooses the BAR type, 32 or 64-bit. • M64 = ‘0’. • M32 = ‘1’. • Default is M64 = ‘0’.
02:03	BAR Mode Bits(0:1)	The BAR Mode bits select how the MMIO address is mapped to a PE#: • 00 – BAR segment number is the PE#. • 01 – MDT table lookup (index is BAR segment number). • 10 – Single PE# (single PE for this BAR region). • 11 – Reserved for future use (behavior not defined). • Default to all zeros.
04:05	Segment Divisor(0:1) OR MDT Column Number(0:1)	These bits are overloaded and change meaning depending on the BAR Mode Bits. These bits are do not care for the BAR Modes not listed here. BAR Mode = ’00’: • These bits are treated as the ‘Segment Divisor’ select bits. This allows the BAR range to be divided by a different number of equal segments. • 00 – Maximum number of PEs supported (256/512) equal segments (default). • 01 – 128 equal segments. • 10 – 64 equal segments. • 11 – 8 equal segments. BAR Mode = ’01’: • The MDT column field choose which column (0 to 3 as numbered) of the MDT entry/segment to use as the PE# when the BAR Mode is selected for MDT table lookup mode. • Default to column #0.
06:07	Reserved	Reserved, all zeros.
08:51	Base Address(08:51)	Base Address bits for compare. Base address is 4KB aligned and supports a 56-bit Power bus address.
52:63	Reserved	Reserved, all zeros.

Table 4-21. MBT Data(Part 1) (Page 1 of 2)

Bits	Field	Definitions
0	BAR Enable Bit (shared bit with Part 0)	Enables this BAR region. This bit must be set to a '1' to obtain a valid match for this BAR. This bit always defaults to '0' after reset. NOTE: This bit is shared between Part 0/1 halves. Firmware can read/write this bit from either half of the entry, whichever is most convenient.
01:07	Reserved	Reserved, all zeros.

Table 4-21. MBT Data(Part 1) (Page 2 of 2)

Bits	Field	Definitions
08:51	Mask Bits(08:51)	This is the address mask that is associated with the Base Address bits for compare.
52:54	Reserved	Reserved, all zeros.
55:63	Segment Base(0:8-N) OR Single PE Number(0:8)	These bits are overloaded and change meaning depending on the BAR Mode Bits. These bits are don't care for the BAR Modes not listed here. BAR Mode = '00': • These bits are treated as the 'Segment Base' number bits. This allows the segment number to start at a non-zero value in this BAR mode. The least significant bits of this number are zeros/unused depending on the Segment Divisor bits. This is where 'N' is the log2 of the number of segments. • Segment Divisor(0:1): • 00 – Maximum number of PEs (the Segment Base bits are not used). • 01 – 128 equal segments, N=7. • 10 – 64 equal segments, N=6. • 11 – 8 equal segments, N=3. • Note that the Segment Base bits are not used when the BAR Mode is configured for MDT Lookup. The MDT Lookup mode always uses the maximum number of PEs as the divisor. BAR Mode = '10': • This is the single, fixed PE number to use when the BAR Mode is configured for a Single PE#. • Default is all zeros.

The MDT Data table is used only in the case when the MBT table entry is configured for 'MDT Table Lookup' for its BAR Mode. Refer to the *Table 4-20 MBT Data (Part 0)*.

Table 4-22. MDT Data

Bits	Fields	Definitions
00:15	PE#A(0:15)	PE Number (Column 0). Number is right justified within the 16 bits.
16:31	PE#B(0:15)	PE Number (Column 1). Number is right justified within the 16 bits.
32:47	PE#C(0:15)	PE Number (Column 2). Number is right justified within the 16 bits.
48:63	PE#D(0:15)	PE Number (Column 3). Number is right justified within the 16 bits.

Table 4-23. PE Error Vector (PEEV)

Bits	Fields	Definitions
00:63	Nth 64bit Vector of PEs(0:63)	Vector of PEs in the error state, first 64 bits. If N = PEEV vector entry, Vector{N x 64 to [(N+1)x 64] -1)} This is a bit vector of PEs currently in the error state. A set bit indicates that either the DMA or MMIO error state bits are set for the PESTA/B entry number equal to the bit position of the set bit. This vector is read-only.

4.4.2.26 PHB4 General Capabilities Register

This register describes the general architected capabilities and functions implemented in the PHB4. This register is read-only.

Mnemonic PHB_GEN_CAP

Address Offset 0x0250

Bit	Field Mnemonic	Type	Reset Value	Description
0	PHB Error Detections Extensions Supported	RO	1	Processor Load/Store Status Register and DMA Channel Status Register are used.
1	Error Recovery Extensions Supported	RO	1	PHB4 function will log PCI errors in the Processor Load/Store Status and DMA Channel Status Registers. Load/Store Error Recovery Procedures will be supported.
2	TCE Support	RO	1	TCEs are supported for translating received PCIe memory request addresses to DMA system addresses.
3	PCI Lock Protocol Support	RO	0	The PHB4 does not support the PCI lock protocol.
4	Execution from PCI Memory Support	RO	0	The PHB4 does not support execution from PCI memory space.
5	64 bit PCI Addressing Support	RO	1	The PHB4 supports full 64 bit PCI addressing.
6	Extended 32-bit BAR Support	RO	0	The PHB4 does not provide an additional 32 bit BAR register (M32B) for MMIO load/stores.
7	Split Root Port Capable	RO	0	The PHB4 does not support the split Root Port capability as described in the IODA specification. The PHB4 acts as a single, atomic PCI Express Root Port.
8	Enhanced TCE Support	RO	1	The PHB4 supports enhanced TCE functions.
9	Hardware TCE Coherency Support	RO	0	The PHB4 does not support receiving DKill (line) commands from the AIB bus.
10	Firmware Managed TCE Coherency (FMTC) Support	RO	1	The PHB4 supports firmware managed TCE coherency.
11	Enhanced Interrupt Support	RO	1	The PHB4 supports enhanced interrupt functions.
12	Extended Error Handling (EEH) Capable	RO	1	The PHB4 supports the EEH error handling extensions.
13:63	Reserved	RO	0	Reserved.

4.4.2.27 PHB4 TCE Capabilities Register

This register describes the TCE capabilities and functions implemented in the PHB4. This register is read-only.

Mnemonic PHB_TCE_CAP

Address Offset 0x0258

Bit	Field Mnemonic	Type	Reset Value	Description
00:11	TVT Table Size	RO	(vA4.1): 0x200 (x16) 0x100 (x8) (vA4.2): 0x400 (x16) 0x200 (x8)	Number of entries in TCE Validation Table (TVT) should match the number of PEs supported. PHB4X08 - 256 entries (vA4.1). PHB4X16 - 512 entries (vA4.1). For vA4.2, the number of entries was doubled to allow for 2 PEs per PE for the maximum PE configuration for the specific PHB4 size. PHB4X08 - 512 entries (vA4.2). PHB4X16 - 1024 entries (vA4.2).
12	IOPS (LMB) Option Support	RO	1	The PHB4 support large, LMB, TCE page sizes.
13:16	Reserved	RO	0	Reserved.
17:19	Number of TCE Page Sizes Supported	RO	0x4	The PHB4 supports 4 different TCE page sizes: • 4K • 64K • 2MB (new/changed for PHB4) • 1GB (new/changed for PHB4)
20	Indirect/Two-level TCE Support	RO	0	The PHB4 supports indirect/two-level TCEs. This function aids in TCE page migration and relocation. (This field is old and does not apply to PHB4.)
21	First-level TCEs Cached	RO	1	The first-level TCE fetched from system memory is stored in the cache. (This field is old and does not apply to PHB4.)
22	Second-level TCEs Cached	RO	0	The PHB4 does not cache second-level TCEs during for its indirect TCE functionality. The second-level TCEs are used once by the instigating request and must be fetched from memory if needed in the future. The first-level TCEs are always cached. (This field is old and does not apply to PHB4.)
23:51	Reserved	RO	0	Reserved.
52:63	Number of TCEs Cached	RO	0x400 (x16) 0x200 (x8)	Number of individual TCE translation cached in the PHB4. PHB4X08 – 512 TCEs. PHB4X16 – 1024 TCEs.

4.4.2.28 PHB4 Interrupt Capabilities Register

This register describes the Interrupt capabilities and functions implemented in the PHB4. This register is read-only.

Mnemonic PHB_INT_CAP

Address Offset 0x0260

Bit	Field Mnemonic	Type	Reset Value	Description
00:03	Reserved	RO	0	Reserved.
04:07	Number of LSI Interrupts	RO	0x8	The PHB4 supports up to 8 LSI interrupt sources.
08:47	Reserved	RO	0	Reserved.
48:63	Number of MSI interrupts	RO	0x1000 (x16) 0x800 (x8)	Number of MSI interrupts supported. PHB4X08 – 2048 MSI interrupts. PHB4X16 – 4096 MSI interrupts.

4.4.2.29 PHB4 EEH Capabilities Register

This register describes the Extended Error Handling (EEH) capabilities and functions implemented in the PHB4. This register is read-only.

Mnemonic PHB_EEH_CAP

Address Offset 0x02B0

Bit	Field Mnemonic	Type	Reset Value	Description
00:11	Number of PEs	RO	0x200 (x16) 0x100 (x8)	Number of Partitionable Endpoints (PEs). These endpoints define the EEH domain granularity. PHB4X08 - 256 PEs. PHB4X16 - 512 PEs.
12:63	Reserved	RO	0	Reserved.

4.4.2.30 PAPR Error Injection Control Register

This register controls the special error injection capability described by the Power Architecture Platform Requirements (PAPR) architecture specification. The Control, Address, and Address Mask registers encompass the required set of registers to support this capability.

When an inbound or outbound Injection Enable is set then the next passing packet in that direction is subject to potential error injection. To determine if an injection will occur, the address in that packet is logically ANDed with the PAPR Error Injection Address Mask Register bit by bit (the packet address is right justified in the 64 bit field). Then that 64 bit result is compared with the PAPR Error Injection Address Register. If there is a match then that packet is marked for error injection.

The PAPR Error Injection Address and Address Mask Registers use the same matching logic structure as the MMIO Base/Mask registers. The only difference is that the PAPR registers compare on the full, raw 64-bit addresses.

When an outbound packet is marked for injection, then the packet is passed to the PBL core with the inject enable signal asserted. This tells the PBL core to mark that packet for injection. This packet will eventually be transmitted by the PCIe cores with a bad ECRC. A bit in the ECRC is flipped to cause the error. The ECRC error is always attached to that packet even if the packet is replayed by the PCIe cores.

All outbound injections, MMIO and CFG, will generate a bad ECRC.

For inbound packet injection, only DMA reads or DMA writes are affected. For DMAs, they are treated as if they are a normal rejected error case. That is to say, DMA Writes are dropped and DMA Reads return a UR or CA response.

The read and write command bits are used to specify what command type to inject the error. If both bits are set or cleared, then the inject will occur on the address for the command type that occurs first.

Mnemonic PAPR_EICR

Address Offset 0x02B0

Bit	Field Mnemonic	Type	Reset Value	Description
0	Inbound Injection Enable	RW	0	Write to a ‘1’ to enable inbound error injection. This will enable errors to be injected by comparing the raw PCI address sent by the PBL core over the BLIF interface.
1	Outbound Injection Enable	RW	0	Write a ‘1’ to enable outbound error injection. This will enable errors to be injected by comparing the raw AIB interface address, which is the original physical address sent by the processor.

Bit	Field Mnemonic	Type	Reset Value	Description
2	Sticky Injection Bit	RW	0	Write to a '1' to enable sticky error injection. By default, errors are injected in a one-shot manner. That is to say, once a single error has been injected a second error will not be injected unless this register is updated to do so again. Enabling this bit will continuously inject errors as long as either the inbound or outbound injection enable bits are asserted and there is a valid address match. The default one-shot behavior will inject the error specified by the injection enable bit and then clear both injection enable bits in this register once the error has been injected. Note: The sticky injection applies to bit 0, 1, and 3 of this register. That is, also to the Outbound CFG Injections.
3	Outbound CFG Injection Enable	RW	0	Write a '1' to enable outbound CFG error injection. This will enable errors to be injected by comparing the raw AIB interface address, which is the original physical address sent by the processor. Default value is 0.
4	Read Command bit	RW	0	Inject error on a Read Command to the specified address. Note: This should also affect CFG transactions in addition to MMIOs.
5	Write Command bit	RW	0	Inject error on a Write Command to the specified address. NOTE: This should also affect CFG transactions in addition to MMIOs.
6	EEH Freeze Disable	RW	0	Disable freezing the endpoint when the injection is triggered. This will take precedence over the setting of the EEH Freeze Enable register for the error bit. Note: The cycle that a PAPR injection error is triggered and EEH Freeze Disable is on, other errors that might freeze and endpoint will not freeze.
7	Reserved	RW	0	Reserved, but implemented register(s) for future use.
08:15	Reserved	RO	0	Reserved.
16:31	PCIE RID(0:15)	RW	0	PCIE RequesterID value to use for comparison on received request TLP's value. If set to a non-zero value, the value will be used to compare for a match against received request TLPs RequesterID (inbound only). The RID value must match for the injection to occur.
32:63	Reserved	RO	0	Reserved.

4.4.2.31 PAPR Error Injection Address Register

This register is used in conjunction with the other PAPR Error Injection Registers.

For outbound error injections, register bits [8:63] are used to match the Incoming AIB packet address field. For inbound error injections, the entire 64 bit register is used. A 32-bit TLP address is right justified in the 64 bit field.

For CFG error injections, register bits [0:31] are used to match the CFG address value written to CFG address register at 0x140. CFG address bits [0:3] and [30:31] are internally marked as “Don't Care”.

Mnemonic PAPR_EIAR

Address Offset 0x02B8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Injection Compare Address(00:63)	RW	0	The 64-bit address used in the comparison for both inbound and outbound injection.

4.4.2.32 PAPR Error Injection Address Mask Register

This register is used in conjunction with the other PAPR Error Injection Registers.

For outbound error injections, register bits [8:63] are used as the mask field. For inbound error injections, the entire 64 bit register is used.

For CFG error injections, register bits [0:31] are used as the mask field.

Mnemonic PAPR_EIAMR

Address Offset 0x02C0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Injection Address Mask(00:63)	RW	0	The 64-bit address mask used in for both inbound and outbound injection. A ‘0’ in the bit position essentially ignores that bit of the address.

4.4.2.33 ETU Error Summary Status Register

This is a read-only register that provides a single place to query if there are any error bits set in the traps or LEM FIR in the ETU block.

Mnemonic ETU_ESSR

Address Offset 0x02C8

Bit	Field Mnemonic	Type	Reset Value	Description
0	LEM FIR Bit Set	ROH	0	A bit is set in the LEM FIR Accumulator Register. Refer to *Section 4.4.4.1 LEM FIR Accumulator Register* on page 120.
1	PHB Error Status Bit Set	ROH	0	A bit is set in the PHB4 Error Status Register (trap). Refer to *Section 4.4.4.10 PHB4 Error Status Register* on page 130.
2	TXE Error Status Bit Set	ROH	0	A bit is set in the TXE Error Status Register (trap). Refer to *Section 4.4.4.21 TXE Error Status Register* on page 136.
3	RXE_ARB Error Status Bit Set	ROH	0	A bit is set in the RXE_ARB Error Status Register (trap). Refer to *Section 4.4.4.32 RXE_ARB Error Status Register* on page 145.
4	RXE_MRG Error Status Bit Set	ROH	0	A bit is set in the RXE_MRG Error Status Register (trap). Refer to *Section 4.4.4.43 RXE_MRG Error Status Register* on page 153.
5	RXE_TCE Error Status Bit Set	ROH	0	A bit is set in the RXE_TCE Error Status Register (trap). Refer to *Section 4.4.4.54 RXE_TCE Error Status Register* on page 159.
06:63	Reserved	RO	0	Reserved.

4.4.2.34 Interrupt Notify Base Address Register

This register specifies the CI Store address base value to use for interrupt notification. Both LSI and MSI interrupts use this base address to send notify CI Stores.

The number of LSI interrupt sources is hard-coded to 8 for all implementations of the PHB4 (3 bit source number).

Note: The CI Store data size for interrupt notifications is always 8-byte and the data is based on the contents of the Notify Index register and the Interrupt Source ID. For details refer to *Interrupt Notify Base Index Register* on page 110.

Mnemonic INT_NOTIFY_ADDR

Address Offset 0x0300

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	Reserved	RO	0	Reserved, all zeros.
08:60	Notify Base Address(08:60)	RW	0	Notify Base Address(08:60). This is the base address bits to send with the CI Store, interrupt notify command.
61:63	Reserved	RO	0	Reserved, all zeros.

4.4.2.35 Interrupt Notify Base Index Register

This register specifies the CI Store data value to use for interrupt notification. The values in this register are combined with the Interrupt Source ID to form the 8-byte data sent with the Interrupt Notify CI Store. Refer to *Table 4-24* on page 110.

Mnemonic INT_NOTIFY_INDEX

Address Offset 0x0308

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	Reserved	RO	0	Reserved, all zeros.
32:55	Notify Base Index(32:55)	RW	0	Notify Base Index(32:55). IVE Block+Index value, as defined by the system.
56:63	Reserved	RO	0	Reserved, all zeros.

Table 4-24. Interrupt Notify Data Table (CI Store, 8-byte data)

PHB4 Type	Notify Data
PHBX08	{0x40000000, Notify Base Index(32:52), Source ID(1:11)}
PHBX16	{0x40000000, Notify Base Index(32:51), Source ID(0:11)}

4.4.3 Fundamental Register Set B

4.4.3.1 PHB4 - Version Register

This register defines the version information specific to the current PHB4 implementation. This register is read-only.

PHB4 Major Revision/Minor Revision is referred to as v[Major #].[Minor #]. Example: vA2.1.
PHB4 vA4.1 – PHB4 used in Nimbus DD1.0.
PHB4 vA4.2 – PHB4 used in Nimbus DD2.0.

Mnemonic PHB_VR

Address Offset 0x0800

Bit	Field Mnemonic	Type	Reset Value	Description
00:23	Reserved	RO	0	Reserved.
24:31	Major Revision ID	RO	0xA4	This is the major revision number of the PHB4. It identifies the architectural level which the PHB4 is based. This number would only be changed if the PHB4 had been changed to a new architectural level.
32:55	Reserved	RO	0	Reserved.
56:63	Minor Revision ID	RO	0x02	This is the minor revision number of the PHB4. It is used to identify small changes in the PHB4 design. This is the revision number of the PHB4 implementation within a chip. This number can or can not increment with successive chip releases. If a new chip is released and the PHB4 logic remained unchanged this revision number would not be changed.

4.4.3.2 PHB4 - Control Register

This register contains general status information and control bits for the PHB4.

Mnemonic PHB_CTRLR

Address Offset 0x0810

Bit	Field Mnemonic	Type	Reset Value	Description
00:10	Reserved	RW	0	Reserved, but implemented register(s) for future use.
11	Interrupt Page Size Bit	RW	0	Page Size select bit for interrupt state processing. This affects the interrupt source ID offset for incoming CI Loads and Stores used for Interrupt EOI events. This bit is global for all interrupts, LSI and MSI. '0' = 4KB (default), the Source ID is bits [40:51] of the AIB address. '1' = 64KB, the Source ID is bits [36:47] of the AIB address.

Bit	Field Mnemonic	Type	Reset Value	Description
12	Interrupt stEOI Command Enable (vA4.2) Reserved (vA4.1)	RW	0	Interrupt CI Store w/EOI Command Enable. The control bit will switch the behavior of how the CI Store to trigger an interrupt will be handled, in the PHB4. If this bit is enabled, it will disable the original behavior and ability to trigger an interrupt if the state of the interrupt is PQ=00. Instead CI Stores to the interrupt space will do the following when this bit is enabled: • Adds the functionality of a CI Load/EOI command. • Clears P bit when PQ=10 state (no queued second interrupt). • Sends a new notification for an interrupt if PQ=11 state (queued second interrupt). Disabled = 0 (default): • CI Store Trigger command original behavior. Enabled = 1 (enable new CI Store w/EOI behavior): • Replaces original CI Store Trigger command behavior.
13	MMIO Read Strict Ordering Enable	RW	0	This bit enables a function in downbound flow where very strong ordering of Read requests and their responses is enforced. PHB4 always issues MMIO Reads in the order received, queueing up to 8. However, it cannot guarantee the order in which the responses to these requests are received from the PCIe link. When set, this bit will cause the PHB4 to issue the next MMIO Read request received for a particular PE#, then wait for the response to that same request to be received before issuing another MMIO Read for that same PE#. If a MMIO Read request is received for different PE#, it will be issued. In effect, it means that only one read request can be outstanding for each endpoint. In this way, ordering of the requests and their responses is maintained on a PE basis. When cleared, the PHB4 will issue MMIO Read requests in the order received without regard for PE#. This bit has no effect on MMIO Writes, or configuration R/Ws. This function is in response to a PAPR requirement to enforce strict ordering of reads, to prevent a read from passing another read on the link, where the PCIe specification can allow such a condition to occur. 0=Normal operation. 1=Enable Strict Read ordering. Default value is 0.
14	MMIO EEH Disable	RW	0	Set to '1' to disable EEH for all MMIO commands. The EEH error state will be ignored and no EEH related errors will be flagged. This includes PCI CFG commands. • Disables PE# checking for MMIO requests. • Disables EEH freeze for MMIO induced errors. • Disables any and all side effects of EEH for MMIO commands. This disable bit could be a workaround for certain Linux applications.
15	PCI CFG response EEH, PE Stopped State Check Enable	RW	0	Switch to enable checking the PE stopped state for PCI CFG responses. This only affects the checking of the PE stopped state for PCI CFG operations. 0 = do not check PE stopped state. 1 = check PE stopped state.
16	PCIe Link Kill on AIB Fence Disable	RW	0	Switch to disable killing the PCIe link if AIB is fenced.

Bit	Field Mnemonic	Type	Reset Value	Description
17:19	TVT Address Select(0:2)	RW	0	Choose PCI address indexing method for TVT index (IODA2). One or more PCIe address bits are appended to the PE# to form the TVT look up address. The TVT address is formed according to the following decode (The PCI address is shown in little-endian format:: A[63:0]). This is the table for (vA4.1): 000: {PE#[0:8]}, selects 1 TVE per PE; 512 valid PE#s. (this(001) is the recommended firmware setting for vA4.1): 001: {PE#[1:8], A[59]}, selects 2 TVEs per PE; 256 valid PE#s. 010: {PE#[2:8], A[59:58]}, selects 4 TVEs per PE; 128 valid PE#s. 011: {PE#[3:8], A[59:57]}, selects 8 TVEs per PE; 64 valid PE#s. 100: {PE#[4:8], A[59:56]}, selects 16 TVEs per PE; 32 valid PE#s. 101: Reserved; will default to decode 000, 1 TVE per PE. 110: Reserved; will default to decode 000, 1 TVE per PE. 111: Reserved; will default to decode 000, 1 TVE per PE. This is the table for (vA4.2): 000: {PE#[0:8], A[59]}, selects 2 TVEe per PE; 512 valid PE#s. 001: {PE#[1:8], A[59:58]}, selects 4 TVEs per PE; 256 valid PE#s. 010: {PE#[2:8], A[59:57]}, selects 8 TVEs per PE; 128 valid PE#s. 011: {PE#[3:8], A[59:56]}, selects 16 TVEs per PE; 64 valid PE#s. 100: Reserved; will default to decode 000, 2 TVEs per PE. 101: Reserved; will default to decode 000, 2 TVEs per PE. 110: Reserved; will default to decode 000, 2 TVEs per PE. 111: Reserved; will default to decode 000, 2 TVEs per PE.
20	po_tce_set_hash_sel	RW	0	TCE Cache Set Index Hash Selection. This bit controls the set hashing function of the TCE cache. If set to '1' will effectively divide the cache in half. Half will be dedicated to DMA reads and half to DMA writes. When '0' all sets can be allocated to any operation type DMA read or write.
21	po_tce_clb_disable	RW	0	TCE Cache Line Buffer Disable. Set this to '1' to disable the internal cache line buffer for TCE fetches. This should be enabled '0' unless there is a logic bug as it can drastically reduce performance of the IO if disabled.
22	po_lsi_int_disable	RW	0	LSI Interrupt Disable. Set this to '1' to mask the LSI interrupts from the PCIe link. This will not affect error interrupts from the PHB4. It only affects the PCIe, LSI interrupt levels A,B,C,D. This will effectively ignore LSI interrupts from the PCIe link.
23	po_msi_int_disable	RW	0	MSI Interrupt Disable. Set this to '1' to disable MSI interrupts from the PCIe link. If set, when any valid MSI interrupt is received it will effectively be dropped. No interrupt notifications will take place and the interrupt state will transition from RESET to OFF state. This bit does not affect interrupts that have already been issued. It only affects 'new' interrupts that are in the RESET state.

Bit	Field Mnemonic	Type	Reset Value	Description
24	po_tvt_sel_gte4gb (vA4.2)	RW	0	TVT Select 'GTE4GB' Option. This is a control bit for a sub-option of the TVT Address Select(0:2) field. If '0', there is no change to the TVT Address Select(0:2) field. If '1', the field changes to the following: 000: {PE#[0:8], GTE4GB}, selects 2 TVEe per PE; 512 valid PE#s. This only affects the '000' TVT Select encoding. The least significant bit was PCI Address bit 59. It is now changed to use the new option bit. The 'GTE4GB' bit is a '1' when the PCI Address is greater than or equal to 4GB (not 32-bit). This allows a single PE to span 2 TVE entries if its address space spans across the 4GB line. That is, PCI addresses from the PE that are below 4GB will use TVT entry offset 0, greater than or equal to 4GB will use TVT entry offset 1.
25:27	Reserved	RW	0	Reserved, but implemented register(s) for future use.
28:31	DMA Read Request Spacing Count(0:3)	RW	0	This value is used to insert a fixed number of cycles between DMA read requests issued over AIB. The default value of zero means DMA reads will issue as soon as they can. This value relates to a performance DCR.
32:47	Reserved	RW	0	Reserved, but implemented register(s) for future use.
48:63	Reserved	RO	0	Reserved.

4.4.3.3 PHB4 - AIB Fence Control Register

This register controls the AIB fence related registers for the AIB TX port of the PHB4.

Note: That software is not required to modify this register. This register exists for debug or work around purposes only.

Mnemonic PHB_AIB_FCR

Address Offset 0x0860

Bit	Field Mnemonic	Type	Reset Value	Description
0	Raise Fence	RWH	0	This field directly addresses the register bit that drives the *ai_tx_raise_fence* signal. Software can set or clear this register for debug purposes if desired.
1	Fence Active	RWH	0	This field directly addresses the internal register bit that will prevent the AIB TX port from sending new requests. If set, this bit forces the AIB TX logic to act as if it did not have command or data credits. Software can set or clear this register for debug purposes if desired.
02:63	Reserved	RO	0	Reserved

4.4.3.4 PHB4 - TCE Tag Enable Register

This register enables or disables internal TCE tags. New TLP headers, request, and responses must obtain a TCE Tag to be processed.

Mnemonic PHB_TCE_TER

Address Offset 0x0868

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	TCE Tag Enable Vector(00:31)	RW	0xFFFFFFFF	Each bit of this vector enables an individual TCE Tag, 0 to 31. Enable=1, Disable=0.
32:63	TCE Tag Enable Vector(32:63) (x16) Reserved (x8)	RW (x16) RO (x8)	0xFFFFFFFF (x16) 0 (x8)	Each bit of this vector enables an individual TCE Tag, 32 to 63. Enable=1, Disable=0. Tags 32 to 63 are only valid for PHBX16. Tags 32 to 63 are not implemented for PHBX08.

4.4.3.5 PHB4 - TCE Tag Watermark Register

This register controls the allocation of TCE tags by the various requests and responses. The watermarks should be set appropriately so that sufficient tags remain to service all types.

The number of TCE tags left must be greater than the watermark value to allow the command to allocate a new TCE Tag.

Note: The watermark must be set to one more than is desired for a command to allocate since the watermark test result is registered. For instance, if you want to guarantee that a command will not allocate the last 2 tags that are left, set the watermark setting to a 3. The watermark value is a whole number so, 0 means 0.

Mnemonic PHB_TCE_TWR

Address Offset 0x0870

Bit	Field Mnemonic	Type	Reset Value	Description
0	Reserved	RW	0	Reserved, but implemented register(s) for future use.
01:07	Posted Low Watermark	RW	0	Low watermark value. Posted requests should be allowed to consume most all TCE tags since Completions and nonposted requests cannot pass them anyway.
8	Reserved	RW	0	Reserved, but implemented register(s) for future use.
09:15	Nonposted Low Watermark	RW	0x03	Low watermark value. Nonposted requests should not be allowed to consume all TCE tags since they must allow Completions and Posted Requests to get through.
16	Reserved	RW	0	Reserved, but implemented register(s) for future use.
17:23	Completion Low Watermark	RW	0	Low watermark value. Completions behave similar to nonposted requests.
24:32	Reserved	RW	0	Reserved, but implemented register(s) for future use.
33:39	RID Speculation Request Low Watermark	RW	0x03	Low watermark value. RID speculation requests for the RTC cache.
40	Reserved	RW	0	Reserved, but implemented register(s) for future use.
41:47	Speculation Fetches for DMA Reads Low Watermark	RW	0x03	Low watermark value. Speculative requests for DMA reads. See *Speculation Control Register* on page 86.
48	Reserved	RW	0	Reserved, but implemented register(s) for future use.
49:55	Speculation Fetches for DMA Writes Low Watermark	RW	0x03	Low watermark value. Speculative requests for DMA writes. See *Speculation Control Register* on page 86.
56:58	Reserved	RW	0	Reserved, but implemented register(s) for future use.
59:63	Nonposted Limit(0:5)	RW	0	Nonposted Limit value. Defines how many total nonposted requests are allowed to be pending in the PHB4 pipeline. Used for traffic tuning/balancing with Posted requests. A value of zeros indicates no limit; requests will be serviced as long as there are available TCE tags, and the nonposted low watermark has not been reached.

4.4.3.6 PHB4 - Timeout Control Register 1

This register controls timeout counters in the PHB4 Internal Logic.

The configuration value of the timeout will always represent a minimum/maximum timeout range.

- Minimum timeout value is: $2^{(\text{timeout value})}$ x 16ns.
- Maximum timeout value is: $2^{(\text{timeout value})}$ x 24ns.

Mnemonic PHB_TCR_1

Address Offset 0x0878

Bit	Field Mnemonic	Type	Reset Value	Description
00:09	Reserved	RW	0	Reserved, but implemented register(s) for future use.
10:15	BLIF Forward Progress Timeout Timer Value	RW	0x16	BLIF interface Forward Progress Timer Value. A timer is started when there is a new command waiting to be issued over the BLIF interface and there are not sufficient credits to issue the command. If the BLIF cannot accept another command before the timer expires, it is an error. There are 2 timers, one for Read and one for Write commands. Setting bit 10 to '1' disables the timers. Bits 11:15 define the timeout value. Default value is 0x16. This provides an initial timeout range of ~67-100ms.
16:17	Reserved	RW	0	Reserved, but implemented register(s) for future use.
18:23	MMIO Request Timeout Timer Value	RW	0x16	MMIO Request Timeout Value. Each time a Outbound MMIO Request command is issued to the PCIe logic over BLIF, a timer is started. If the Response data does not return before the timer expires, it is an error. There are 8 timers, one for each possible MMIO Request issued. **Note:** For the PHB4 design only MMIO Read Requests are timed and expect a response. MMIO Writes are not timed and never receive a response. Setting bit 18 to '1' disables the timers. Bits 19:23 define the timeout value. Default value is 0x16. This provides an initial timeout range of ~67-100ms.
24:39	Reserved	RO	0	Reserved.
40:41	Reserved	RW	0	Reserved, but implemented register(s) for future use.
42:47	CFG Request Timeout Timer Value	RW	0x20	Each time a CFG Request (read or write) is issued to the PCIe logic over BLIF, a timer is started. If a response does not return before the timer expires, it is an error. Setting bit 42 to '1' disables the timers. Bits 43:47 define the timeout value. Default value is 0x20. This timer is disabled by default.
48:63	Reserved	RO	0	Reserved.

4.4.3.7 PHB4 - Timeout Control Register 2

This register controls timeout counters in the PHB4 Internal Logic. The configuration value of the timeout will always represent a minimum/maximum timeout range.

- Minimum timeout value is: $2^{(\text{timeout value})}$ x 16ns.
- Maximum timeout value is: $2^{(\text{timeout value})}$ x 24ns.

Mnemonic PHB_TCR_2

Address Offset 0x0880

Bit	Field Mnemonic	Type	Reset Value	Description
00:17	Reserved	RW	0	Reserved, but implemented register(s) for future use.
18:23	TCE Req Timeout Timer Value	RW	0x16	Each time a TCE Read Request command is issued to the host, a timer is started. If the Response data does not return before the timer expires, it is an error. Setting bit 18 to '1' disables the timers. Bits 19:23 define the timeout value. Default value is 0x16. This provides an initial timeout range of ~67-100ms.
24:25	Reserved	RW	0	Reserved, but implemented register(s) for future use.
26:31	AIB Transmit Timeout Timer Value	RW	0x16	This timer measures inactivity on the AIB Transmit interface. When there is a command pending, and it cannot be issued, a timer is started. The timer runs until AIB credits allow the pending command to be issued. If the timer expires, it is an error. Setting bit 26 to '1' disables the timer. Bits 27:31 define the timeout value. Default value is 0x16. This provides an initial timeout range of ~67-100ms.
32:63	Reserved	RO	0	Reserved.

4.4.3.8 PHB4 - Quiesce DMA Register

This register is used to temporarily prevent the internal logic from processing new DMA read/write/MSI interrupts from the BLIF inbound interface.

Mnemonic PHB_Q_DMA_R

Address Offset 0x0888

Bit	Field Mnemonic	Type	Reset Value	Description
0	Quiesce DMA	RWH	0	Value of 0: Allows all commands from the BLIF inbound interface to be processed normally. Value of 1: Prevents the internal logic from processing new DMA read, write, or MSI interrupts from the BLIF inbound interface. Software can use this register to temporarily halt new commands from being processed at the BLIF inbound interface. This effectively halts all new PCIe commands from being processed. Note: If asserted too long, adapter timeouts can result.
1	Enable Quiesce DMA Auto-Reset HW Function	RWH	0	Setting this bit, along with bit 0, will enable the PHB4 hardware to automatically reset both bits [0:1] to release the quiesce condition without requiring a software write to clear bit 0. The auto-reset condition is as follows: When bits [0:1] of this register = 0b11, the PHB4 logic will sample the status bits {4,6,7} (not the MMIO status bit 5). When all sampled status bits are zeros, the logic will clear bits [0:1] of this register. Then the quiesce condition will be removed allowing TCE/DMA request traffic to flow again. Default value is zero.
02:03	Reserved	RO	0	Reserved.
4	DMA Response Status pb_etu_stat_dmard_rsp_pend	ROH	0	This is a status bit that indicates if the PHB4 expects to receive one or more DMA Responses from memory; either a DMA Read Data response or a Write Done response. For PHB4, there are no Write Done responses.
5	MMIO Response Status	ROH	0	This is a status bit that indicates if the PHB4 expects to receive one or more MMIO Responses from the PCIe link.
6	TCE Response Status	ROH	0	This is a status bit that indicates if the PHB4 expects to receive one or more TCE Responses from memory.
7	TCE Kill Status	ROH	0	This is a status bit that indicates the PHB4 is currently processing one or more TCE Kill requests against the TCE cache. A TCE Kill can take multiple cycles to affect the TCE cache state.
08:63	Reserved	RO	0	Reserved.

4.4.3.9 PHB4 - TCE Tag Status Register

This register shows the current tag status for TCE tags.

Mnemonic PHB_TCE_TSR

Address Offset 0x0908

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	TCE Tag Status Vector(00:31)	ROH	0	Each bit of this vector indicates the status for an individual tag, 0 to 31. Not used=0, Used=1.
32:63	TCE Tag Status Vector(32:63) (x16) Reserved (x8)	ROH (x16) RO (x8)	0	Each bit of this vector indicates the status for an individual tag, 32 to 63. Not used=0, Used=1. Tags 32 to 63 are only valid for PHBX16. Tags 32 to 63 are not implemented for PHBX08.
16:63	Reserved	RO	0	Reserved.

4.4.4 Error Registers

This section covers the following error registers:

- LEM FIR Accumulator Register, page 120.
- PHB4 Error Status Register, page 130.
- TXE Error Status Register, page 136.
- RXE_ARB Error Status Register, page 145.
- RXE_MRG Error Status Register, page 153.
- RXE_TCE Error Status Register, page 159.

4.4.4.1 LEM FIR Accumulator Register

This is the main Fault Isolation Register (FIR). It accumulates all the current errors that are reported by the PHB4. This is a standard STG pervasive compliant register.

For this register, multiple error bits might be asserted at the same time. Software can write specific error bits to ‘1’ or can clear existing errors by writing a specific bit to a ‘0’. Partial reads or writes to this register are not supported. Software can use the LEM FIR And/Or Mask Registers to clear and set bits, thus avoiding a read-modify-write operation.

Mnemonic LEM_FIR_AR

Address Offset 0x0C00

Bit	Field Mnemonic	Type	Reset Value	Description	'OR' of Error Bits	Error Class
0	TXE: AIB Command Invalid	RWHS	0	See TXE Error Status Register page 136, bit 0 for details.	TXE: 0	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	'OR' of Error Bits	Error Class
1	TXE: AIB Addressing Error	RWHS	0	See TXE Error Status Register page 136, bit 2 for details.	TXE: 2	INF
2	TXE:AIB Size/Alignment Error	RWHS	0	See TXE Error Status Register page 136, bit 3/8 for details.	TXE: 3,8	Fatal
3	TXE: PAPR Outbound Injection Error Triggered	RWHS	0	The condition set up by the PAPR Error Injection Registers triggered its intended condition to inject an outbound error. This is more of a status than an error, but it can be treated like a normal error.	TXE: 28,30	ER (SINGLE)
4	TXE: AIB Fatal Class Errors	RWHS	0	See TXE Error Status Register (page 136) bits for details.	TXE: 4,5,9,10,11, 14,15	Fatal
5	TXE: AIB INF Class Errors	RWHS	0	See TXE Error Status Register (page 136) bits for details.	TXE: 12,36,42,46	INF
6	Reserved	RWHS	0	Reserved.		
7	TXE: AIB DAT_ERR Indication	RWHS	0	See TXE Error Status Register page 136, bit 13, 22 for details.	TXE: 13,22	Fatal
8	TXE: AIB Common Array Based Fatal Errors	RWHS	0	See TXE Error Status Register (page 136) bits for details.	TXE: 23,37,38,40 ,43,44, 45,47,48, 49	Fatal
9	TXE: AIB Common Bus/Register Based Fatal Errors	RWHS	0	See TXE Error Status Register page 136, bit 50, 51, 52 for details.	TXE: 50,51,52	Fatal
10	TXE: AIB Common Logic Based Fatal Errors	RWHS	0	See TXE Error Status Register page 136, bit 19, 20, 21, 53, 54, 55 for details.	TXE: 19,20,21, 53,54,55	Fatal
11	TXE: BLIF Controls Parity Error	RWHS	0	See TXE Error Status Register page 136, bit 16 for details.	TXE: 16	Fatal
12	TXE: PCIe CFG Write CA or UR Response	RWHS	0	See TXE Error Status Register page 136, bit 17 for details.	TXE: 17	ER (SINGLE)
13	TXE: BLIF Forward Progress Timeout	RWHS	0	See TXE Error Status Register page 136, bit 18 for details.	TXE: 18	Fatal
14	TXE: RRB Sourced Errors	RWHS	0	See TXE Error Status Register (page 136) bits for details.	TXE: 56,57,58,59	Fatal
15	TXE: CFG Request Related Errors	RWHS	0	See TXE Error Status Register page 136, bit 24, 25, 26, 27, 29 for details.	TXE: 24,25,26, 27,29	INF
16	RSB: PHB Register Bad Address Error	RWHS	0	A general register access error – access to illegal address.	PHB: 00	INF
17	RSB: FDA Fatal Class Errors	RWHS	0	Fundamental 'A' Registers, address range 1003F8.	PHB: 2,3, 5	Fatal
18	RSB: FDA INF Class Errors	RWHS	0	Fundamental 'A' Registers, address range 1003F8.	PHB: 1,4	INF
19	RSB: FDB Fatal Class Errors	RWHS	0	Fundamental 'B' Registers, address range 800BF8.	PHB: 9,10	Fatal
20	RSB: FDB INF Class Errors	RWHS	0	Fundamental 'B' Registers, address range 800BF8.	PHB: 8	INF
21	RSB: ERR Fatal Class Errors	RWHS	0	RSB Local Error Registers, address range C00CF8.	PHB: 7	Fatal
22	RSB: ERR INF Class Errors	RWHS	0	RSB Local Error Registers, address range C00CF8.	PHB: 6	INF

Bit	Field Mnemonic	Type	Reset Value	Description	'OR' of Error Bits	Error Class
23	RSB: DBG Fatal Class Errors	RWHS	0	Debug Registers.	PHB: 13,14	Fatal
24	RSB: DBG INF Class Errors	RWHS	0	Debug Registers.	PHB: 12	INF
25	RSB: PCIe Register Access Size/Align Error	RWHS	0	A Register access to PCIe space had size or address alignment error.	PHB: 11	Fatal
26	RSB: Logic Error	RWHS	0	State error, or Write Data Parity error.	PHB: 15, 27	Fatal
27	RSB: UVI Fatal Class Errors	RWHS	0	RSB Ultravisor Register space Access Errors.	PHB: 17, 18	Fatal
28	RSB: UVI INF Class Errors	RWHS	0	RSB Ultravisor Register space Access Errors.	PHB: 16	INF
29	RSB: SCOM Fatal Class Errors	RWHS	0	Register Access via SCOM interface Errors.	PHB: 30, 31	Fatal
30	RSB: SCOM INF Class Errors	RWHS	0	Register Access via SCOM interface Errors.	PHB: 28, 29	INF
31	PCIe Macro Error Signals Active	RWHS	0	The PCIe macro Error logic has a signal active.	PHB: 24, 25, 26	Class depends on source bits
32	ARB: IODA Fatal Error	RWHS	0	A fatal internal IODA logic error was detected.	ARB: 33	Fatal
33	ARB: IODA MSI PE Mismatch Error	RWHS	0	A mismatch error occurred between the MIST and RTT tables.	ARB: 27	ER (SINGLE)
34	ARB: MSI Size / Address Alignment Error	RWHS	0	An MSI interrupt was received and was not 16 byte address aligned, or the size field was larger than one data beat (16 byte).	ARB: 2,3	Fatal
35	ARB: IODA TVT Errors	RWHS	0	TCE Validation Table error occurred. The entry is invalid, or the PCI Address was out of range as defined by the TTA bounds in the TVE entry.	ARB: 26, 28	ER (SINGLE)
36	ARB: PCIe Fatal Error Message Received	RWHS	0	The PHB4 received a ERR_FATAL message from an endpoint.	ARB: 57	ER (PELTV)
37	ARB: PCIe Nonfatal Error Message Received	RWHS	0	The PHB4 received an ERR_NONFATAL message from an endpoint.	ARB: 58	ER (PELTV)
38	ARB: PCIe Correctable Error Message Received	RWHS	0	The PHB4 received an ERR_CORR message from an endpoint.	ARB: 59	INF
39	ARB: PAPR Inbound Injection Error Triggered	RWHS	0	The condition set up by the PAPR Error Injection Registers triggered its intended condition to inject an inbound error. This is more of a status than an error, but it can be treated like a normal error.	ARB: 39	ER (SINGLE)
40	ARB: Inbound Fatal Errors	RWHS	0	A fatal inbound logic or array error occurred. See ARB Error Status register bits for details.	ARB: 8, 9, 10, 11, 12, 13, 14, 15, 22, 36, 37, 38, 42, 43, 44, 45, 46, 47, 56 ARB: 18 (vA4.1) ARB: 54 (vA4.2)	Fatal
41	ARB: Internal BAR Disabled Error	RWHS	0	An access was made to a resource and the resource BAR register is not properly set up. In ARB this is the PELTV, RTT.	ARB: 32, 41	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	'OR' of Error Bits	Error Class
42	ARB: Inbound Completion Status not zeros	RWHS	0	A completion TLP had completion status not equal zeros. This status is Unsupported Request (UR) or Completor Abort (CA). In the case of a CFG access, it could include retry status (CRS).	ARB: 0, 1, 19 (Fatal)	Fatal (all) (vA4.1) ER (SINGLE) Fatal (19 only) (vA4.2)
43	ARB: PCT Errors	RWHS	0	A PCI completion timeout occurred for an outstanding PCIe transaction, or an unexpected PCI completion was received and did not match any outstanding PCI requests.	ARB: 34, 35	Fatal
44	ARB: Inbound ECC Correctable Error	RWHS	0	A single bit, correctable error occurred in one of the arrays in the inbound logic.	ARB: 25, 29 ARB: 18 (vA4.2)	INF
45	ARB: Inbound ECC Uncorrectable Error	RWHS	0	A double-bit, uncorrectable error occurred in one of the arrays in the inbound logic.	ARB: 30, 31	Fatal
46	ARB: Inbound TLP Data Poisoned Error OR ARB: Secure Address Error (vA4.2)	RWHS	0	A TLP was received and its TLP header indicates that the data was poisoned. Secure Address Error (vA4.2) Non-translated DMA address matched against secure address checker. Refer to *UV - Secure Address Exclude CMP/MSK Register* on page 295.	ARB: 24 ARB: 20 (vA4.2) ARB: 54 (vA4.1)	ER (SINGLE)
47	ARB: RTC PE# Invalid Error	RWHS	0	The PE number in the RTE was all ones (all 16 bits). This indicates that it is an invalid entry.	ARB: 40	INF
48	MRG: Inbound Fatal Errors	RWHS	0	A fatal inbound logic or array error occurred. See MRG Error Status register bits for details.	MRG: 8 to 16,21,22,23,26,28,30,31,32 to 38,48 to 50 MRG: 25 (vA4.2)	Fatal
49	MRG: Internal BAR Disabled Error	RWHS	0	An access was made to a resource and the resource BAR register is not properly set up. In MRG this is the PEST.	MRG: 51	Fatal
50	MRG: Inbound ECC Correctable Error	RWHS	0	A single bit, correctable error occurred in one of the arrays in the inbound logic.	MRG: 40, 56, 58, 60	INF
51	MRG: Inbound ECC Uncorrectable Error	RWHS	0	A double-bit, uncorrectable error occurred in one of the arrays in the inbound logic.	MRG: 41, 57, 59, 61	Fatal
52	AIB TX Timeout Error	RWHS	0	An inbound transaction, request or response, could not be sent on the AIB TX interface before a set timeout period.	MRG: 24	Fatal
53	Migration Register Table Error	RWHS	0	A Migration Register Table error occurred.	MRG: 17,18	ER (SINGLE)

Bit	Field Mnemonic	Type	Reset Value	Description	'OR' of Error Bits	Error Class
54	MSI Secure Address Error	RWHS	0	This error checks that the interrupt notification address is within the secure range when the secure check is enabled. If the address is not in the secure range it will flag this error. Refer to the interrupt notification registers starting on page 110 and the secure control registers starting on page 297.	MRG: 39	Fatal
55	pb_etu_ai_rx_raise_fence (rising edge)	RWHS	0	PB logic above the PHB4 asserted its raise fence signal. This error is also routed to the LEM FIR. This can be used for debug in conjunction with the LEM WOF to test if the PB logic asserted its fence before other PHB4 errors. The PHB4 logic detects the error as a rising edge of the PB signal. The error will not reassert if the PB signal remains constant.	MRG: 63	Fatal
56	TCE: IODA Page Access Error	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 1,2 TCE: 32,33 (vA4.2)	ER (SINGLE)
57	TCE: TCE Request Timeout Error	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 8	Fatal
58	TCE: TCE Unexpected Response Error	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 13	Fatal
59	TCE: TCE Common Fatal Errors	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 0,3,4,6-8,13-14,16-24,29-31 TCE: 5 (vA4.2)	Fatal
60	TCE: ECC Correctable Error	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 9,11	INF
61	TCE: ECC Uncorrectable Error	RWHS	0	Refer to *Table 4.4.4.54 RXE_TCE Error Status Register* on page 159.	TCE: 10,12	Fatal
62	Reserved	RWHS	0	Reserved		
63	LEM FIR Internal Parity Error PCI Clock Domain SCOM Error	RWHS	0	Parity error on the LEM FIR has occurred. This signal is from inside the local FIR macro logic. The SCOM error is asserted from a source signal external to the PHB4. It indicates there was an SCOM state machine error during a register operation.	Direct, not from trap	Fatal

4.4.4.2 LEM FIR AND Mask Register

This is a write-only address used for the sole purpose of masking certain bits in the LEM FIR Accumulator Register. This address is used as an AND mask for the LEM FIR Accumulator Register. Partial writes to this address are not supported. Reads of this address are not supported.

Mnemonic LEM_FIR_AND_MR

Address Offset 0x0C08

Attributes AND: 0x8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIR Accumulator	WOAND	0	Software must write a desired value at this address to mask off a specific bit or bits in the LEM FIR Accumulator Register. Once the value is written to this address, the mask is applied and the contents of the LEM FIR Accumulator Register will be updated. Writing to this address performs a bitwise AND function. The value written is ANDed, bit for bit, with the current LEM FIR Accumulator Register value, and the results of that action become the new register value. Where a mask and register bit are both ‘1’, the resulting bit will be ‘1’. Therefore, software must write bits to 0 that it wants to clear and bits to 1 that it wants to remain unchanged in the LEM FIR Accumulator Register.

4.4.4.3 LEM FIR OR Mask Register

This is a write-only address used for the sole purpose of masking certain bits in the LEM FIR Accumulator Register. This address is used as an OR mask for the LEM FIR Accumulator Register. Partial writes to this address are not supported. Reads of this address are not supported.

Mnemonic LEM_FIR_OR_MR

Address Offset 0x0C10

Attributes OR: 0x10

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIRAccumulator	WOOR	0	Software must write a desired value in this address to mask off a specific bit or bits in the LEM FIR Accumulator Register. Once the value is written to this address, the mask is applied and the contents of the LEM FIR Accumulator Register will be updated. Writing to this address performs a bitwise OR function. The value written is ORed, bit for bit, with the current LEM FIR Accumulator Register value, and the results of that action become the new register value. Where either a mask and register bit are ‘1’, the resulting bit will be ‘1’. Therefore, software must write bits to ‘1’ that it wants to set and bits to ‘0’ that it wants to remain unchanged in the LEM FIR Accumulator Register.

4.4.4.4 LEM Error Mask Register

This register contains the mask for the LEM FIR Accumulator Register. Partial reads or writes to this register are not supported. Software can use the LEM Error And/Or Mask Registers to clear and set bits, thus avoiding a read-modify- write operation. The Mask register is all ones at reset. This prevents any spurious errors from signaling an interrupt until the logic is initialized.

Mnemonic LEM_EMR

Address Offset 0x0C18

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIR Accumulator	RWS	0xFFFFFFFFFFFFFFFF	The bits in this register correspond to the bits in the LEM FIR Accumulator Reg ister. Bit = 0: do not mask corresponding LEM FIR Accumulator error bit. Bit = 1: mask the corresponding LEM FIR Accumulator error bit. This register masks the individual FIR Accumulator bits from being reported. A masked FIR Accumulator bit will prevent any of the LEM Action 0/1 Regis- ter side effects from occurring.

4.4.4.5 LEM Error AND Mask Register

This is a write-only address used for the sole purpose of masking certain bits in the LEM Error Mask Register. This address is used as an AND mask for the LEM Error Mask Register. Partial writes to this address are not supported. Reads of the address are not supported.

Mnemonic LEM_E_AND_MR

Address Offset 0x0C20

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as Error Mask	WO	0	Software must write a desired value to this address to mask off a specific bit or bits in the LEM Error Mask Register. Once the value is written to this address, the mask is applied and the contents of the LEM Error Mask Register will be updated. Writing to this address performs a bitwise AND function. The value written is ANDed, bit for bit, with the current LEM Error Mask Register value, and the results of that action become the new register value. Where a mask and register bit are both ‘1’, the resulting bit will be ‘1’. Therefore, software must write bits to 0 that it wants to clear and bits to 1 that it wants to remain unchanged in the LEM Error Mask Register.

4.4.4.6 LEM Error OR Mask Register

This is a write-only address used for the sole purpose of masking certain bits in the LEM Error Mask Register. This address is used as an OR mask for the LEM Error Mask Register. Partial writes to this address are not supported. Reads of this address are not supported.

Mnemonic LEM_E_OR_MR

Address Offset 0x0C28

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as Error Mask	WO	0	Software must write a desired value in this address to mask off a specific bit or bits in the LEM Error Mask Register. Once the value is written to this address, the mask is applied and the contents of the LEM Error Mask Register will be updated. Writing to this address performs a bitwise OR function. The value written is ORed, bit for bit, with the current LEM Error Mask Register value, and the results of that action become the new register value. Where either a mask and register bit are ‘1’, the resulting bit will be ‘1’. Therefore, software must writes bits to ‘1’ that it wants to set and bits to ‘0’ that it wants to remain unchanged in the LEM Error Mask Register.

4.4.4.7 LEM Action 0 Register

This is register contains bit 0 of the ‘Action’ encoding for each FIR Accumulator bit. Partial reads or writes to this register are not supported. This register is all ones at reset.

Mnemonic LEM_A_0_R

Address Offset 0x0C30

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIR Accumulator	RWS	0xFFFFFFFF_FFFFFFFF	This register is combined with the LEM Action 1 Register to form a two bit ‘Action’ for each of the LEM FIR Accumulator Register error bits. Each FIR error bit can have a different side effect, this is its action. The encoding for the ‘Action’ is as follows: Action(0:1). 00 - Checkstop Error. 01 - Recoverable Error. 10 - Recoverable Interrupt (not used). 11 – No Action.

4.4.4.8 LEM Action 1 Register

This register contains bit 1 of the ‘Action’ encoding for each FIR Accumulator bit. Partial reads or writes to this register are not supported. This register is all ones at reset.

Mnemonic LEM_A_1_R

Address Offset 0x0C38

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIR Accumulator	RWS	0xFFFFFFFF_FFFFFFFF	This register is combined with the LEM Action 0 Register to form a two bit ‘Action’ for each of the LEM FIR Accumulator Register error bits. Each FIR error bit can have a different side effect, this is its action. The encoding for the ‘Action’ is as follows: Action(0:1). 00 Checkstop Error. 01 Recoverable Error. 10 Recoverable Interrupt (not used). 11 No Action.

4.4.4.9 LEM WOF Register

This register contains ‘Who’s on first?’ (WOF) error indicator bits. These bits identify which one of the LEM FIR Accumulator Register bits was asserted first. This identifies the first error in the FIR. Partial reads or writes to this register are not supported. This register will automatically zero out when the LEM FIR is zeroed out. This is the default behavior for the LEM “ASIC” mode.

Mnemonic LEM_WOF_R

Address Offset 0x0C40

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Same as FIR Accumulator	RWHS	0	Each bit in this register corresponds to the same bit in the LEM FIR Accumulator Register. Only one of these bits should be asserted at a time, which indicates which of the FIR error bits was asserted first. Software can write this register will all zeros to clear it. Writing this register with a nonzero value can yield an unpredictable result.

4.4.4.10 PHB4 Error Status Register

This register contains error status information for the PHB4 Register block and PHB4 General Errors.

Mnemonic PHB_FESR

Address Offset 0x0C88

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	ETU/RSB Request Address Error	RW1CHS	0	SCOM or AIB register access to non-existent register address. Address did not match any register sets within the ETU_RSB macro.	INF
1	Fundamental A Request Address Error	RW1CHS	0	ETU/Fundamental A Registers: SCOM or AIB register access to non-existent register address.	INF
2	Fundamental A Request Size/Alignment Error	RW1CHS	0	ETU/Fundamental A Registers: SCOM or AIB register access was not 8-byte size and/or aligned.	Fatal
3	Fundamental A PCI CFG Addr/Size Error	RW1CHS	0	ETU/Fundamental A Registers: SCOM or AIB, PCI CFG_DATA access addr+size crossed a 4-byte boundary.	Fatal
4	Fundamental A IODA Table Access Error	RW1CHS	0	ETU/Fundamental A Registers: SCOM or AIB, IODA table access to unused/reserved table select in the table address register.	INF
5	Fundamental A Internal Registers Parity Error	RW1CHS	0	ETU/Fundamental A Registers: Internal parity error detected in the registers.	Fatal
6	PHB Error Registers Request Address Error	RW1CHS	0	ETU/Error Registers: SCOM or AIB register access to non-existent register address.	INF
7	PHB Error Registers Request Size/Alignment Error	RW1CHS	0	ETU/Error Registers: SCOM or AIB register access was not 8-byte size and/or aligned.	Fatal
8	Fundamental B Request Address Error	RW1CHS	0	ETU/Fundamental B Registers: SCOM or AIB register access to non-existent register address.	INF
9	Fundamental B Request Size/Alignment Error	RW1CHS	0	ETU/Fundamental B Registers: SCOM or AIB register access was not 8-byte8-bytesize and/or aligned.	Fatal
10	Fundamental B Internal Registers Parity Error	RW1CHS	0	ETU/Fundamental B Registers: Internal parity error detected in the registers.	Fatal
11	Internal Bus Logic Bad PCIe Macro Request Address	RW1CHS	0	Internal Bus Logic Bad PCIe Macro Request Address sent from the ETU/RSB to the PCIe macro.	Fatal
12	Debug Request Address Error	RW1CHS	0	ETU/Debug Registers: SCOM or AIB register access to non-existent register address.	INF
13	Debug Request Size/Alignment Error	RW1CHS	0	ETU/Debug Registers: SCOM or AIB register access was not 8-byte size and/or aligned.	Fatal

1. The OR Error Status bits are only used to capture which error trap occurred first. The initialization sequence will force the interrupt enable bits for these bits to zero.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
14	Debug Internal Registers Parity Error	RW1CHS	0	ETU/Debug Registers: Internal parity error detected in the registers.	Fatal
15	Internal Bus Logic State Machine One-Hot Error	RW1CHS	0	Internal Bus Logic State Machine One-Hot Error	Fatal
16	UV Page Request Address Error	RW1CHS	0	ETU/UV Page Registers: SCOM or AIB register access to non-existent register address.	INF
17	UV Page Request Size/Alignment Error	RW1CHS	0	ETU/UV Page Registers: SCOM or AIB register access was not 8-byte size and/or aligned.	Fatal
18	UV Page Internal Registers Parity Error	RW1CHS	0	ETU/UV Page Registers: Internal parity error detected in the registers.	Fatal
19	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
20	RXE_ARB OR Error Status	RW1CHS	0	RXE_ARB error bits, page 145. OR of all error status bits. This is used to test which macro error trap error occurred first among all the error macros.	INF[1]
21	RXE_MRG OR Error Status	RW1CHS	0	RXE_MRG error bits, page 153. OR of all error status bits. This is used to test which macro error trap error occurred first among all the error macros.	INF[1]
22	RXE_TCE OR Error Status	RW1CHS	0	RXE_TCE error bits, page 159. OR of all error status bits. This is used to test which macro error trap error occurred first among all the error macros.	INF[1]
23	TXE OR Error Status	RW1CHS	0	TXE error bits, page 136. OR of all error status bits. This is used to test which macro error trap error occurred first among all the error macros.	INF[1]
24	pcie_etu_regb_err_inf	RW1CHS	0	Error signal from PCIE/REGB. Refer to REGB error status bits, page 285.	INF
25	pcie_etu_regb_err_erc	RW1CHS	0	Error signal from PCIE/REGB. Refer to REGB error status bits, page 285.	ER (ALL)
26	pcie_etu_regb_err_fat	RW1CHS	0	Error signal from PCIE/REGB. Refer to REGB error status bits, page 285.	Fatal
27	bus_regs_req_wr_data_p_e	RW1CHS	0	Internal register bus write data parity error.	Fatal
28	SCOM HV Indirect Access Error	RW1CHS	0	SCOM HV Indirect Access Error. The indirect valid bit was not set in the indirect address register for the access. Therefore, the internal logic sent a NAK response on the SCOM bus.	INF
29	SCOM UV Indirect Access Error	RW1CHS	0	SCOM UV Indirect Access Error. The indirect valid bit was not set in the indirect address register for the access. Therefore, the internal logic sent a NAK response on the SCOM bus.	INF
30	SCOM Internal Registers Parity Error	RW1CHS	0	SCOM Registers: Internal parity error detected in the registers.	Fatal

1. The OR Error Status bits are only used to capture which error trap occurred first. The initialization sequence will force the interrupt enable bits for these bits to zero.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
31	SCOM Satellite Finite State Machine Error	RW1CHS	0	An error detected in the SCOM satellite finite state machine.	Fatal
32:39	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
40:63	Reserved	RO	0	Reserved, not implemented	Fatal

1. The OR Error Status bits are only used to capture which error trap occurred first. The initialization sequence will force the interrupt enable bits for these bits to zero.

4.4.4.11 PHB4 First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the PHB4 Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic PHB_FESR

Address Offset 0x0C88

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWHS	0	Bits correspond to the error that occurred first.

4.4.4.12 PHB4 Error Injection Register

This register controls error injection for the individual error bits defined in the Error Status Register. Writing bits to a ‘1’ in this register will inject the error corresponding to the bit that is written. The bits will automatically clear to ‘0’ after the error is injected. The corresponding bit in the Error Status Register is set automatically when the error occurs. The individual injection bit will clear to '0' after the hardware performs the injection action for that bit.

Mnemonic PHB_EIR

Address Offset 0x0C90

Bit	Field Mnemonic	Type	Reset Value	Description
0	Reserved	RWH	0	Reserved, but implemented for future use.
01:63	Reserved	RO	0	Reserved.

4.4.4.13 PHB4 Error LEM Report Enable Register

This register enables a PHB4 error to report that error through the PHB4 LEM structure. The bits of this register match bit for bit to those defined in the PHB4 Error Status Register. Setting a bit to '1' will enable that error.

Mnemonic PHB_E_LEM_RER

Address Offset 0x0C98

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWH	0	Bits correspond to the error that will be driven to its specific LEM bit.

4.4.4.14 PHB4 Error System Interrupt Enable Register

This register enables a PHB4 error to report that error through the system interrupt structure. The bits of this register match bit for bit to those defined in the PHB4 Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single PHB4 system interrupt.

Mnemonic PHB_E_SYS_IER

Address Offset 0x0CA0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWH	0	Bits correspond to the errors that will send a PHB4 sourced LSI error interrupt.

4.4.4.15 PHB4 Error EEH Freeze Enable Register

This register enables a PHB4 error to place the PHB4 or an endpoint in a EEH freeze state. The bits of this register match bit for bit to those defined in the PHB4 Error Status Register. Setting a bit to '1' will enable that error.

Mnemonic PHB_E_EEH_FER

Address Offset 0x0CA8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWH	0	Bits correspond to the error that will place the endpoint in the EEH freeze state. Generally, bits labeled ER class will set the EEH state and issue a write to the PEST.

4.4.4.16 PHB4 Error AIB Fence Enable Register

This register enables a PHB4 error to place the PHB4 in a AIB interface fenced condition. The bits of this register match bit for bit to those defined in the PHB4 Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single PHB4 AIB Fence signal to the fence logic.

Mnemonic PHB_E_AIB_FER

Address Offset 0x0CB0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWH	0	Bits correspond to the errors that will fence the AIB bus. Generally, bits labeled FATAL class will set the fence state.

4.4.4.17 PHB4 Error Log Register 0

This register contains error log information for errors that are active in the PHB4 Error Status register. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic PHB_ELR_0

Address Offset 0x0CC0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWH	0	See the individual Error Status Register error bits for log register bit assignments.

4.4.4.18 PHB4 Error Log Register 1

This register hold error log information for errors that are active in the PHB4 Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

Mnemonic PHB_ELR_1

Address Offset 0x0CC8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWHS	0	Each mask bit corresponds to an error bit in the PHB4 Error Status Register.

4.4.4.19 PHB4 Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

Because link sourced error messages are now handled in the normal inbound command path, bits [33:35] of this register should be masked to prevent error message reporting through this register set.

Mnemonic PHB_ESMR

Address Offset 0x0CD0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWHS	0	Each mask bit corresponds to an error bit in the PHB4 Error Status Register.

4.4.4.20 PHB4 First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It should be used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the First Error Status Register. It does not prevent the side effects of the error.

In addition to masking the First Error Status Register, this mask will prevent an error from capturing logging information in the error log registers.

Because Link sourced error messages are now handled in the normal inbound command path, bits [33:35] of this register should be masked to prevent error message reporting through this register set.

Mnemonic PHB_FESMR

Address Offset 0x0CD8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 130.	RWHS	0	Each mask bit corresponds to an error bit in the PHB4 First Error Status Register.

4.4.4.21 TXE Error Status Register

This register contains error status information for the TXE (MMIO) directed errors. No error interrupt should be sent for the SUE case, bit 46. The initialization sequence settings will turn off the interrupt generation for this error bit. It is INF class, but it is just recorded in the TXE error trap and LEM FIR.

Mnemonic TXE_ESR

Address Offset 0x0D0

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	AIB Command Invalid	RW1CHS	0	The command field of an incoming AIB packet contains a command not recognized as valid by the PHB4. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
1	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
2	AIB Address Decode Error – No BAR Match	RW1CHS	0	The address field of an incoming AIB packet matches none of the address regions supported by PHB4. This means the address decode missed all enabled MBT entries, does not target internal register space, and is not a peer-to-peer write. The TXE Log Registers 0 and 1 will contain the AIB command header.	INF
3	AIB Size Invalid	RW1CHS	0	The Size field of an incoming AIB packet is not valid. PHB4 supports the following lengths for commands with data: 1-8, 16, 32, 64, 128 bytes. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
4	AIB Cmd Ctrls Parity Error	RW1CHS	0	Parity on the Control signals associated with the AIB Command Bus is not valid. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
5	AIB Data Ctrls Parity Error	RW1CHS	0	Parity on the Control signals associated with the AIB Data Bus is not valid. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
06:07	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
8	AIB Alignment Error	RW1CHS	0	The size field ≥ 8 of an incoming AIB packet is not on a naturally aligned address. All AIB packets with sizes equal or greater than 8 must be naturally aligned. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
9	AIB Cmd Bus Parity Error	RW1CHS	0	Parity on the AIB Command Bus is not valid. Bits <32:47> of TXE Log Register 1 will contain the parity error vector.	Fatal
10	AIB Data Bus UE ECC Error	RW1CHS	0	UE ECC detected on the AIB Data Bus. If the MMIO targets the link, the PHB4 will set the BLIF signal - etu_pcie_blif_out_dat_err, so the packet will get nullified. Bits <48:63> of TXE Log Register 1 will contain the Syndrome and error vector information.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
11	AIB Data Ctrls Sequence Error	RW1CHS	0	The Data Control signals were in some illegal sequence state. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
12	AIB Data Bus CE ECC Error	RW1CHS	0	CE ECC detected on the AIB Data Bus. Bits <48:63> of TXE Log Register 1 will contain the Syndrome and error vector information.	INF
13	TCE Read Response DAT_ERR Indication	RW1CHS	0	The PHB4 received a TCE Read Response, and the DAT_ERR signals were set to discard the packet. The packet will be discarded and eventually this will trigger a TCE timeout error.	Fatal
14	AIB Command Credits Error	RW1CHS	0	A command was received, and the command credit count for that channel was zero. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
15	AIB Data Credits Error	RW1CHS	0	A command with data was received, and the data credit count for that channel was zero. The TXE Log Registers 0 and 1 will contain the AIB command header.	Fatal
16	BLIF Controls Parity Error	RW1CHS	0	The BLIF outbound interface control signals (credit or ready) parity check failed.	Fatal
17	CFG Write Error CA or UR response	RW1CHS	0	The PCIe logic received a Completion with CA or UR response for the current CFG Write command. The CONFIG_ADDR and CONFIG_DATA registers will contain the information about the current CFG command.	ER (SINGLE)
18	BLIF Forward Progress Timeout	RW1CHS	0	A timeout has occurred on BLIF interface. This will occur when the PCIe can make no forward progress on TXE MMIO request packets.	Fatal
19	MMIO RD Pending Error	RW1CHS	0	The PHB4 received another MMIO Read Req while one is still pending. A request is considered pending when a credit for the request has not been returned over AIB.	Fatal
20	MMIO WR Pending Error	RW1CHS	0	The PHB4 received another MMIO Write while one is still pending. A command is considered pending when a credit for the command has not been returned over AIB.	Fatal
21	MMIO CFG Pending Error	RW1CHS	0	The PHB4 received another CFG command while one is still pending. PHB4 can only service one CFG command at a time. A command is considered pending when a credit for the command has not been returned over AIB.	Fatal
22	MMIO Write DAT_ERR Indication	RW1CHS	0	The PHB4 received a MMIO Write command, and the DAT_ERR signals were set to discard the packet. The PHB4 will set the BLIF signal - *etu_pcie_blif_out_dat_err*, so the packet will get nullified.	Fatal
23	CI Store Data Fifo Error	RW1CHS	0	The CI Store data fifo took a fatal error. This includes underflow/overflow, internal parity or entry valid error. Log0 will contain details of the specific error.	Fatal
24	CFG Enable Error, RRB	RW1CHS	0	A CFG access was attempted but the Enable bit was not set.	INF
25	CFG Size Error	RW1CHS	0	A CFG access was attempted with an illegal size. Only lengths of 1-4 bytes are valid. See description of the CONFIG_DATA register page 69 for valid size/address combinations.	INF

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
26	CFG Bus Address Error	RW1CHS	0	A CFG access was attempted and at least one of the following is true: • The BUS/DEV fields in the CONFIG_ADDR register are out of bounds. • A CFG access was attempted while the cores are reset (PGRESET active). • A CFG access was attempted while the PHB4 is fenced.	INF
27	CFG Link Down Error	RW1CHS	0	A CFG access was attempted but the PCIe link is currently in the “down” state. The access can be from SCOM or AIB.	INF
28	PAPR TXE Injection Error Triggered (AIB/MMIO operation)	RW1CHS	0	The condition set up by the PAPR Error Injection Registers triggered its intended condition to inject an TXE error (for an AIB/MMIO operation). This is more of a status than an error, but it can be treated like a normal error.	ER (SINGLE)
29	CFG Write Request Timeout	RW1CHS	0	The timeout has occurred for a pending CFG Write request. CFG Read requests will be covered by the PCT timeout timer.	INF
30	PAPR TXE Injection Error Triggered (PCI CFG operation)	RW1CHS	0	The condition set up by the PAPR Error Injection Registers triggered its intended condition to inject an TXE error (for PCI CFG operation). This is more of a status than an error, but it can be treated like a normal error.	ER (SINGLE)
31:35	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
36	CI Trigger Buffer ECC Correctable Error	RW1CHS	0	A correctable ECC error (CE) was detected on CI Trigger Buffer. TXE Log Register 1 will contain details of the error.	INF
37	CI Trigger Buffer ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error (UE) was detected on CI Trigger Buffer. TXE Log Register 1 will contain details of the error.	Fatal
38	CI Trigger Buffer Stage Data Parity Error	RW1CHS	0	A Parity error was detected on CI Trigger Buffer. TXE Log Register 1 will contain details of the error.	Fatal
39	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
40	MMIO BAR Table (MBT) Parity Error	RW1CHS	0	MBT table detected an internal parity error.	Fatal
41	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
42	MMIO Domain Table (MDT) ECC Correctable Error	RW1CHS	0	A correctable ECC error (CE) was detected by the MMIO Domain Table logic.	INF
43	MMIO Domain Table (MDT) ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error (UE) was detected by the MMIO Domain Table logic.	Fatal
44	MMIO Domain Table (MDT) Stage Parity Error	RW1CHS	0	An internal stage parity error was detected by the MMIO Domain Table logic.	Fatal
45	MMIO Domain Table (MDT) Stage Valid Error	RW1CHS	0	An internal stage valid error was detected by the MMIO Domain Table logic.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
46	AIB Data Special Uncorrectable Error (SUE)	RW1CHS	0	A 'Special' uncorrectable ECC (SUE) error was indicated on the AIB data received from the PB block. The PB LEM FIR will provide details on why the SUE was forwarded. If the MMIO targets the link, the PHB4 will set the BLIF signal - etu_pcie_blif_out_dat_err, so the packet will get nullified.	INF*
47	MMIO Domain Table (MDT) Read Pipe Error	RW1CHS	0	A fatal error occurred while accessing the MMIO Domain Table logic. This includes overflow/underflow, data parity, and valid errors.	Fatal
48	P2P Store Data Fifo Error	RW1CHS	0	The P2P Store data fifo took a fatal error. This includes under-flow/overflow, internal parity or entry valid error. Log0 will contain details of the specific error.	Fatal
49	EPAT Table Parity Error	RW1CHS	0	A parity error was detected by the End Point Active Table (EPAT) logic.	Fatal
50	MMIO Cmd Parity Error	RW1CHS	0	A parity error has occurred in the MMIO command pipeline.	Fatal
51	BLIF1 Register Parity Error	RW1CHS	0	A parity error has occurred in the register that drives the BLIF Com-mand for CI loads and stores.	Fatal
52	P2P1 Register Parity Error	RW1CHS	0	A parity error has occurred in the register that drives the BLIF Com-mand for P2P stores.	Fatal
53	P2P WR Pending Error	RW1CHS	0	The ETU received another P2P Write while one is still pending. A command is considered pending when a credit for the command has not been returned over AIB.	Fatal
54	CRW Onehot Error	RW1CHS	0	The Credit Return Write State Machine signaled a state (onehot) error.	Fatal
55	CRW Pending Error	RW1CHS	0	The CRW SM issued another CRW Write while one is still pending.	Fatal
56	RRB Parity Error	RW1CHS	0	A parity error was detected in the RRB block for a register imple-mented in the TXE macro.	Fatal
57	RRB Size/Alignment Error	RW1CHS	0	A read/write access to a register had incorrect size or address align-ment.	Fatal
58	s_bad_addr_e_q	RW1CHS	0	TXE local error, internal register bad address error.	Fatal
59	s_req_size_align_e_q	RW1CHS	0	TXE local error, internal register read / write access to a register had incorrect size or address alignment.	Fatal
60:63	Reserved	RO	0	Reserved.	

4.4.4.22 TXE First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the TXE Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic TXE_FESR

Address Offset 0x0D08

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWHS	0	Bits correspond to the error that occurred first.

4.4.4.23 TXE Error Injection Register

This register is used for error injection. Setting the bit in this register will cause the selected error to occur by forcing the error condition. Writing bits to a ‘1’ in this register will schedule an inject of the error corresponding to the bit that is written. The bit(s) will automatically clear to ‘0’ as each error is injected. The corresponding bit in the Error Status Register is set automatically when the error inject occurs. The individual injection bit will clear to '0' after the hardware performs the injection action for that bit.

Mnemonic TXE_EIR

Address Offset 0x0D10

Bit	Field Mnemonic	Type	Reset Value	Description
0	AIB Command Invalid	RWH	0	Error will be injected on the next command received.
1	Reserved	RO	0	Reserved.
2	AIB Address Decode Error – No BAR Match	RWH	0	Error will be injected on the next command received.
3	AIB Size Invalid	RWH	0	Error will be injected on the next command received.
4	AIB Cmd Ctrls Parity Error	RWH	0	Error will be injected on the next command received.
5	AIB Data Ctrls Parity Error	RWH	0	Error will be injected on the next command received.
06:07	Reserved	RO	0	Reserved.
8	AIB Alignment Error	RWH	0	Error will be injected on the next command received.
9	AIB Cmd Bus Parity Error	RWH	0	Error will be injected on the next command received.

Bit	Field Mnemonic	Type	Reset Value	Description
10	AIB Data Bus UE ECC Error	RWH	0	Error will be injected on the next command received.
11	AIB Data Ctrls Sequence Error	RWH	0	Error will be injected on the next command received.
12	AIB Data Bus CE ECC Error	RWH	0	Error will be injected on the next command received.
13:17	Reserved	RO	0	Reserved.
18	BLIF Forward Progress Timeout	RWH	0	Error will be injected on the next command received that targets the link.
19	MMIO RD Pending Error	RWH	0	Error will be injected on the next CI Load command received that targets the link.
20	MMIO WR Pending Error	RWH	0	Error will be injected on the next CI Store command received that targets the link.
21	MMIO CFG Pending Error	RWH	0	Error will be injected on the next CFG request generated.
22:28	Reserved	RO	0	Reserved.
29	CFG Write Request Timeout	RWH	0	Error will be injected on the next CFG Write request generated.
30:35	Reserved	RO	0	Reserved.
36	CI Trigger Buffer ECC Correctable Error	RWH	0	Error will be injected the next time the Trigger Buffer is accessed by a CI Store to generate an interrupt.
37	CI Trigger Buffer ECC Uncorrectable Error	RWH	0	Error will be injected the next time the Trigger Buffer is accessed by a CI Store to generate an interrupt.
38:41	Reserved	RO	0	Reserved.
42	MMIO Domain Table (MDT) ECC Correctable Error	RWH	0	Error will be injected on the next command where the MDT is accessed.
43	MMIO Domain Table (MDT) ECC Uncorrectable Error	RWH	0	Error will be injected on the next command where the MDT is accessed.
44:49	Reserved	RO	0	Reserved.
50	MMIO Cmd Parity Error	RWH	0	Error will be injected on the next command received that targets the link.
51	BLIF1 Register Parity Error	RWH	0	Error will be injected on the next command received that targets the link.
52	P2P1 Register Parity Error	RWH	0	Error will be injected on the next P2P Write command received.
53	P2P WR Pending Error	RWH	0	Error will be injected on the next P2P Write command received.
54	Reserved	RO	0	Reserved.
55	CRW Pending Error	RWH	0	Error will be injected on the next CRW command generated.
56:63	Reserved	RO	0	Reserved.

4.4.4.24 TXE Error LEM Report Enable Register

This register enables a TXE Error to report that error through the PHB4 LEM structure. The bits of this register match bit for bit to those defined in the TXE Error Status Register. Setting a bit to ‘1’ will enable that error. All the enabled errors are ORed together to create a single interrupt.

Mnemonic TXE_LEM_RER

Address Offset 0x0D18

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWH	0	Bits correspond to the error that will be driven to its specific LEM bit.

4.4.4.25 TXE Error System Interrupt Enable Register

This register enables a PHB4 error to report that error through the system interrupt structure. The bits of this register match bit for bit to those defined in the TXE Error Status Register. Setting a bit to ‘1’ will enable that error. All the enabled errors are ORed together to create a single PHB4 system interrupt.

Mnemonic TXE_SYS_IER

Address Offset 0x0D20

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWH	0	Bits correspond to the errors that will send a PHB4 sourced LSI error interrupt.

4.4.4.26 TXE Error EEH Freeze Enable Register

This register enables a TXE error to place the PHB4 or an endpoint in a EEH freeze state. The bits of this register match bit for bit to those defined in the TXE Error Status Register. Setting a bit to ‘1’ will enable that error to set the freeze state.

Mnemonic TXE_EEH_FER

Address Offset 0x0D28

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWH	0	Bits correspond to the errors that will set the EEH freeze state. These errors set EEH freeze/error bits for all endpoints or a specific endpoint depending on the error.

4.4.4.27 TXE Error AIB Fence Enable Register

This register enables a TXE error to place the PHB4 in a AIB interface Fenced condition. The bits of this register match bit for bit to those defined in the TXE Error Status Register. Setting a bit to '1' will enable that error. All the enables are ORed together to create a single fence signal to the fence logic.

Mnemonic TXE_AIB_FER

Address Offset 0x0D30

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWH	0	Bits correspond to the errors that will fence the AIB bus.

4.4.4.28 TXE Error Log Register 0

This register contains log information for the TXE Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic TXE_LOG0

Address Offset 0x0D40

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWH	0	See the individual Error Status Register error bits for Log register bit assignments.

4.4.4.29 TXE Error Log Register 1

This register contains additional error log information for TXE Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic TXE_LOG1

Address Offset 0x0D48

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWH	0	See the individual Error Status Register error bits for Log register bit assignments.

4.4.4.30 TXE Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error

Mnemonic TXE_ESMR

Address Offset 0x0D50

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWHS	0	Each mask bit corresponds to an error bit in the Error Status Register.

4.4.4.31 TXE First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the first Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit. If a bit in the First Error mask register is set to a '1' it will prevent the corresponding error from being logged in the First Error Status Register. It does not prevent the side effects of the error. In addition to masking the First Error Status Register, this mask will prevent an error from capturing logging information in the error log registers.

Mnemonic TXE_FESMR

Address Offset 0x0D58

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 136.	RWHS	0	Each mask bit corresponds to an error bit in the First Error Status Register.

4.4.4.32 RXE_ARB Error Status Register

This register contains error status information for the inbound directed errors.

Note: Bit 59, ERR_CORR error message is typically classified as INF but can also be classified as ER(PELTV) for workaround or debug purposes.

Mnemonic RXE_ARB_ESR

Address Offset 0x0D80

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	BLIF Inbound CA Completion Error	RW1CHS	0	An inbound completion TLP was received with status of completer abort (CA).	Fatal (vA4.1) ER (SINGLE) (vA4.2)
1	BLIF Inbound UR Completion Error	RW1CHS	0	An inbound completion TLP was received with status of unsupported request (UR).	Fatal (vA4.1) ER (SINGLE) (vA4.2)
2	MSI Size Error	RW1CHS	0	An MSI interrupt was received on the BLIF inbound interface with a size greater than one data beat (16-byte).	Fatal
3	MSI Address Alignment Error	RW1CHS	0	An MSI interrupt was received and was not 16-byte address aligned.	Fatal
4	Reserved	RW1CHS	0	Reserved.	Fatal
5	BLIF Inbound Header ECC Correctable (CE) Error	RW1CHS	0	An ECC CE detected on an inbound BLIF command header.	INF
6	BLIF Inbound Header ECC Uncorrectable (UE) Error	RW1CHS	0	An ECC UE detected on an inbound BLIF command header.	Fatal
7	ARB Stage Valid Error	RW1CHS	0	More than one stage valid indication was asserted in the same clock cycle.	Fatal
8	TCE Tag Release Unused	RW1CHS	0	A release of an unused TCE Tag is attempted.	Fatal
9	TCE Tag Used, Not Free	RW1CHS	0	An attempt to use a TCE Tag that is already in use.	Fatal
10	ARB MMIO Buffer Overflow	RW1CHS	0	A buffer overflow condition.	Fatal
11	ARB MMIO Buffer Underflow	RW1CHS	0	A buffer underflow condition.	Fatal
12	ARB MMIO Internal Parity Error	RW1CHS	0	An internal logic parity error was detected.	Fatal
13	ARB DMA Buffer Overflow	RW1CHS	0	A buffer overflow condition.	Fatal
14	ARB DMA Buffer Underflow	RW1CHS	0	A buffer underflow condition.	Fatal
15	ARB DMA Internal Parity Error	RW1CHS	0	An internal logic parity error was detected.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
16	BLIF Header Control Bits Parity Error	RW1CHS	0	A parity error was detected on the BLIF header control bits.	Fatal
17	BLIF Data Control Bits Parity Error	RW1CHS	0	A parity error was detected on the BLIF data control bits.	Fatal
18	BLIF Unsupported Request (UR) Error	RW1CHS	0	A command was received on the BLIF interface indicating it was an unsupported request. A 'Nonposted' request will be acknowledged with a Completion response with UR status. A 'Posted' request type will be dropped.	INF
19	BLIF Completion Timeout Error	RW1CHS	0	A completion was received on the BLIF interface with a time-out indication. The PCIE/PBL block detected the timeout and reported it via the BLIF interface.	Fatal
20	SEID Table ECC Correctable (CE) Error (vA4.1) Secure Address Error (vA4.2)	RW1CHS	0	An ECC CE detected in the SEID table. (vA4.1). Secure Address Error (vA4.2). Non-translated DMA address matched against secure address checker. Refer to *Section 4.7.2.1 UV - Secure Address Exclude CMP/MSK Register* on page 295.	INF (vA4.1) ER (SINGLE) (vA4.2)
21	SEID Table ECC Uncorrectable (UE) Error (vA4.1) Reserved (vA4.2)	RW1CHS	0	An ECC UE detected in the SEID table. (vA4.1). Reserved (vA4.2).	Fatal
22	NBW Size Error	RW1CHS	0	A Non-Blocking Write was received on the BLIF inbound interface with invalid size. The legal NBW sizes are 16, 32, 64, and 128B. In addition, the size and address alignment of the transaction must match.	Fatal
23	DEC IODA Table Fatal Error	RW1CHS	0	An internal fatal error detected related to the IODA tables inside the DEC block. This covers the MIST, SEID, and TVT tables.	Fatal
24	TLP Poisoned Error	RW1CHS	0	A DMA Write was received and its TLP header indicates that the data was poisoned. This includes writes that decode as MSI.	ER (SINGLE)
25	MIST ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in this table's array.	INF
26	IODA TVT Entry Invalid	RW1CHS	0	IODA Error: An access to an invalid TVT entry.	ER (SINGLE)
27	MSI PE# Mismatch	RW1CHS	0	The PE# (as fetched from the RTT) for an MSI did not match the PE# assigned to this Source ID that was stored in the MIST.	ER (SINGLE)
28	IODA TVT Address Range Error	RW1CHS	0	IODA Error: The PCI Address was out of range as defined by the TTA bounds in the TVE entry. Note: As of POWER9 PHB4, the logic will not flag an error for 32-bit DMAs that are configured as non-translated.	ER (SINGLE)
29	TVT ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in this table's array.	INF
30	TVT ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in this table's array.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
31	MIST ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in this table's array.	Fatal
32	PELT-V BAR Disabled Error	RW1CHS	0	An Error Message attempted to access the PELT-V BAR when it was disabled.	Fatal
33	IODA Table Parity Error	RW1CHS	0	A parity error occurred in one of the IODA tables in the DEC block.	Fatal
34	PCT Timeout	RW1CHS	0	A timeout occurred in the PCT table. A PCI completion was never returned for a PCIe request.	Fatal
35	PCT Unexpected Completion	RW1CHS	0	A PCI completion was received but did not match any outstanding requests.	Fatal
36	PCT Parity Error	RW1CHS	0	A parity error occurred in the PCI Completion Table.	Fatal
37	DEC Stage Valid Error	RW1CHS	0	More than one stage valid indication was asserted in the same clock cycle.	Fatal
38	DEC Stage Parity Error	RW1CHS	0	A parity error detected in the control pipeline stages.	Fatal
39	PAPR Inbound Injection Error Triggered	RW1CHS	0	The condition set up by the PAPR Error Injection Registers triggered its intended condition to inject an inbound error. This is more of a status than an error, but it can be treated like a normal error.	ER (SINGLE)
40	DMA/MSI: RTE PE Number All Ones Error	RW1CHS	0	The PE number in the RTE was all ones (all 16 bits). This indicates that it is an invalid entry.	INF
41	RTT BAR Disabled Error	RW1CHS	0	A request attempted to access the RTT BAR when it was disabled.	Fatal
42	RTC Internal Parity Error	RW1CHS	0	An internal parity error was detected in the RTC logic.	Fatal
43	RTC Queue Overflow	RW1CHS	0	An overflow condition occurred in one of the RTC queues.	Fatal
44	RTC Queue Underflow	RW1CHS	0	An underflow condition occurred in one of the RTC queues.	Fatal
45	RTC Stage Valid Error	RW1CHS	0	More than one stage valid indication was asserted in the same clock cycle.	Fatal
46	RTC RCAM Bad State Error	RW1CHS	0	An illegal state was reached in the RCAM.	Fatal
47	RTC RCAM Multiple Hit Error	RW1CHS	0	A RTC Requester ID look up in the RCAM hit against multiple entries.	Fatal
48	RRB Parity Error	RW1CHS	0	An internal parity error was detected in the RRB logic.	Fatal
49	RRB request Size / Alignment Error	RW1CHS	0	A request to the Remote Register Block (RRB) for this macro had bad size or address alignment.	INF
50	s_bad_addr_e_q	RW1CHS	0	RXE_ARB local error, internal register bad address error.	Fatal
51	s_req_size_align_e_q	RW1CHS	0	RXE_ARB local error, internal register read / write access to a register had incorrect size or address alignment.	Fatal
52:53	Reserved	RW1CHS	0	Reserved.	Fatal
54	Discontiguous DMA Write Fragmentation Error	RW1CHS	0	An error with an inbound DMA write that had discontiguous byte enables in the last byte enable (LBE). The error tests for the following: 1) LBE is fragmented, and the total byte count is > 8. 2) Address is not 8-byte aligned. This error should be an internal hardware problem because this condition should be caught as a malformed TLP error in the PCIE/PBL block.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
55	LIST Table Parity Error	RW1CHS	0	Parity error detected in the LIST table.	Fatal
56	LKP PEST Data Queue Error	RW1CHS	0	PEST Data Queue signaled error.	Fatal
57	PCIE Fatal Error Message Received	RW1CHS	0	The PHB4 received an ERR_FATAL message from an endpoint or switch.	ER (PELTV)
58	PCIE Nonfatal Error Message Received	RW1CHS	0	The PHB4 received an ERR_NONFATAL message from an endpoint or switch.	ER (PELTV)
59	PCIE Correctable Error Message Received	RW1CHS	0	The PHB4 received an ERR_CORR message from an endpoint or switch.	INF / ER(PELTV)
60:63	Reserved	RW1CHS	0	Reserved.	Fatal

4.4.4.33 RXE_ARB First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the RXE_ARB Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle.

A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_ARB_FESR

Address Offset 0x0D88

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWHS	0	Bits correspond to the error that occurred first.

4.4.4.34 RXE_ARB Error Injection Register

This register is used for error injection. Setting the bit in this register will cause the selected error to occur by forcing the error condition. Writing bits to a ‘1’ in this register will inject the error corresponding to the bit that is written. The bits will automatically clear to ‘0’ after the error is injected. The corresponding bit in the Error Status Register is set automatically when the error occurs. The individual injection bit will clear to '0' after the hardware performs the injection action for that bit.

Mnemonic RXE_ARB_EIR

Address Offset 0x0D90

Bit	Field Mnemonic	Type	Reset Value	Description
0	Inbound CA Completion Error	RWH	0	This error will be injected for the next valid token after this injection bit is asserted.
1	Inbound UR Completion Error	RWH	0	This error will be injected for the next valid token after this injection bit is asserted.
02:28	Reserved	RO	0	Reserved.
29	TVT ECC Correctable Error	RWH	0	This error will be injected for the next valid token after this injection bit is asserted. The error is injected when the TVT entry is written. The error is detected when the TVT entry is read. Example steps: 1) Set this injection bit. 2) Write to a TVT entry (the data will have bad ECC). 3) Issue a DMA that will target that TVT entry.
30	TVT ECC Uncorrectable Error	RWH	0	This error will be injected for the next valid token after this injection bit is asserted. The error is injected when the TVT entry is written. The error is detected when the TVT entry is read. Example steps: 1) Set this injection bit. 2) Write to a TVT entry (the data will have bad ECC). 3) Issue a DMA that will target that TVT entry.
31:63	Reserved	RO	0	Reserved.

4.4.4.35 RXE_ARB Error LEM Report Enable Register

This register enables an inbound error to report that error through the PHB4 LEM structure. The bits of this register match bit for bit to those defined in the RXE_ARB Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single LEM interrupt.

Mnemonic RXE_ARB_E_LEM_RER

Address Offset 0x0D98

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWH	0	Bits correspond to the error that will be driven to its specific LEM bit.

4.4.4.36 RXE_ARB Error System Interrupt Enable Register

This register enables a PHB4 error to report that error through the system interrupt structure. The bits of this register match bit for bit to those defined in the INA Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single PHB4 system interrupt.

Mnemonic RXE_ARB_E_SYS_IER

Address Offset 0x0DA0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	RXE_ARB Error Status Mask Register	RWH	0	Bits correspond to the errors that will send a PHB4 sourced LSI error interrupt.

4.4.4.37 RXE_ARB Error EEH Freeze Enable Register

This register enables an error to place the PHB4 or an endpoint in a EEH freeze state. The bits of this register match bit for bit to those defined in the RXE_ARB Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single freeze signal to the error logic.

Mnemonic RXE_ARB_E_EEH_FER

Address Offset 0x0DA8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWH	0	Bits correspond to the errors that will set the EEH freeze state. These errors set EEH freeze/error bits for all endpoints or a specific end-point depending on the error.

4.4.4.38 RXE_ARB Error AIB Fence Enable Register

This register enables an error to place the PHB4 in a AIB interface fenced condition. The bits of this register match bit for bit to those defined in the RXE_ARB Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single AIB fence signal to the fence logic.

Mnemonic RXE_ARB_E_AIB_FER

Address Offset 0x0DB0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWH	0	Bits correspond to the errors that will fence the AIB bus.

4.4.4.39 RXE_ARB Error Log Register 0

This register contains log information for the RXE_ARB errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_ARB_ELR_0

Address Offset 0x0DC8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWHS	0	Some error bits do not have log information. The log contents will be all zeros for those errors.

4.4.4.40 RXE_ARB Error Log Register 1

This register contains additional log information for the RXE_ARB errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_ARB_ELR_1

Address Offset 0x0DC8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWHS	0	Some error bits do not have log information. The log contents will be all zeros for those errors.

4.4.4.41 RXE_ARB Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic RXE_ARB_ESMR

Address Offset 0x0DD0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWHS	0	Each mask bit corresponds to an error bit in the Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

4.4.4.42 RXE_ARB First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic RXE_ARB_FESMR

Address Offset 0x0DD8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, Page 145.	RWHS	0	Each mask bit corresponds to an error bit in the First Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error. In addition to masking the status register, this mask will prevent an error from capturing logging information in the error log registers.

4.4.4.43 RXE_MRG Error Status Register

This register contains error status information for the inbound directed errors.

Mnemonic RXE_MRG_ESR

Address Offset 0x0E00

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
00:07	Reserved	RO	0	Reserved.	Fatal
8	MRG TMB Allocation Error	RW1CHS	0	A token attempted to overwrite and allocate over a valid TMB entry.	Fatal
9	MRG TMB Response Invalid Error	RW1CHS	0	A TCE response was received for an invalid TMB entry.	Fatal
10	MRG TMB Response Ready Error	RW1CHS	0	An unsolicited TCE response was received for a TMB entry that was not waiting for a response.	Fatal
11	MRG MMIO Queue Overflow Error	RW1CHS	0	An MMIO related queue overflowed.	Fatal
12	MRG MMIO Queue Under-flow Error	RW1CHS	0	An MMIO related queue underflowed.	Fatal
13	MRG MMIO Internal Parity Error	RW1CHS	0	An MMIO internal resource had a parity error.	Fatal
14	MRG DMA Queue Overflow Error	RW1CHS	0	A DMA related queue overflowed.	Fatal
15	MRG DMA Queue Underflow Error	RW1CHS	0	A DMA related queue underflowed.	Fatal
16	MRG DMA Internal Parity Error	RW1CHS	0	A DMA internal resource had a parity error.	Fatal
17	MRG Migration Register Table Target Page Less Than Source Page Error	RW1CHS	0	The referenced Migration Register Table entry for a memory migrate operation contained a target page size less than the source page size of the original operation.	ER (SINGLE)
18	MRG Migration Register Table Valid Entry Error	RW1CHS	0	The referenced Migration Register Table entry for a memory migrate operation has the “Valid” bit (bit[0]) cleared to zero, signaling an invalid/uninitiated entry.	ER (SINGLE)
19	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
20	s_bad_addr_e_q	RW1CHS	0	RXE_MRG local error, internal register bad address error.	Fatal
21	s_req_size_align_e_q	RW1CHS	0	RXE_MRG local error, internal register read / write access to a register had incorrect size or address alignment.	Fatal
22	RRB Parity Error	RW1CHS	0	An internal parity error was detected in the RRB logic.	Fatal
23	RRB request Size/Alignment Error	RW1CHS	0	A request to the Remote Register Block (RRB) for this macro had bad size or address alignment.	INF
24	DSP AIB TX Timeout Error	RW1CHS	0	A pending transaction timeout expired waiting for AIB credits.	Fatal
25	Reserved (vA4.1) Discontiguous Byte Enable Write and Page Migration Error (vA4.2)	RW1CHS	0	PHB4 received a PCIe Discontiguous Byte Enable Writes to an IO page that is enabled for page migration in its TCE entry.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
26	DSP AIB TX CMD Credit Parity Error	RW1CHS	0	Parity Error on the AIB TX command credit return bus.	Fatal
27	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
28	DSP AIB TX DAT Credit Parity Error	RW1CHS	0	Parity Error on the AIB TX data credit return bus.	Fatal
29	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
30	DSP Command Credit Overflow Error	RW1CHS	0	The internal command credit counters overflowed.	Fatal
31	DSP Command Credit Underflow Error	RW1CHS	0	The internal command credit counters underflowed.	Fatal
32	DSP Command Credit Parity Error	RW1CHS	0	The internal command credit counters had a parity error.	Fatal
33	DSP Data Credit Overflow Error	RW1CHS	0	The internal data credit counters overflowed.	Fatal
34	DSP Data Credit Underflow Error	RW1CHS	0	The internal data credit counters underflowed.	Fatal
35	DSP Data Credit Parity Error	RW1CHS	0	The internal data credit counters had a parity error.	Fatal
36	DSP Completion State Machine One-Hot Error	RW1CHS	0	The state machine was found in multiple/invalid states.	Fatal
37	DSP Write Thread State Machine One-Hot Error	RW1CHS	0	The state machine was found in multiple/invalid states.	Fatal
38	DSP DMA Secure Address Error (vA4.2)	RW1CHS	0	This is the 'fail-safe' check for a Secure Address violation in a DMA read or write request before it is issued to the AIB bus. Refer to the *UV - Secure Address Exclude CMP/MSK Register* on page 295.	Fatal
39	DSP MSI Interrupt Notification Secure Address Error (vA4.2)	RW1CHS	0	This error checks that the interrupt notification address is within the secure range when the secure check is enabled. If the address is not in the secure range it will flag this error. Refer to the interrupt notification registers on page 110. Refer to the secure control registers on page 295.	Fatal
40	DSP TREQ ECC Correctable Error	RW1CHS	0	ECC correctable error detected in the TREQ queue.	INF
41	DSP TREQ ECC Uncorrectable Error	RW1CHS	0	ECC uncorrectable error detected in the TREQ queue.	Fatal
42	DSP MMIO Queue Overflow Error	RW1CHS	0	An MMIO related queue overflowed.	Fatal
43	DSP MMIO Queue Underflow Error	RW1CHS	0	An MMIO related queue underflowed.	Fatal
44	DSP MMIO Internal Parity Error	RW1CHS	0	An MMIO internal resource had a parity error.	Fatal
45	DSP DMA Queue Overflow Error	RW1CHS	0	A DMA related queue overflowed.	Fatal
46	DSP DMA Queue Underflow Error	RW1CHS	0	A DMA related queue underflowed.	Fatal
47	DSP DMA Internal Parity Error	RW1CHS	0	A DMA internal resource had a parity error.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
48	DSP Read Thread State Machine One-Hot Error	RW1CHS	0	The state machine was found in multiple/invalid states.	Fatal
49	DSP Table State Machine One-Hot Error	RW1CHS	0	The state machine was found in multiple/invalid states.	Fatal
50	DSP NBW State Machine One-Hot Error	RW1CHS	0	The state machine was found in multiple/invalid states. Active in x16 builds only.	Fatal
51	DSP TSM PEST BAR Disabled Error	RW1CHS	0	The error reporting logic attempted to access the PEST BAR when it was disabled.	Fatal
52:55	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
56	IPD ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the array.	INF
57	IPD ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in array.	Fatal
58	ICPLD ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the array.	INF
59	ICPLD ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in array.	Fatal
60	NBWD ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the Non-Blocking Write data array. Active in x16 builds only.	INF
61	NBWD ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in the Non-Blocking Write data array. Active in x16 builds only	Fatal
62	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
63	pb_etu_ai_rx_raise_fence (rising edge)	RW1CHS	0	PB logic above the PHB4 asserted its raise fence signal. This error is also routed to the LEM FIR. This can be used for debug in conjunction with the LEM WOF to test if the PB logic asserted its fence before other PHB4 errors. The PHB4 logic detects the error as a rising edge of the PB signal. The error will not reassert if the PB signal remains constant.	Fatal

4.4.4.44 RXE_MRG First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the RXE_MRG Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_MRG_FESR

Address Offset 0x0E08

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWHS	0	Bits correspond to the error that occurred first.

4.4.4.45 RXE_MRG Error Injection Register

This register is used for error injection. Setting the bit in this register will cause the selected error to occur by forcing the error condition. Writing bits to a ‘1’ in this register will schedule an inject of the error corresponding to the bit that is written. The bit(s) will automatically clear to ‘0’ as each error is injected. The corresponding bit in the Error Status Register is set automatically when the error inject occurs. The individual injection bit will clear to '0' after the hardware performs the injection action for that bit.

Mnemonic RXE_MRG_EIR

Address Offset 0x0E10

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWHS	0	Bits correspond to the error that can be injected.

4.4.4.46 RXE_MRG Error LEM Report Enable Register

This register enables a RXE_MRG Error to report that error through the PHB4 LEM structure. The bits of this register match bit for bit to those defined in the RXE_MRG Error Status Register. Setting a bit to ‘1’ will enable that error. All the enabled errors are ORed together to create a single interrupt.

Mnemonic RXE_MRG_LEM_RER

Address Offset 0x0E18

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Bits correspond to the error that will be driven to its specific LEM bit.

4.4.4.47 RXE_MRG Error System Interrupt Enable Register

This register enables a PHB4 error to report that error through the system interrupt structure. The bits of this register match bit for bit to those defined in the RXE_MRG Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single PHB4 system interrupt.

Mnemonic RXE_MRG_SYS_IER

Address Offset 0x0E20

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Bits correspond to the errors that will send a PHB4 sourced LSI error interrupt.

4.4.4.48 RXE_MRG Error EEH Freeze Enable Register

This register enables a RXE_MRG Error to place the PHB4 or an endpoint in a EEH freeze state. The bits of this register match bit for bit to those defined in the RXE_MRG Error Status Register. Setting a bit to '1' will enable that error to set the freeze state.

Mnemonic RXE_MRG_EEH_FER

Address Offset 0x0E28

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Bits correspond to the errors that will set the EEH freeze state. These errors set EEH freeze/error bits for all endpoints or a specific endpoint depending on the error.

4.4.4.49 RXE_MRG Error AIB Fence Enable Register

This register enables a RXE_MRG Error to place the PHB4 in a AIB interface Fenced condition. The bits of this register match bit for bit to those defined in the RXE_MRG Error Status Register. Setting a bit to '1' will enable that error. All the enables are ORed together to create a single fence signal to the fence logic.

Mnemonic RXE_MRG_AIB_FER

Address Offset 0x0E30

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Bits correspond to the errors that will fence the AIB bus.

4.4.4.50 RXE_MRG Error Log Register 0

This register contains log information for the RXE_MRG Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_MRG_LOG0

Address Offset 0x0E40

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information.	RWH	0	See the individual Error Status Register error bits for Log register bit assignments.

4.4.4.51 RXE_MRG Error Log Register 1

This register contains additional error log information for RXE_MRG Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_MRG_LOG1

Address Offset 0x0E48

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information.	RWH	0	See the individual Error Status Register error bits for Log register bit assignments.

4.4.4.52 RXE_MRG Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

If a bit in the mask register is set to a ‘1’ it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

Mnemonic RXE_MRG_ESMR

Address Offset 0x0E50

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Each mask bit corresponds to an error bit in the Error Status Register.

4.4.4.53 RXE_MRG First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the first Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

If a bit in the First Error mask register is set to a '1' it will prevent the corresponding error from being logged in the First Error Status Register. It does not prevent the side effects of the error. In addition to masking the First Error Status Register, this mask will prevent an error from capturing logging information in the error log registers

Mnemonic RXE_MRG_FESMR

Address Offset 0x0E58

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 153.	RWH	0	Each mask bit corresponds to an error bit in the First Error Status Register.

4.4.4.54 RXE_TCE Error Status Register

This register contains error status information for the inbound directed errors.

Mnemonic RXE_TCE_ESR

Address Offset 0x0E80

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	TCE CMP Internal Parity Error	RW1CHS	0	An internal parity error was detected in the TCE logic.	Fatal
1	TCE Request Page Access Error	RW1CHS	0	IODA Error: An access type conflicted with the allowed permissions in the TCE translation during the TCE request phase (on TCE cache lookup).	ER (SINGLE)
2	TCE Response Page Access Error	RW1CHS	0	IODA Error: An access type conflicted with the allowed permissions in the TCE translation during the TCE response phase.	ER (SINGLE)
3	TCE CMP Queue Overflow	RW1CHS	0	An overflow condition occurred in one of the TCE queues.	Fatal
4	TCE CMP Queue Underflow	RW1CHS	0	An underflow condition occurred in one of the TCE queues.	Fatal
5	TCE Secure Address Error	RW1CHS	0	This is the 'fail-safe' secure address range check for TCE requests that can be issued to the AIB bus. The logic will not allow a TCE request to be issued that matches a secure address space. This fail-safe covers the multi-level TCE table cases and all cases where HV firmware might have pre-loaded the caches, and so forth.	Fatal
6	TCE Cache Bad State Error	RW1CHS	0	An illegal state was reached in the TCE Cache.	Fatal

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
7	TCE Cache Multi-Way Hit Error	RW1CHS	0	A TCE Address look up in the TCE cache hit against multiple way entries.	Fatal
8	TCE Request Timeout Error	RW1CHS	0	A TCE request did not receive its TCE response before the time-out period.	Fatal
9	TCE TCR ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the TCR array.	INF
10	TCE TCR ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in the TCR array.	Fatal
11	TCE TDR ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the TDR array.	INF
12	TCE TDR ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in the TDR array.	Fatal
13	TCE Unexpected Response Error	RW1CHS	0	A TCE response was received with a TCE tag that did not match any outstanding TCE requests.	Fatal
14	RRB Parity Error	RW1CHS	0	An internal parity error was detected in the RRB logic.	Fatal
15	RRB request Size/Alignment Error	RW1CHS	0	A request to the Remote Register Block (RRB) for this macro had bad size or address alignment.	INF
16	TCE RES Internal Parity Error	RW1CHS	0	An internal parity error was detected in the TCE logic.	Fatal
17	s_bad_addr_e_q	RW1CHS	0	TCE local error, internal register bad address error.	Fatal
18	s_req_size_align_e_q	RW1CHS	0	TCE local error, internal register read / write access to a register had incorrect size or address alignment.	Fatal
19	TCE RES Queue Overflow	RW1CHS	0	An overflow condition occurred in one of the TCE queues.	Fatal
20	TCE RES Queue Underflow	RW1CHS	0	An underflow condition occurred in one of the TCE queues.	Fatal
21	TCE Response Data Parity Error	RW1CHS	0	The TCE data from the TXE macro interface had a parity error.	Fatal
22	TCE TCLB CAM Bad State Error	RW1CHS	0	An illegal state was reached in the TCLB.	Fatal
23	TCE TCLB CAM Multi-Hit Error	RW1CHS	0	A TCE Address look up in the TCLB hit against multiple entries.	Fatal
24	TCE Kill Internal Parity Error	RW1CHS	0	An internal parity error was detected in the TCE logic.	Fatal
25	TCE THASH Array ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the THASH array.	INF
26	TCE THASH Array ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in the THASH array.	Fatal
27	TCE TCLB TDAT ECC Correctable Error	RW1CHS	0	A correctable ECC error was detected in the TCLB array.	INF

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
28	TCE TCLB TDAT ECC Uncorrectable Error	RW1CHS	0	An uncorrectable ECC error was detected in the TCLB array.	Fatal
29	TCE Kill State Machine One-Hot Error	RW1CHS	0	The internal TCE kill state machine signaled a one-hot state error.	Fatal
30	TCE Kill Queue Overflow	RW1CHS	0	An overflow condition occurred in one of the TCE queues.	Fatal
31	TCE Kill Queue Underflow	RW1CHS	0	An underflow condition occurred in one of the TCE queues.	Fatal
32	TCE Request Secure Address Register	RW1CHS	0	Assert an error if a new TCE request from the DEC/TVT block if the fetch address matches secure range.	ER (SINGLE)
33	TCE Response Secure Address Register	RW1CHS	0	Error check of RPN of TCE response for DMAs. This is a check of the secure range of the RPN bits.	ER (SINGLE)
34:35	Reserved	RW1CHS	0	Reserved, but implemented for future use.	Fatal
36:63	Reserved	RO	0	Reserved.	

4.4.4.55 RXE_TCE First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the RXE_TCE Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_TCE_FESR

Address Offset 0x0E88

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWH	0	Bits correspond to the error that occurred first

4.4.4.56 RXE_TCE Error Injection Register

This register is used for error injection. Setting the bit in this register will cause the selected error to occur by forcing the error condition. Writing bits to a ‘1’ in this register will inject the error corresponding to the bit that is written. The bits will automatically clear to ‘0’ after the error is injected. The corresponding bit in the Error Status Register is set automatically when the error occurs. The individual injection bit will clear to '0' after the hardware performs the injection action for that bit.

Mnemonic RXE_TCE_EIR

Address Offset 0x0E90

Bit	Field Mnemonic	Type	Reset Value	Description
00:08	Reserved	RO	0	Reserved.
9	TCE TCR ECC Correctable Error	RWH	0	Bit corresponds to the error that can be injected.
10	TCE TCR ECC Uncorrectable Error	RWH	0	Bit corresponds to the error that can be injected.
11	TCE TDR ECC Correctable Error	RWH	0	Bit corresponds to the error that can be injected.
12	TCE TDR ECC Uncorrectable Error	RWH	0	Bit corresponds to the error that can be injected.
13:24	Reserved	RO	0	Reserved.
25	TCE THASH Array ECC Correctable Error	RWH	0	Bit corresponds to the error that can be injected.
26	TCE THASH Array ECC Uncorrectable Error	RWH	0	Bit corresponds to the error that can be injected.
27	TCE TCLB TDAT ECC Correctable Error	RWH	0	Bit corresponds to the error that can be injected.
28	TCE TCLB TDAT ECC Uncorrectable Error	RWH	0	Bit corresponds to the error that can be injected.
29:63	Reserved	RO	0	Reserved.

4.4.4.57 RXE_TCE Error LEM Report Enable Register

This register enables an inbound error to report that error through the PHB4 LEM structure. The bits of this register match bit for bit to those defined in the RXE_TCE Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single LEM interrupt.

Mnemonic RXE_TCE_E_LEM_RER

Address Offset 0x0E98

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWH	0	Bits correspond to the error that occurred first.

4.4.4.58 RXE_TCE Error System Interrupt Enable Register

This register enables a PHB4 error to report that error through the system interrupt structure. The bits of this register match bit for bit to those defined in the INA Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single PHB4 system interrupt.

Mnemonic RXE_TCE_E_SYS_IER

Address Offset 0x0EA0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWH	0	Bits correspond to the errors that will send a PHB4 sourced LSI error interrupt.

4.4.4.59 RXE_TCE Error EEH Freeze Enable Register

This register enables an error to place the PHB4 or an endpoint in a EEH freeze state. The bits of this register match bit for bit to those defined in the RXE_TCE Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single freeze signal to the error logic.

Mnemonic RXE_TCE_E_EEH_FER

Address Offset 0x0EA8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWH	0	Bits correspond to the errors that will set the EEH freeze state. These errors set EEH freeze/error bits for all endpoints or a specific end-point depending on the error.

4.4.4.60 RXE_TCE Error AIB Fence Enable Register

This register enables an error to place the PHB4 in a AIB interface Fenced condition. The bits of this register match bit for bit to those defined in the RXE_TCE Error Status Register. Setting a bit to '1' will enable that error. All the enabled errors are ORed together to create a single AIB Fence signal to the fence logic.

Mnemonic RXE_TCE_E_AIB_FER

Address Offset 0x0EB0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWH	0	Bits correspond to the errors that will fence the AIB bus.

4.4.4.61 RXE_TCE Error Log Register 0

This register contains log information for the RXE_TCE Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_TCE_ELR_0

Address Offset 0x0EC0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWHS	0	Some error bits do not have log information. The log contents will be all zeros for those errors.

4.4.4.62 RXE_TCE Error Log Register 1

This register contains additional error log information for RXE_TCE Errors defined. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic RXE_TCE_ELR_1

Address Offset 0x0EC8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log Information	RWHS	0	Some error bits do not have log information. The log contents will be all zeros for those errors.

4.4.4.63 RXE_TCE Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic RXE_TCE_ESMR

Address Offset 0x0ED0

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWHS	0	Each mask bit corresponds to an error bit in the Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

4.4.4.64 RXE_TCE First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic RXE_TCE_FESMR

Address Offset 0x0ED8

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See Error Status Register Bits, page 159.	RWHS	0	Each mask bit corresponds to an error bit in the First Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error. In addition to masking the status register, this mask will prevent an error from capturing logging information in the error log registers.

4.4.5 Debug Registers

This section describes the debug registers.

4.4.5.1 PHB4 - Trace Control Register

This register controls the traced signals in the PHB4/ETU. The traced signals are used for PHB4 debug purposes.

Mnemonic PHB_TCR

Address Offset 0x0F80

Bit	Field Mnemonic	Type	Reset Value	Description
0	regs_bclk_trc_enable	RW	0	Trace Enable bit. Set to '1' to enable tracing from PHB4. This signal also acts as a clock gating signal, set to '0' when not tracing to save power.
01:03	Reserved	RW	0	Reserved, but implemented register(s) for future use.
04:07	regs_bclk_trc_sel(0:3)	RW	0	4 bit field used to select 1 of 16 possible groups to place on the trace mux output po_phb_trc_vec(0:87).
04:63	Reserved	RO	0	Reserved.

Table 4-25. PHB4 Trace Group

Group	Name	Description	Page
0	unused	unused group, all zeros trace output	
1	AIB RX	AIB Bus, receive bus / PHB4 input	167
2	AIB TX0	AIB Bus, transmit bus / PHB4 output (Format 0)	167
3	AIB TX1	AIB Bus, transmit bus / PHB4 output (Format 1)	167
4	BLIF RX0	BLIF RX TLP Header (Format 0)	168
5	BLIF RX1	BLIF RX TLP Header (Format 1)	168
6	BLIF TX0	BLIF TX TLP Header (Format 0)	168
7	BLIF TX1	BLIF TX TLP Header (Format 1)	169
8 to 14	unused	unused group, all zeros trace output	
15	pcie_etu_trc_vec(0:87)	PCIe/REGB Trace Group	281

Table 4-26. Trace Group AIX RX

Byte	Bit	Facility	Comment
0	0	ai_rx_cmd_val	
0	001:007	ai_rx_cmd_type(1:7)	
1	008:015	ai_rx_cmd_size(4:11)	Least significant 8 bits of size field.
2	016:023	ai_rx_cmd_tag(0:7)	
3 to 9	024:079	ai_tx_cmd_addr(8:63)	
10	080:087	ai_tx_cmd_attr(0:7)	AIB command/attribute bits.

Table 4-27. Trace Group AIB TX0

Byte	Bit	Facility	Comment
0	0	ai_tx_cmd_val	
0	001:007	ai_tx_cmd_type(1:7)	
1	008:015	ai_tx_cmd_size(0:7)	
2	016:019	ai_tx_cmd_size(8:11)	
2	020:023	ai_tx_cmd_tag(0:3)	
3	024:027	ai_tx_cmd_tag(4:7)	
3	028:031	ai_tx_cmd_addr(8:11)	
4 to 9	032:079	ai_tx_cmd_addr(12:59)	
10	080:083	ai_tx_cmd_addr(60:63)	
10	084:087	ai_tx_cmd_attr(16:19)	AIB command/attribute bits.

Table 4-28. Trace Group AIB TX1

Byte	Bit	Facility	Comment
0	0	ai_tx_cmd_val	
0	001:007	ai_tx_cmd_type(1:7)	
1	008:015	ai_tx_cmd_size(0:7)	
2	016:019	ai_tx_cmd_size(8:11)	
2	020:023	ai_tx_cmd_tag(0:3)	
3	024:027	ai_tx_cmd_tag(4:7)	
3	028:031	ai_tx_cmd_addr(8:11)	
4 to 6	032:055	ai_tx_cmd_addr(12:35)	
7	056:059	ai_tx_cmd_addr(36:39)	
7	060:063	ai_tx_cmd_attr(4:7)	AIB command/attribute bits.
8 to 10	064:087	ai_tx_cmd_attr(8:31)	AIB command/attribute bits.

Table 4-29. Trace Group BLIF RX0

Bit	Facility	Comment
0	blif_trace_rxhdr_val	
001:007	blif_trace_rxhdr(001 to 007)	Byte 0 (First reserved bit replaced with tlif_trace_rxhdr_val).
008:015	blif_trace_rxhdr(016 to 023)	Byte 2 (Skip byte 1, it only has traffic class info in it).
016:023	blif_trace_rxhdr(024 to 031)	Byte 3.
024:031	blif_trace_rxhdr(032 to 039)	Byte 4.
032:039	blif_trace_rxhdr(040 to 047)	Byte 5.
040:047	blif_trace_rxhdr(048 to 055)	Byte 6.
048:055	blif_trace_rxhdr(056 to 063)	Byte 7.
056:063	blif_trace_rxhdr(064 to 071)	Byte 8.
064:071	blif_trace_rxhdr(072 to 079)	Byte 9.
072:079	blif_trace_rxhdr(080 to 087)	Byte 10.
080:087	blif_trace_rxhdr(088 to 095)	Byte 11.

Table 4-30. Trace Group BLIF RX1

Bit	Facility	Comment
0	blif_trace_rxhdr_val	
001:007	blif_trace_rxhdr(001 to 007)	Byte 0 (First reserved bit replaced with tlif_trace_rxhdr_val).
008:015	blif_trace_rxhdr(016 to 023)	Byte 2 (Skip byte 1, it only has traffic class info in it).
016:023	blif_trace_rxhdr(024 to 031)	Byte 3.
024:031	blif_trace_rxhdr(032 to 039)	Byte 4.
032:039	blif_trace_rxhdr(040 to 047)	Byte 5.
040:047	blif_trace_rxhdr(048 to 055)	Byte 6.
048:055	blif_trace_rxhdr(056 to 063)	Byte 7.
056:063	blif_trace_rxhdr(096 to 103)	Byte 12.
064:071	blif_trace_rxhdr(104 to 111)	Byte 13.
072:079	blif_trace_rxhdr(104 to 111)	Byte 14.
080:087	blif_trace_rxhdr(120 to 127)	Byte 15.

Table 4-31. Trace Group BLIF TX0

Bit	Facility	Comment
0	blif_trace_txhdr_val	
001:007	blif_trace_txhdr(001 to 007)	Byte 0 (First reserved bit replaced with tlif_trace_rxhdr_val).
008:015	blif_trace_txhdr(016 to 023)	Byte 2 (Skip byte 1, it only has traffic class info in it).
016:023	blif_trace_txhdr(024 to 031)	Byte 3.
024:031	blif_trace_txhdr(032 to 039)	Byte 4.

Table 4-31. Trace Group BLIF TX0

Bit	Facility	Comment
032:039	blif_trace_txhdr(040 to 047)	Byte 5.
040:047	blif_trace_txhdr(048 to 055)	Byte 6.
048:055	blif_trace_txhdr(056 to 063)	Byte 7.
056:063	blif_trace_txhdr(064 to 071)	Byte 8.
064:071	blif_trace_txhdr(072 to 079)	Byte 9.
072:079	blif_trace_txhdr(080 to 087)	Byte 10.
080:087	blif_trace_txhdr(088 to 095)	Byte 11.

Table 4-32. Trace Group BLIF TX1

Bit	Facility	Comment
0	blif_trace_txhdr_val	
001:007	blif_trace_txhdr(001 to 007)	Byte 0 (First reserved bit replaced with tlif_trace_rxhdr_val).
008:015	blif_trace_txhdr(016 to 023)	Byte 2 (Skip byte 1, it only has traffic class info in it).
016:023	blif_trace_txhdr(024 to 031)	Byte 3.
024:031	blif_trace_txhdr(032 to 039)	Byte 4.
032:039	blif_trace_txhdr(040 to 047)	Byte 5.
040:047	blif_trace_txhdr(048 to 055)	Byte 6.
048:055	blif_trace_txhdr(056 to 063)	Byte 7.
056:063	blif_trace_txhdr(096 to 103)	Byte 12.
064:071	blif_trace_txhdr(104 to 111)	Byte 13.
072:079	blif_trace_txhdr(104 to 111)	Byte 14.
080:087	blif_trace_txhdr(120 to 127)	Byte 15.

4.4.5.2 Performance Monitor Configuration Register

This register controls what is logged in the four performance monitor counter (PMC) registers.

Note: The leading/trailing edge enables are the same when both are '00' or both are '11'. That is, they will ignore edges. They only honor edges when only one is active at a time. Also, in the case when they are the same they will count the event every cycle it is active '1'.

Mnemonic PMCR

Address Offset 0x0F88

Bit	Field Mnemonic	Type	Reset Value	Description
0	Counter Enable	RW	0	When set to ‘1’ the PMCs are incremented each time a selected event occurs. When set to ‘0’ the PMCs are never incremented.

Bit	Field Mnemonic	Type	Reset Value	Description
01:03	Reserved	RW	0	Unused, implemented bits.
4	Stop Correlation Enable	RW	0	When set to ‘1’ the PMCs will be correlated. Correlation means that when any of the counters most significant bit is a one then all counters are stopped. When set to ‘0’ none of the counters will stop counting.
5	Reserved	RW	1	Unused, implemented bits was the performance monitor interrupt disable bit
6	Counter 0 Leading Edge Enable	RW	0	When set to ‘1’ the PMC0 will only count the rising edge of events rather than every cycle an event is active. When set to ‘0’ the PMC0 will count the event every cycle it is active.
7	Counter 1 Leading Edge Enable	RW	0	When set to ‘1’ the PMC1 will only count the rising edge of events rather than every cycle an event is active. When set to ‘0’ the PMC1 will count the event every cycle it is active.
8	Counter 2 Leading Edge Enable	RW	0	When set to ‘1’ the PMC2 will only count the rising edge of events rather than every cycle an event is active. When set to ‘0’ the PMC2 will count the event every cycle it is active.
9	Counter 3 Leading Edge Enable	RW	0	When set to ‘1’ the PMC3 will only count the rising edge of events rather than every cycle an event is active. When set to ‘0’ the PMC3 will count the event every cycle it is active.
10	Counter 0 Trailing Edge Enable	RW	0	When set to ‘1’ the PMC0 will only count the falling edge of events rather than every cycle an event is active. When set to ‘0’ the PMC0 will count the event every cycle it is active.
11	Counter 1 Trailing Edge Enable	RW	0	When set to ‘1’ the PMC1 will only count the falling edge of events rather than every cycle an event is active. When set to ‘0’ the PMC1 will count the event every cycle it is active.
12	Counter 2 Trailing Edge Enable	RW	0	When set to ‘1’ the PMC2 will only count the falling edge of events rather than every cycle an event is active. When set to ‘0’ the PMC2 will count the event every cycle it is active.
13	Counter 3 Trailing Edge Enable	RW	0	When set to ‘1’ the PMC3 will only count the falling edge of events rather than every cycle an event is active. When set to ‘0’ the PMC3 will count the event every cycle it is active.
14:17	Reserved	RW	0	Unused, implemented bits.
18:23	Counter 0 Event Select(0:5)	RW	0	This is the event selection value for PMC0. This selects the event to be counted by the counter register. See *Table 4-33* on page 171 for event selection encoding values.
24:25	Reserved	RW	0	Unused, implemented bits.
26:31	Counter 1 Event Select(0:5)	RW	0	This is the event selection value for PMC1. This selects the event to be counted by the counter register. See *Table 4-33* on page 171 for event selection encoding values.
32:33	Reserved	RW	0	Unused, implemented bits.
34:39	Counter 2 Event Select(0:5)	RW	0	This is the event selection value for PMC2. This selects the event to be counted by the counter register. See *Table 4-33* on page 171 for event selection encoding values.
40:41	Reserved	RW	0	Unused, implemented bits.

Bit	Field Mnemonic	Type	Reset Value	Description
42:47	Counter 3 Event Select(0:5)	RW	0	This is the event selection value for PMC3. This selects the event to be counted by the counter register. See *Table 4-33* on page 171 for event selection encoding values.
48:63	Reserved	RO	0	Reserved.

Table 4-33. Performance Counter Selection Encodes

Encodes	Internal Signal Name	Description	Leading Edge Enable
0	'0'	Counter is disabled. No events are counted.	Don't care
1	'1'	Count PHB4 clock cycles.	0
2	txe_perf_events(00)	AIB Receive Command Valid.	0
3	txe_perf_events(01)	AIB Receive Data Valid.	0
4	txe_perf_events(02)	AIB Receive Command or Data Valid.	0
5	txe_perf_events(03)	AIB Read Command received (any address space).	0
6	txe_perf_events(04)	AIB Write Command received (any address space).	0
7	txe_perf_events(05)	Reserved (always zero).	0
8	txe_perf_events(06)	Reserved (always zero).	0
9	txe_perf_events(07)	Interrupt CI Load/Store Command received (any). Interrupt Trigger or EOI, and so forth.	0
10	txe_perf_events(08)	TCE Read Response Command received.	0
11	txe_perf_events(09)	DMA Read Response Command received. (PHB/ETU does not receive DMA Read Responses in this design, so this event is never used, always zero.)	0
12	txe_perf_events(10)	Internal Register Address Space Match (on AIB/MMIO command).	0
13	txe_perf_events(11)	M32 Address Space Match (on AIB/MMIO command).	0
14	txe_perf_events(12)	Reserved (always zero).	0
15	txe_perf_events(13)	M64 Address Space Match (on AIB/MMIO command).	0
16	txe_perf_events(14)	Peer-to-Peer CI Store for M64 MMIO Space Received.	0
17	txe_perf_events(15)	Peer-to-Peer CI Store for M32 MMIO Space Received.	0
18	txe_perf_events(16)	Reserved (always zero).	0
19	txe_perf_events(17)	Reserved (always zero).	0
20	rxe_perf_any_dma_rcv	DMA (any, read or write) received from the PCIe link. (Note this does not include speculation requests.)	0
21	rxe_perf_dmard_rcv	DMA Read received from the PCIe link. (Note this does not include speculation requests.)	0
22	rxe_perf_dmawr_rcv	DMA Write received from the PCIe link. (Note this does not include speculation requests.)	0
23	rxe_perf_lsi_int_pres	LSI interrupt internal represent.	0

Table 4-33. Performance Counter Selection Encodes

Encodes	Internal Signal Name	Description	Leading Edge Enable
24	rxe_perf_msi_int_rcv	MSI interrupt received from PCIe link. (Note this does not include speculation requests.)	0
25	rxe_perf_mmio_ldrsp_rcv	MMIO Load Response received from PCIe link. (Note this include ALL responses from the PCIe link including completions for PCI CFG reads and writes.)	0
26	rxe_perf_spec_pref_req	Speculation request received. This can be for a DMA read or DMA write.	0
27	rxe_perf_ipd_pop	IPD Inbound Posted Data Buffer popped (1 data beat moved).	0
28	rxe_perf_icpld_pop	ICPLD Inbound Completion Data Buffer popped (1 data beat moved).	0
29	rxe_perf_tce_miss_any	TCE Cache Miss (any).	0
30	rxe_perf_tce_miss_dmard	TCE Cache Miss on a DMA Read.	0
31	rxe_perf_tce_miss_dmawr	TCE Cache Miss on a DMA Write.	0
32	rxe_perf_tclb_miss_any	TCLB Miss (any).	0
33	rxe_perf_tclb_miss_dmard	TCLB Miss on a DMA Read.	0
34	rxe_perf_tclb_miss_dmawr	TCLB Miss on a DMA Write.	0
35	rxe_perf_rtc_miss_any	RTC Miss (any).	0
36	rxe_perf_rtc_miss_dmard	RTC Miss on a DMA Read.	0
37	rxe_perf_rtc_miss_dmawr	RTC Miss on a DMA Write.	0
38 to 63	Reserved	Unused event encodes.	

4.4.5.3 Performance Monitor Counter Register 0

This register counts specific events and is referred to as PMC0. This register is controlled by the Performance Monitor Configuration Register and counts the events selected by that register.

The counter increments for its selected events until its MSB, bit 0, becomes a '1' at that time an interrupt is generated if it is enabled in the Performance Monitor Configuration Register. The counter continues to increment after the interrupt and eventually will roll-over back to all zeros.

The counter can be programmed to stop counting when its MSB, bit 0, becomes a '1' if the Stop Correlation Enable bit is set in the Performance Monitor Configuration Register.

The counter can be written to any initial value before it is enabled to achieve shorter intervals for interrupts and reaching MSB=1 for the counter. This is useful because, a 48-bit counter starting at a value of zero and incrementing every clock cycle will take 6.5 days to set the MSB.

Mnemonic PMCR_0

Address Offset 0x0F90

Bit	Field Mnemonic	Type	Reset Value	Description
00:47	Counter Value	RWH	0	This is the current count of events selected by the Performance Monitor Configuration Register.
48:63	Reserved	RO	0	Reserved.

4.4.5.4 Performance Monitor Counter Register 1

This register counts specific events and is referred to as PMC1. This register is controlled by the Performance Monitor Configuration Register and counts the events selected by that register

The counter increments for its selected events until its MSB, bit 0, becomes a '1' at which time an interrupt is generated if it is enabled in the Performance Monitor Configuration Register. The counter continues to increment after the interrupt and eventually will roll-over back to all zeros.

The counter can be programmed to stop counting when its MSB, bit 0, becomes a '1' if the Stop Correlation Enable bit is set in the Performance Monitor Configuration Register.

The counter can be written to any initial value before it is enabled to achieve shorter intervals for interrupts and reaching MSB=1 for the counter. This is useful because, as a 48-bit counter starting at a value of zero and incrementing every clock cycle will take 6.5 days to set the MSB.

Mnemonic PMCR_1

Address Offset 0x0F98

Bit	Field Mnemonic	Type	Reset Value	Description
00:47	Counter Value	RWH	0	This is the current count of events selected by the Performance Monitor Configuration Register.
48:63	Reserved	RO	0	Reserved.

4.4.5.5 Performance Monitor Counter Register 2

This register counts specific events and is referred to as PMC2. This register is controlled by the Performance Monitor Configuration Register and counts the events selected by that register.

The counter increments for its selected events until its MSB, bit 0, becomes a ‘1’ at which time an interrupt is generated if it is enabled in the Performance Monitor Configuration Register. The counter continues to increment after the interrupt and eventually will roll-over back to all zeros.

The counter can be programmed to stop counting when its MSB, bit 0, becomes a ‘1’ if the Stop Correlation Enable bit is set in the Performance Monitor Configuration Register.

The counter can be written to any initial value before it is enabled to achieve shorter intervals for interrupts and reaching MSB=1 for the counter. This is useful because, as a 48-bit counter starting at a value of zero and incrementing every clock cycle will take 6.5 days to set the MSB.

Mnemonic PMCR_2

Address Offset 0x0FA0

Bit	Field Mnemonic	Type	Reset Value	Description
00:47	Counter Value	RWH	0	This is the current count of events selected by the Performance Monitor Configuration Register.
48:63	Reserved	RO	0	Reserved.

4.4.5.6 Performance Monitor Counter Register 3

This register counts specific events and is referred to as PMC3. This register is controlled by the Performance Monitor Configuration Register and counts the events selected by that register.

The counter increments for its selected events until its MSB, bit 0, becomes a ‘1’ at which time an interrupt is generated if it is enabled in the Performance Monitor Configuration Register. The counter continues to increment after the interrupt and eventually will roll-over back to all zeros.

The counter can be programmed to stop counting when its MSB, bit 0, becomes a ‘1’ if the Stop Correlation Enable bit is set in the Performance Monitor Configuration Register.

The counter can be written to any initial value before it is enabled to achieve shorter intervals for interrupts and reaching MSB=1 for the counter. This is useful because, as a 48-bit counter starting at a value of zero and incrementing every clock cycle will take 6.5 days to set the MSB.

Mnemonic PMCR_3

Address Offset 0x0FA8

Bit	Field Mnemonic	Type	Reset Value	Description
00:47	Counter Value	RWH	0	This is the current count of events selected by the Performance Monitor Configuration Register.
48:63	Reserved	RO	0	Reserved.

4.5 PCI Configuration Space Registers

The PCI Configuration Space Registers are the PHB4's local Root Complex registers. These registers are architected in the PCI Local Base Specification and the current PCIe Architecture Specification for the Type-1, root complex space.

These registers are mapped directly into the chip address space. Refer to the *PHB4 Unified Address Space* on page 56. The PCI configuration space registers range from 0x1000 to 0x17FF. These registers can only be accessed via the PHB4 register address space and cannot be access via other means. They can be accessed only by the SCOM or AIB interfaces on the PHB4.

The local, direct mapped/root complex registers only allow 4-byte length and aligned access from SCOM or AIB/MMIO interfaces.

4.5.1 PCI Configuration Space Register Address Map

These registers are specific to a root complex only, PCI Header Type-1 configuration space.

Table 4-34. PCI Configuration Space Register Address Map (Page 1 of 4)

Offset	Description	Page
PCI - Header Type-1 Registers (0x1000 to 0x103C)		
0x1000	PCI - Vendor ID / Device ID Register	179
0x1004	PCI - Command / Status Register	180
0x1008	PCI - Revision ID / Class Code Register	182
0x100C	PCI - Cache Line Size, Master Latency, Header Type, BIST Register	182
0x1010	PCI - Base Address Register 0 (BAR0), not used in the Root Complex (returns all zeros when read, ACK)	
0x1014	PCI - Base Address Register 1 (BAR1), not used in the Root Complex (returns all zeros when read, ACK)	
0x1018	PCI - Primary / Secondary / Subordinate Bus Number Register (Type-1)	183
0x101C	PCI - IO Base/Limit and Secondary Status Register	183
0x1020	PCI - Memory Base/Limit Register	184
0x1024	PCI - Prefetch Base/Limit Register	184
0x1028	PCI - Prefetch Base Upper Register	185
0x102C	PCI - Prefetch Limit Upper Register	185
0x1030	PCI - IO Base/Limit Upper Register, not used (returns all zeros when read, ACK)	
0x1034	PCI - Capabilities Pointer Register	195
0x1038	PCI - Bridge Expansion ROM Base Address Register, not used (returns all zeros when read, ACK)	
0x103C	PCI - Bridge Control Register	187
PM - Power Management Registers (0x1040 to 0x1044)		
0x1040	PM - Capability ID / Next PTR / PMC Register	188

Table 4-34. PCI Configuration Space Register Address Map (Page 2 of 4)

Offset	Description	Page
0x1044	PM - PMCSR / PMCSR_BSE / Data Register	189
EC - PCI Express Capabilities Registers (0x1048 to 0x1078		
0x1048	EC - Capability ID / Next PTR / PCI Express Capabilities Register	190
0x104C	EC - Device Capabilities Register	191
0x1050	EC - Device Control / Status Register	208
0x1054	EC - Link Capabilities Register	195
0x1058	EC - Link Control / Status Register	196
0x105C	EC - Slot Capabilities Register	198
0x1060	EC - Slot Control / Status Register	201
0x1064	EC - Root Control Register	202
0x1068	EC - Root Status Register, not used (returns all zeros when read, ACK)	
0x106C	EC - Device Capabilities Register 2	205
0x1070	EC - Device Control / Status Register 2	205
0x1074	EC - Link Capabilities Register 2	206
0x1078	EC - Link Control / Status Register 2	208
0x107C	EC - Slot Capabilities 2 Register, not used (returns all zeros when read, ACK)	
0x1080	EC - Slot Control/Status 2 Register, not used (returns all zeros when read, ACK)	
Reserved Registers (0x1084 to 0x10FC)		
0x1084 to 0x10FC	Reserved and undefined (returns all ones when read, NAK)	
0x1084 to 0x10FC	Reserved and undefined (returns all ones when read, NAK)	
AER - Advanced Error Reporting Registers (0x1100 to 0x1144)		
0x1100	AER - PCI Express Enhanced Capability Header	213
0x1104	AER - Uncorrectable Error Status Register	213
0x1108	AER - Uncorrectable Error Mask Register	214
0x110C	AER - Uncorrectable Error Severity Register, not used (returns all zeros when read, ACK)	
0x1110	AER - Correctable Error Status Register	215
0x1114	AER - Correctable Error Mask Register, not used (returns all zeros when read, ACK)	
0x1118	AER - Capabilities and Control Register	216
0x111C	AER - Header Log Register #1	217
0x1120	AER - Header Log Register #2	217

Table 4-34. PCI Configuration Space Register Address Map (Page 3 of 4)

Offset	Description	Page
0x1124	AER - Header Log Register #3	218
0x1128	AER - Header Log Register #4	218
0x112C	AER - Root Error Control Register, not used (returns all zeros when read, ACK)	
0x1130	AER - Root Error Status Register	219
0x1134	AER - Error Source Identification Register	219
0x1138	AER - TLP Prefix Log Register #1, not used (returns all zeros when read, ACK)	
0x113C	AER - TLP Prefix Log Register #2, not used (returns all zeros when read, ACK)	
0x1140	AER - TLP Prefix Log Register #3, not used (returns all zeros when read, ACK)	
0x1144	AER - TLP Prefix Log Register #4, not used (returns all zeros when read, ACK)	
SEC - Secondary PCI Express Extended Capability Registers (0x1148 to 0x1170)		
0x1148	SEC - PCI Express Extended Capability Header Register	220
0x114C	SEC - Link Control 3 Register	220
0x1150	SEC - Lane Error Status Register, not used (returns all zeros when read, ACK)	
0x1154	SEC - Lane Equalization Control Register #1, not used (returns all zeros when read, ACK)	
0x1158	SEC - Lane Equalization Control Register #2, not used (returns all zeros when read, ACK)	
0x115C	SEC - Lane Equalization Control Register #3, not used (returns all zeros when read, ACK)	
0x1160	SEC - Lane Equalization Control Register #4, not used (returns all zeros when read, ACK)	
0x1164	SEC - Lane Equalization Control Register #5, not used (returns all zeros when read, ACK)	
0x1168	SEC - Lane Equalization Control Register #6, not used (returns all zeros when read, ACK)	
0x116C	SEC - Lane Equalization Control Register #7, not used (returns all zeros when read, ACK)	
0x1170	SEC - Lane Equalization Control Register #8, not used (returns all zeros when read, ACK)	
P16 – Physical Layer 16.0 GT/s Capability Registers (0x1174 to 0x119C) (vA4.2)		
0x1174	P16 – Physical Layer 16 GT/s Extended Capability Header Register	
0x1178	P16 – 16 GT/s Capabilities Register	221
0x117C	P16 – 16 GT/s Control Register	221
0x1180	P16 – 16 GT/s Status Register	222

Table 4-34. PCI Configuration Space Register Address Map (Page 4 of 4)

Offset	Description	Page
0x1184	P16 – 16 GT/s Local Data Parity Mismatch Status Register	223
0x1188	P16 – 16 GT/s First Retimer Data Parity Mismatch Status Register	223
0x118C	P16 – 16 GT/s Second Retimer Data Parity Mismatch Status Register	224
0x1190	P16 – 16 GT/s Lane Equalization Control Register # 1, not used (returns all zeros when read, ACK)	
0x1194	P16 – 16 GT/s Lane Equalization Control Register # 2, not used (returns all zeros when read, ACK)	
0x1198	P16 – 16 GT/s Lane Equalization Control Register # 3, not used (returns all zeros when read, ACK)	
0x119C	P16 – 16 GT/s Lane Equalization Control Register # 4, not used (returns all zeros when read, ACK)	
LMR – Lane Margining at the Receiver Capability Registers (0x11A0 to 0x11E4) (vA4.2)		
0x11A0	LMR – Margining Extended Capability Header Register	224
0x11A4	LMR – Margining Port Capabilities / Status Register	225
0x11A8	LMR – Margining Lane Control / Status Register # 1	225
0x11AC	LMR – Margining Lane Control / Status Register # 2	226
0x11B0	LMR – Margining Lane Control / Status Register # 3	227
0x11B4	LMR – Margining Lane Control / Status Register # 4	227
0x11B8	LMR – Margining Lane Control / Status Register # 5	228
0x11BC	LMR – Margining Lane Control / Status Register # 6	228
0x11C0	LMR – Margining Lane Control / Status Register # 7	229
0x11C4	LMR – Margining Lane Control / Status Register # 8	229
0x11C8	LMR – Margining Lane Control / Status Register # 9	230
0x11CC	LMR – Margining Lane Control / Status Register # 10	230
0x11D0	LMR – Margining Lane Control / Status Register # 11	231
0x11D4	LMR – Margining Lane Control / Status Register # 12	231
0x11D8	LMR – Margining Lane Control / Status Register # 13	232
0x11DC	LMR – Margining Lane Control / Status Register # 14	232
0x11E0	LMR – Margining Lane Control / Status Register # 15	233
0x11E4	LMR – Margining Lane Control / Status Register # 16	233
DLF – Data Link Feature Capability Registers (0x11E8 to 0x11F0) (vA4.2)		
0x11E8	DLF – Data Link Feature Extended Capability Header Register	234
0x11EC	DLF – Data Link Feature Capabilities Register	234
0x11F0	DLF – Data Link Feature Status Register	235
Reserved Registers (0x11F4 to 0x17FC)		
0x11F4 to 0x17FC	Reserved and undefined (returns all ones when read, NAK)	

4.5.2 PCI - Header Type-1 Registers (0x000 to 0x03C)

4.5.2.1 PCI - Vendor ID/Device ID Register

The PCI Vendor ID and Device ID registers are read-only registers, used to communicate device identification information to the software operating system. These registers are required for all PCI compliant devices.

Mnemonic

Address Offset 0x000

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	pi_sys_pci00_vendor_id(15:0)	RO	0x1014	Vendor ID Register [15:00]. IBM's PCI Vendor ID.
31:16	pi_sys_pci00_device_id(15:0)	RO	0x04C1	Device ID Register [15:00]. This value is dependent on the chip/ASIC where the root complex is used. Must drive the sys_pci00_device_id [15:00] signal with the correct value. POWER9 value: 0x04C1.

4.5.2.2 PCI - Command/Status Register

The PCI Command and Status registers are required for all PCI compliant devices and are used to communicate device status and control information between the hardware and software.

Mnemonic

Address Offset 0x004

Bit	Field Mnemonic	Type	Reset Value	Description
0	Reserved	RW	0	Command Register [0] - I/O Space Enable. IO space is no longer used. This bit is reserved as a placeholder, is writeable.
1	cfgrco_pci04_mem_enable	RW	0	Command Register [1] - Memory Space Enable. Controls the ability of the port to master Memory or I/O requests in the outbound direction. When this bit is not set, Memory and I/O requests are handled as Unsupported Requests (UR). When = 0: • Outbound Posted commands will be dropped, not forwarded to the link. • Outbound Nonposted commands are dropped, report error response to AIB. • Outbound Completion commands are forwarded to the link normally. • All other Outbound commands are unaffected by this bit. When = 1: • All outbound requests are forwarded to the link. NOTE: Message requests and configuration requests are not affected by this control, they are always forwarded to the link.
2	cfgrco_pci04_busmaster_enable	RW	0	Command Register [2] - Bus Master Enable. When enabled, the port is allowed to process and receive inbound Memory requests. When = 0: • Inbound Posted commands will be dropped, not forwarded to AIB. • Inbound Nonposted commands are dropped, UR completion issued back to requester. • Inbound Completion commands are forwarded to AIB normally. When = 1: • All inbound requests are processed normally.
05:03	Reserved	RO	0	Command Register [5:3] - Reserved.
6	Reserved	RO	0	Command Register [6] - Parity Error (PERR) Enable.
7	Reserved	RO	0	Command Register [7] - Reserved.
8	PCI SERR Enable Bit	RO	1	Command Register [8] - SERR Enable. Current implementation hard-wires this value to '1'.
9	Reserved	RO	0	Command Register [9] - Reserved.

Bit	Field Mnemonic	Type	Reset Value	Description
10	Reserved	RO	0	Command Register [10] - Interrupt Disable.
15:11	Reserved	RO	0	Command Register [15:11] - Reserved.
19:16	Reserved	RO	0	Status Register [3:0] - Reserved.
20	PCI Express Capabilities Structure Present Bit	RO	1	Status Register [4] - Capabilities List. Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, this bit must be hard wired to '1'.
23:21	Reserved	RO	0	Status Register [7:5] - Reserved.
24	Reserved	RO	0	Status Register [8] - Master Data Parity Error. For root ports, this bit is always zero.
26:25	Reserved	RO	0	Status Register [10:9] - Reserved.
27	Reserved	RO	0	Status Register [11] - Signaled Target Abort. For root ports, this bit is always zero.
28	Reserved	RO	0	Status Register [12] - Received Target Abort. For root ports, this bit is always zero.
29	Reserved	RO	0	Status Register [13] - Received Master Abort. For root ports, this bit is always zero.
30	Reserved	RO	0	Status Register [14] - Signaled System Error. For root ports, this bit is set if an ERROR_FATAL or ERROR_NONFATAL Message is received, and PCI3C[17] is Set to forward the error. Current implementation hard-wires this value to '0' and does not used this bit.
31	Reserved	RO	0	Status Register [15] - Detected Parity Error. For root ports, this bit is always zero.

4.5.2.3 PCI - Revision ID/Class Code Register

The PCI Revision ID and Class Code registers are read-only registers used to communicate additional device identification information to the software operating system. These registers are required for all PCI compliant devices.

Mnemonic

Address Offset 0x008

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	pi_sys_pci08_revision_id(7:0)	RO	0	Revision ID Register [07:00]. This value is dependent on the chip/ASIC where the root complex is used. Must drive the sys_pci08_revision_id [07:00] signal with the correct value.
31:08	pi_sys_pci08_class_code(23:0)	RO	0x060400	Class Code Register [23:00]. This value is dependent on the chip/ASIC where the root complex is used. Must drive the sys_pci08_class_code [23:00] signal with the correct value.

4.5.2.4 PCI - Cache Line Size, Master Latency, Header Type, BIST Register

PCI specification register. Most of this register contains legacy functions that do not apply to PCI Express.

Mnemonic

Address Offset 0x00C

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	PCI Legacy Bits	RO	0	Bits are hard-wired to zeros.
23:16	Header Type(7:0)	RO	0x01	Header Type-1 configuration space. Root Complexes are Type-1. Endpoints are Type-0.
31:24	PCI Legacy Bits	RO	0	Bits are hard-wired to zeros.

4.5.2.5 PCI - Primary/Secondary/Subordinate Bus Number Register

The Primary, Secondary, and Subordinate Bus Number Registers are required for all TYPE-1 devices. Software must set these bits to assign the correct PCI bus hierarchy relative to the root complex. Hardware uses the programmed value to decode the target of PCI configuration space accesses.

Mnemonic

Address Offset 0x018

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	cfgrco_pci18_primary_bus(7:0)	RW	0	Primary PCI Bus number. This is typically programmed to 0x00 for the root complex.
15:08	cfgrco_pci18_secondary_bus(7:0)	RW	0	Secondary PCI Bus number. This is typically programmed to 0x01 for the root complex.
23:16	cfgrco_pci18_subordinate_bus(7:0)	RW	0	Subordinate PCI Bus number. This is typically programmed to 0xFF for the root complex.
31:24	Reserved	RO	0	Reserved.

4.5.2.6 PCI - IO Base/Limit and Secondary Status Register

The IO Base/Limit values specify the address range of IO commands that are can be forwarded to the PCIe link. This register can be set to allow all IO commands to be sent to the PCIe link. This register contains Secondary status information, that reflects the status information on the Secondary Bus.

Mnemonic

Address Offset 0x01C

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	Reserved	RO	0	IO Base Register[7:0]. Current implementation does not use this value.
15:08	Reserved	RO	0	IO Limit Register[7:0]. Current implementation does not use this value.
23:16	Reserved	RO	0	Reserved.
24	Signaled Poisoned TLP	RW1CH	0	Secondary Status Register[8] - Master Data Parity Error. This status bit reflects the state of the internal signal pi_err_sig_poisoned_tlp. This indicates the logic signaled that it was sending a poisoned TLP outbound to the PCIe link. -- gate pi_err_sig_poisoned_tlp with enable bit from reg pci_3c(16) s_sig_poison ≤ pi_err_sig_poisoned_tlp and s_pci_3c_rd_data(16);
26:25	Reserved	RO	0	Reserved.

Bit	Field Mnemonic	Type	Reset Value	Description
27	pi_err_sig_cmpl_abort	RW1CH	0	Secondary Status Register[11] - Signaled Target Abort. This status bit reflects the state of the internal signal pi_err_sig_cmpl_abort. This indicates the logic signaled that it was sending a Completion TLP with Completer Abort status outbound to the PCIe link.
28	pi_err_rcv_cpl_ca	RW1CH	0	Secondary Status Register[12] - Received Target Abort. This status bit reflects the state of the internal signal pi_err_rcv_cpl_ca. This indicates the logic received a Completion TLP with Completer Abort status inbound from the PCIe link.
29	pi_err_rcv_cpl_ur	RW1CH	0	Secondary Status Register[13] - Received Master Abort. This status bit reflects the state of the internal signal pi_err_rcv_cpl_ur. This indicates the logic received a Completion TLP with Unsupported Request status inbound from the PCIe link.
30	pi_aer30_fatal_error_msg or pi_aer30_non_fatal_error_msg	RW1CH	0	Secondary Status Register[14] - Received System Error. This status bit reflects the state of the internal signals pi_aer30_fatal_error_msg OR pi_aer30_non_fatal_error_msg. This indicates the logic received an ERR_FATAL or ERR_NONFATAL Message TLP from the PCIe link.
31	pi_err_rcv_poisoned_tlp	RW1CH	0	Secondary Status Register[15] - Detected Parity Error. This status bit reflects the state of the internal signal pi_err_rcv_poisoned_tlp. This indicates the logic received a TLP with a poisoned data indication in the header. The can be either for a Posted or Completion TLP.

4.5.2.7 PCI - Memory Base/Limit Register

The Memory Base/Limit values specify the address range of memory commands that are can be forwarded to the PCIe link. Set this register to allow all memory commands to be sent to the PCIe link. The memory mapped space accessed is assumed to be greater than 4GB.

Mnemonic

Address Offset 0x020

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	CFG_PCI20_MEM_BASE[03:00]	RO	0	Memory Base Register[03:00]. Hard-coded to '0'.
15:04	CFG_PCI20_MEM_BASE[15:04]	RO	0x001	Memory Base Register[15:04]. Hard-coded to a value of 0x001 so that the Base > Limit. This will allow all addresses to pass inbound.
31:16	CFG_PCI20_MEM_LIMIT[15:00]	RO	0	Memory Limit Register[15:00]. Hard-coded to a value of all zeros.

4.5.2.8 PCI - Prefetch Base/Limit Register

The Prefetch Base/Limit values specify the address range of memory commands that are can be forwarded to the PCIe link. This register is set to allow all memory commands to be sent to the PCIe link. The prefetchable memory space can exist anywhere in the 64-bit address space.

Mnemonic

Address Offset 0x024

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	CFG_PCI24_PREFETCH_BASE[03:00]	RO	0x1	Prefetch Base Register[03:00]. Value will be 0x0 for 32-bit Prefetch space addressing. Value will be 0x1 for 64-bit Prefetch space addressing.
15:04	CFG_PCI24_PREFETCH_BASE[15:04]	RO	0x001	Prefetch Base Register[15:04]. Hard-coded to a value of 0x001 so that the Base > Limit. This will allow all addresses to pass inbound.
19:16	CFG_PCI24_PREFETCH_LIMIT[03:00]	RO	0x1	Prefetch Limit Register[03:00]. Value will be 0x0 for 32-bit Prefetch space addressing. Value will be 0x1 for 64-bit Prefetch space addressing.
31:20	CFG_PCI24_PREFETCH_LIMIT[15:04]	RO	0	Prefetch Limit Register[15:04]. Hard-coded to a value of all zeros.

4.5.2.9 PCI - Prefetch Base Upper Register

The Prefetch Base Upper values specify the upper 32 bits of the address range of memory commands that are can be forwarded to the PCIe link. This register is set to allow all memory commands to be sent to the PCIe link. The prefetchable memory space can exist anywhere in the 64-bit address space.

Mnemonic

Address Offset 0x028

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	CFG_PCI28_PREFETCH_BASE_UPPER[31:00]	RO	0	Prefetch Base Upper Register[31:00]. Contains the upper 32 bits of the 64 bit Prefetchable address space.

4.5.2.10 PCI - Prefetch Limit Upper Register

The Prefetch Limit Upper values specify the upper 32 bits of the address range of memory commands that are can be forwarded to the PCIe link. This register is set to allow all memory commands to be sent to the PCIe link. The prefetchable memory space can exist anywhere in the 64-bit address space.

Mnemonic

Address Offset 0x02C

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	CFG_PCI28_PREFETCH_LIMIT_UPPER[31:00]	RO	0	Prefetch Limit Upper Register[31:00]. Contains the upper 32 bits of the 64 bit Prefetchable address space limit.

4.5.2.11 PCI - Capabilities Pointer Register

PCI specification register.

Mnemonic

Address Offset 0x034

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	Capability Pointer(7:0)	RO	0x40	Pointer to next capability space with the PCI CFG space.
31:08	Reserved	RO	0	Reserved.

4.5.2.12 Bridge Control Register

This register is used by software for the sole purpose of setting and clearing the Secondary Bus Reset bit 22. All other bits in this register are not to be set and are unused. This is one means of asserting PCIe Hot Reset.

Mnemonic

Address Offset 0x03C

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	pci_bcr_perr_rsp_en	RO	0	Bridge Control Register [0] - Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register - 0x1C, bit 24. This bit is hard-coded to '0' and is not used.
15:08	pci_bcr_serr_en	RO	1	Bridge Control Register [1] - System Error Enable. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages from the secondary bus to the primary bus. This bit is hard-coded to '1'.
16	Reserved	RO	0	Bridge Control Register [5:2] - Reserved.
17	cfgrco_pci3c_bcr_sb_reset	RW	0	Bridge Control Register [6] - Secondary Bus Reset. This bit will put the PCIe link in the PCIe Hot Reset state.
21:18	Reserved	RO	0	Bridge Control Register [15:7] - Reserved.
22	pci_bcr_perr_rsp_en	RO	0	Bridge Control Register [0] - Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register - 0x1C, bit 24. This bit is hard-coded to '0' and is not used.
31:23	pci_bcr_serr_en	RO	1	Bridge Control Register [1] - System Error Enable. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages from the secondary bus to the primary bus. This bit is hard-coded to '1'.

4.5.3 PM - Power Management Registers (0x040 to 0x044)

4.5.3.1 PM - Capability ID/Next PTR/PMC Register

This register describes the power management capabilities of the PCI Express port and is required for all PCI Express devices.

Mnemonic

Address Offset 0x040

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	pm_capability_id	RO	0x01	PM - Capability ID Register [07:00].
15:08	pm_next_ptr	RO	0x48	PM - Next Pointer Register [07:00]. Pointer to next capability structure. Points to 0x48 for the EC Capabilities structure.
18:16	pm_pmc_version_num	RO	0x3	PM - Capabilities Register (PMC) [02:00] - PM Version Number.
26:19	Reserved	RO	0	PM - Capabilities Register (PMC) [10:03] - Various. Driven to zeros in PHB4.
31:27	pm_pmc_pme_support	RO	0x19	PM - Capabilities Register (PMC) [15:11] - PME Support. This field indicates the power states in which a device may generate a Power Management Event (PME). PMC[15] - from D3 Cold (always returns 1). PMC[14] - from D3 Hot (always returns 1). PMC[13] - from D2 (always returns 0). PMC[12] - from D1 (always returns 0). PMC[11] - from D0 (always returns 1).

4.5.3.2 PM - PMCSR/PMCSR_BSE/Data Register

The 16-bit power management control/status register (PMCSR) is used to communicate power management control and status information between the hardware and software.

Mnemonic

Address Offset 0x044

Bit	Field Mnemonic	Type	Reset Value	Description
01:00	pm_pmcsr_pwr_state	RO	0	PM - PMCSR [01:00] - Power State. Software uses this field to determine the current PM state of the Root port by reading this field or places the Root port in a new PM state by writing this field. If there is a write value to a state that is not supported the write completes but will not change the current value in this field. Power State: • 00 - D0 • 01 - D1 (not supported) • 10 - D2 (not supported) • 11 - D3 Hot This will always report D0 state, all zeros.
31:02	Reserved	RO	0	PM - PMCSR [15:02] - Various Bits. Not used by PHB4.
31:02	Reserved	RO	0	PM – PMCS – Bridge Support Extensions [07:00]. Not used by PHB4.
31:02	Reserved	RO	0	PM - Data Register [07:00]. Not used by PHB4.

4.5.4 EC - PCI Express Capabilities Registers (0x048 to 0x078)

4.5.4.1 EC - Capability ID/Next PTR/PCI Express Capabilities Register

The PCI Express Capabilities Register communicates various functional capabilities within the PCIe port.

Mnemonic

Address Offset 0x048

Bit	Field Mnemonic	Type	Reset Value	Description
07:00	ec_capability_id	RO	0x10	EC - Capability ID Register [07:00].
15:08	ec_next_ptr	RO	0	EC - Next PTR Register [07:00]. Pointer to next capability structure. Points to the end since the EC structure is the last capability structure.
19:16	ec_capability_ver	RO	0x2	EC - Capabilities Register [03:00] - PCI Express Capabilities Version 2.0.
23:20	ec_device_port_type	RO	0x4	EC - Capabilities Register [07:04] - Device/Port Type. The Device/Port Type will always indicate a root complex.
24	pi_sys_ec00_slot	ROH	0	EC - Capabilities Register [8] - Slot Implemented. Indicates the Root port or Switch downstream port is connected to an add-in card slot as opposed to an integrated component being disabled. This value is driven by the input signal: pi_sys_ec00_slot.
31:25	Reserved	RO	0	EC - Capabilities Register [15:09] – Various bits. Not used by PHB4.

4.5.4.2 EC - Device Capabilities Register

The Device Capabilities Register is a 32-bit register that describes the PCI Express device specific capabilities in detail to the software.

Mnemonic

Address Offset 0x04C

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	pi_sys_ec04_max_payload(2:0)	ROH	0x2	EC - Device Capabilities Register [02:00] - Max Payload Size. Max_Payload_Size supported by the PCIe port. Defined encoding are: • 000 - 128-byte • 001 - 256-byte • 010 – 512-byte (default) • 011 – 1024-byte (not supported, behavior undefined) • 100 – 2048-byte (not supported, behavior undefined) • 101 – 4096-byte (not supported, behavior undefined) • 110 – Reserved (not supported, behavior undefined) • 111 – Reserved (not supported, behavior undefined)
04:03	Reserved	RO	0	EC - Device Capabilities Register [04:03] - Phantom Functions.
5	pi_sys_ec04_extended_tag	ROH	0	EC - Device Capabilities Register [5] - Extended Tag Field Support. This bit indicates the maximum supported size of the TLP Tag field as a Requester. Defined encodings are: • 0 - 5-bit Tag field supported • 1 - 8-bit Tag field supported NOTE: This is hard-coded to '0'.
08:06	ec_ep_l0s_latency	RO	0	EC - Device Capabilities Register [08:06] - Endpoint L0s Latency Endpoint only; not used.
11:09	ec_ep_l1_latency	RO	0	EC - Device Capabilities Register [11:09] - Endpoint L1 Latency Endpoint only; not used.
14:12	Reserved	RO	0	EC - Device Capabilities Register [14:12] - Reserved
15	ec_rolebased	RO	1	EC - Device Capabilities Register [15] - Rolebased Error Reporting Must be set to '1'.
31:16	Reserved	RO	0	EC - Device Capabilities Register [31:16] - Various.

4.5.4.3 EC - Device Control/Status Register

The Device Control/Status Register is used to communicate device status and control information between the hardware and software.

Mnemonic

Address Offset 0x050

Bit	Field Mnemonic	Type	Reset Value	Description
00	ec_corr_rpt_en	RO	0	EC - Device Control Register [0] - Correctable Error Reporting Enable. Hard-coded to '0'. This register bit is not used in the design.
01	ec_nonfatal_rpt_en	RO	0	EC - Device Control Register [1] - Non-fatal Error Reporting Enable. Hard-coded to '0'. This register bit is not used in the design.
02	ec_fatal_rpt_en	RO	0	EC - Device Control Register [2] - Fatal Error Reporting Enable. Hard-coded to '0'. This register bit is not used in the design.
03	ec_ur_rpt_en	RO	0	EC - Device Control Register [3] - Unsupported Request Reporting Enable. Hard-coded to '0'. This register bit is not used in the design.
04	ec_relaxed_ordering	RO	0	EC - Device Control Register [4] - Relaxed Ordering. Hard-coded to '0'. This register bit is not used in the design.
07:05	cfgrco_ec08_max_payload_size(2:0)	RW	0x2	EC - Device Control Register [07:05] - Max_Payload_Size. Sets the maximum TLP data payload for the device. As a receiver, the device must handle TLPs with payloads as large as the set value. As a transmitter, the device must not generate TLP payloads that exceed this value. Defined encodings are: • 000 - 128-byte • 001 - 256-byte • 010 – 512-byte (default) • 011 – 1024-byte (not supported, behavior undefined) • 100 – 2048-byte (not supported, behavior undefined) • 101 – 4096-byte (not supported, behavior undefined) • 110 – Reserved (not supported, behavior undefined) • 111 – Reserved (not supported, behavior undefined) This value defaults to the value set by the input signal: pi_sys_ec04_max_payload(2:0).
08	pi_sys_ec04_extended_tag	ROH	0	EC - Device Control Register [8] - Extended Tag Field Enable. This value is hard-coded to the value set by the input signal: pi_sys_ec04_extended_tag.
09	ec_phatom_func_en	RO	0	EC - Device Control Register [9] - Phantom Function Enable.
10	ec_aux_pwr_pm_en	RO	0	EC - Device Control Register [10] - Auxiliary (AUX) Power PM Enable.
11	ec_nosnoop_en	RO	0	EC - Device Control Register [11] - Enable No Snoop.

Bit	Field Mnemonic	Type	Reset Value	Description
14:12	ec_max_read_req_size	RO	0	EC - Device Control Register [14:12] - Max_Read_Request_Size. This field sets the maximum Read Request size for the function as a requester. The function must not generate Read Requests with size exceeding the set value. Defined encodings are: • 000 - 128 byte • 001 - 256 byte • 010 - 512 byte • 011 - 1024 byte • 100 - 2048 byte • 101 - 4096 byte • 110 - Reserved • 111 - Reserved This is hard-coded to all zeros for the root complex.
15	ec_corr_err_detected	RW1CH	0	EC - Device Status Register [0] - Correctable Error Detected. When this bit is set it indicates that one or more correctable errors were detected since this bit was last cleared by software. This bit will be set regardless if correctable errors are enabled in the Device Control Register. Correctable error is signaled by the logical 'OR' of the following inputs: • tldlpo_dl_ec08_baddllp • tldlpo_dl_ec08_badtlp • tldlpo_dl_ec08_receivererror • tldlpo_dl_ec08_replayrollover • tldlpo_dl_ec08_replaytimeout
16	ec_nonfatal_err_detected	RW1CH	0	EC - Device Status Register [1] - Non-Fatal Error Detected. When this bit is set it indicates that one or more non-fatal errors were detected since this bit was last cleared by software. This bit will be set regardless if non- fatal errors are enabled in the Device Control Register. Non-fatal error is signaled by the logical 'OR' of the following inputs: tldlpo_dl_ec08_dllpe tldlpo_dl_ec08_surprisedown tldlpo_tl_ec08_fcpe -- pi_err_rcv_ecrc_err pi_err_rcv_malf_tlp pi_err_rcv_overflow pi_err_rcv_poisoned_tlp pi_err_rcv_unexp_cpl pi_err_sig_cmpl_abort pi_err_sig_cmpl_timeout pi_err_sig_poisoned_tlp

Bit	Field Mnemonic	Type	Reset Value	Description
17	ec_fatal_err_detected	RW1CH	0	EC - Device Status Register [2] - Fatal Error Detected. When this bit is set it indicates that one or more fatal errors were detected since this bit was last cleared by software. This bit will be set regardless if fatal errors are enabled in the Device Control Register. Fatal errors are the same as Non-Fatal error listed above.
18	pi_err_rcv_unsup_req	RW1CH		EC - Device Status Register [3] - Unsupported Request Detected. When this bit is set it indicates that one or more unsupported requests were received since this bit was last cleared by software. This bit will be set regardless if unsupported request reporting is enabled in the Device Control Register.
19	Reserved	RO	0	Reserved.
20	ec_corr_err_detected	RW1CH	0	EC - Device Status Register [0] - Correctable Error Detected. When this bit is set it indicates that one or more correctable errors were detected since this bit was last cleared by software. This bit will be set regardless if correctable errors are enabled in the Device Control Register. Correctable error is signaled by the logical 'OR' of the following inputs: • tldlpo_dl_ec08_baddllp • tldlpo_dl_ec08_badtlp • tldlpo_dl_ec08_receivererror • tldlpo_dl_ec08_replayrollover • tldlpo_dl_ec08_replaytimeout

4.5.4.4 EC - Link Capabilities Register

The Link Capabilities Register is a 32-bit register that describes the PCI Express link specific capabilities in detail to the software.

Mnemonic

Address Offset 0x054

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	pi_sys_ec0c_maxlinkspeed(3:0)	ROH	0x4	EC - Link Capabilities Register [03:00] - Max Link Speed. This value is hard-coded to the value set by the input signal: pi_sys_ec0c_maxlinkspeed(3:0).
09:04	pi_sys_ec0c_maxlinkwidth(5:0)	ROH	0x10	EC - Link Capabilities Register [09:04] - Max Link Width. This value is hard-coded to the value set by the input signal: pi_sys_ec0c_maxlinkwidth(5:0).
11:10	ec_aslpm_support	RO	0	EC - Link Capabilities Register [11:10] - Active State Link PM Support. Hard-coded to '0'. This register bit is not used in the design.
14:12	ec_link_l0s_exit_lat	RO	0	EC - Link Capabilities Register [14:12] - L0s Exit Latency. Hard-coded to '0'. This register bit is not used in the design.
17:15	ec_link_l1_exit_lat	RO	0	EC - Link Capabilities Register [17:15] - L1 Exit Latency. Hard-coded to '0'. This register bit is not used in the design.
18	Reserved	RO	0	Reserved.
19	Surprise Down Error Reporting Capable	RO	0	EC - Link Capabilities Register [19]. Hard-coded to '0'. This register bit is not used in the design.
20	Data Link Layer Link Active Reporting Capable	RO	0x1	EC - Link Capabilities Register [20]. This bit must be set to '1' if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable field of the Slot Capabilities register), this bit must be set to '1'.
21	Data Link Bandwidth Notification Capable	RO	0x1	EC - Link Capabilities Register [21] - Link Bandwidth Notification Capable. A value of '1' indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting links wider than x1 and/or multiple link speeds.
23:22	Reserved	RO	0	Reserved.
31:24	ec_port_number	RO	0	EC - Link Capabilities Register [31:24] - Port Number. This value indicates the hardware port number of the PCIe port. Hard-coded to '0'. This register bit is not used in the design.

4.5.4.5 EC - Link Control/Status Register

The Link Control/Status Register is used to communicate link status and control information between the hardware and software.

Mnemonic

Address Offset 0x058

Bit	Field Mnemonic	Type	Reset Value	Description
01:00	ec_linkpm_control	RO	0	EC - Link Control Register [1:0] - ASPM Control. This field controls the level of ASPM supported on the given PCI Express Link. Defined encodings are: • 00 - Disabled • 01 - L0s Entry Enabled • 10 - L1 Entry Enabled • 11 - L0s and L1 Entry Enabled Hard-coded to '0'. This register bit is not used in the design.
2	Reserved	RO	0	EC - Link Control Register [2].
3	ec_rcb128	RO	1	EC - Link Control Register [3] - Read Completion Boundary. Defined encodings are: • 0 - 64 byte • 1 - 128 byte
4	cfgrco_ec10_linkdisable	RW	0	EC - Link Control Register [4] - Link Disable. Disables the link when written to a 1.
5	cfgrco_ec10_retrainlink	WO	0	EC - Link Control Register [5] - Retrain Link. Writing to this register causes the link to be retrained by changing the Physical Layer state from LTSSM state to the Recovery state. Reads from this register bit will always return a 0.
6	ec_common_clock_en	RO	0	EC - Link Control Register [6] - Common Clock Configured. Hard-coded to '0'. This register bit is not used in the design.
7	*cfgrco_ec10*_extendedsynch	RW	0	EC - Link Control Register [7] - Extended Sync. When set to 1 forces extended sequences during link synchronization to aid in achieving bit and symbol lock for test equipment that can be monitoring the link.
8	ec_clk_power_mgmt_en	RO	0	EC - Link Control Register [8] - Enable clock power management. Hard-coded to '0'. This register bit is not used in the design.
9	*cfgrco_ec10*_hwauto_width_disable	RW	0	EC - Link Control Register [9] - Hardware Autonomous Width Disable. When Set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing link width.
10	cfgrco_ec10_linkbwmgmt	RO	0	EC - Link Control Register [10] - Link Bandwidth Management Interrupt Enable. Hard-coded to '0'. This register bit is not used in the design.
11	cfgrco_ec10_linkautobw	RO	0	EC - Link Control Register [11] - Link Autonomous Bandwidth Interrupt Enable. Hard-coded to '0'. This register bit is not used in the design.

Bit	Field Mnemonic	Type	Reset Value	Description
15:12	Reserved	RO	0	EC - Link Control Register [15:12].
19:16	tldlpo_dl_ec10_currentlinkspeed(3:0)	ROH	0x1	EC - Link Status Register [03:00] - Current Link Speed. Defined encodings are: • 0001 - 2.5 GT/s PCIe Link • 0010 - 5.0 GT/s PCIe Link • 0011 - 8.0 GT/s PCIe Link • 0100 – 16.0 GT/s PCIe Link Other encodings are reserved. This field is undefined when the link is not up.
25:20	tldlpo_dl_ec10_neglinkwidth(5:0)	ROH	0x20	EC - Link Status Register [09:04] - Negotiated Link Width. Defined encodings are: • 00_0001 - x1 • 00_0010 - x2 • 00_0100 - x4 • 00_1000 - x8 • 00_1100 - x12 • 01_0000 - x16 • 10_0000 - x32 Return value is based on link arriving at a stable state and can vary based on configuration.
26	Reserved	RO	0	EC - Link Status Register [10].
27	tldlpo_dl_ec10_linktraining	ROH	0	EC - Link Status Register [11] - Link Training. This bit indicates that the Physical Layer LTSSM is in the configuration or recovery state, or that '1' was written to the retrain link bit but link training has not yet begun. Hardware clears this bit to '0' when the LTSSM exits the configuration/recovery state.
28	ec_slot_clk_cfg	RO	0	EC - Link Status Register [12] - Slot Clock Configuration. Indicates whether the port uses the same physical reference clock as provided on the PCI Express connector. This bit must be clear if the component uses an independent clock even if one can be provided by the connector. This bit reflects the value in the PCIe System Configuration Register, bit [31], which drives the sys_ec10_slotclock signal to the CFG core. Always returns value of PCIe System Configuration Register, bit [31]. Hard-coded to '0'. This register bit is not used in the design.
29	tldlpo_tl_ec10_linkactive	ROH	0	EC - Link Status Register [13] - Data Link Layer Link Active. This bit indicates the status of the Data Link Control and Management State Machine. It returns a 1b to indicate the DL_Active state, 0b otherwise.

Bit	Field Mnemonic	Type	Reset Value	Description
30	tldlpo_dl_ec10_linkbwmgmt	RW1CH	0	EC - Link Status Register [14] - 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. Note: This bit is set following any write of 1b to the Retrain Link bit, including when the link is in the process of retraining for some other reason. • 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.
31	tldlpo_dl_ec10_linkautobw	RW1CH	0	EC - Link Status Register [15] - 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.

4.5.4.6 EC - Slot Capabilities Register

The Slot Capabilities register is used to communicate slot capabilities to the software interface.

Mnemonic

Address Offset 0x05C

Bit	Field Mnemonic	Type	Reset Value	Description
00	attn_button_present	RO	0	EC - Slot Capabilities Register [0] – Attention Button Present. When set, this bit indicates that an Attention Button for this slot is electrically controlled by the chassis. Hard-coded to '0'.
01	pwr_ctrlr_present	RO	0	EC - Slot Capabilities Register [1] – Power Controller Present. When set, this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor). Hard-coded to '0'.

Bit	Field Mnemonic	Type	Reset Value	Description
02	MRL_sensor_present	RO	0	EC - Slot Capabilities Register [2] – MRL Sensor Present. When set, this bit indicates that an MRL Sensor is implemented on the chassis for this slot. Hard-coded to '0'.
03	attn_indicator_present	RO	0	EC - Slot Capabilities Register [3] – Attention Indicator Present. When set, this bit indicates that an Attention Indicator is electrically controlled by the chassis. Hard-coded to '0'.
04	pwr_indicator_present	RO	0	EC - Slot Capabilities Register [4] – Power Indicator Present. When set, this bit indicates that a Power Indicator is electrically controlled by the chassis for this slot. Hard-coded to '0'.
05	hot_plug_surprise	RO	0	EC - Slot Capabilities Register [5] – Hot Plug Surprise. When set, this bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation. Hard-coded to '0'.
06	hot_plug_capable	RO	0	EC - Slot Capabilities Register [6] – Hot Plug Capable. When set, this bit indicates that this slot is capable of supporting hot-plug operations. Hard-coded to '0'.
14:07	ec14_slt_pl_msg_value	RW	0	EC - Slot Capabilities Register [14:7] – Slot Power Limit Value. In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by the slot or by other means to the adapter. Power limit (in watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field except when the Slot Power Limit Scale field equals 00b (1.0x) and Slot Power Limit Value exceeds EFh, the following alternative encodings are used: • F0h = 250 watt Slot Power limit • F1h = 275 watt Slot Power limit • F2h = 300 watt Slot Power limit • F3h to FFh = Reserved for Slot Power limit values above 300 watt Writes to this register cause the port to send the Set_Slot_Power_Limit Message.

Bit	Field Mnemonic	Type	Reset Value	Description
16:15	ec14_slt_pl_msg_scale	RW	0	EC - Slot Capabilities Register [16:15] – Slot Power Limit Scale. Specifies the scale used for the Slot Power Limit Value. Range of Values: • 00b = 1.0x • 01b = 0.1x • 10b = 0.01x • 11b = 0.001x Writes to this register also cause the port to send the Set_Slot_Power_Limit Message.
17	electromech_interlock_present	RO	0	EC - Slot Capabilities Register [17] – Electromechanical Interlock Present. When set, this bit indicates that an electromechanical interlock is implemented on the chassis for this slot. Hard-coded to '0'.
18	no_cmd_completed_support	RO	0	EC - Slot Capabilities Register [17] – No Command Completed Support. When set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be set if the hot-plug capable port is able to accept writes to all fields of the Slot Control register without delay between successive writes. Hard-coded to '0'.
31:19	Reserved	RO	0	Reserved.

4.5.4.7 EC - Slot Control/Status Register

The Slot Control/Status register is used to control extended device functions.

Mnemonic

Address Offset 0x060

Bit	Field Mnemonic	Type	Reset Value	Description
12:00	Reserved	RO	0	EC - Slot Control Register [12:00] – Various Enable Controls. These controls are not supported by this design and are read-only. Hard-coded to '0'.
13	ec_18_slt_pl_msg_auto_disable	RW	1	EC - Slot Control Register [13] – Auto Slot Power Limit Disable. When set, this disables the automatic sending of a Set_Slot_Power_Limit Message when a link transitions from a non-DL_Up status to a DL_Up status. Default value is '1'.
15:14	Reserved	RO	0	EC - Slot Control Register [15:14] - Reserved.
31:21	Reserved	RO	0	EC - Slot Status Register [05:00] – Reserved.
22	pi_ec18_presence_detect	RCW	0	EC - Slot Status Register [06] – Presence Detect State. This bit indicates the presence of an adapter in the slot, reflected by the logical “OR” of the physical layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot’s corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. Defined encodings are: • 0b Slot Empty • 1b Card Present in slot
31:23	Reserved	RO	0	EC - Slot Status Register [15:07] – Reserved.

4.5.4.8 EC - Root Control Register

The Root Control register is used to control all root complex specific parameters.

Mnemonic

Address Offset 0x064

Bit	Field Mnemonic	Type	Reset Value	Description
0	ec_root_correrr_en	RO	0	EC - Root Control Register [0] - System Error on Correctable Error Enable. When set to a 1, a system error is generated if the Root Complex receives an ERR_COR message or if a correctable error is detected within the Root Complex itself. Hard-coded to '0'. This register bit is not used in the design.
1	ec_root_nfatalerr_en	RO	0	EC - Root Control Register [1] - System Error on Non-Fatal Error Enable. When set to a 1, a system error is generated if the Root Complex receives an ERR_NONFATAL message or if a non-fatal error is detected within the Root Complex itself. Hard-coded to '0'. This register bit is not used in the design.
2	ec_root_fatalerr_en	RO	0	EC - Root Control Register [2] - System Error on Fatal Error Enable. When set to a 1, a system error is generated if the Root Complex receives an ERR_FATAL message or if a fatal error is detected within the Root Complex itself. Hard-coded to '0'. This register bit is not used in the design.
3	ec_root_pmeint_en	RO	0	EC - Root Control Register [3] - PME Interrupt Enable. When set to a 1, the Root Complex will generate and interrupt to the system if it receives a PME Message from the port. The PME Message will also set the Root Status Register PME bit. Software is capable if generating an interrupt of this bit is written to a 1 while it is cleared at 0 and the PME bit is set in the Root Status Register. Hard-coded to '0'. This register bit is not used in the design.
4	cfgrco_ec1c_crs_sw_visibil_en	RO	0	EC - Root Control Register [4] - CRS Software Visibility Enable. Hard-coded to '0'. this register bit is not sued in the design.
15:05	Reserved	RO	0	Reserved.
16	ec_crs_visibility_cap	RO	0	EC - Root Control Register [16] - CRS Software Visibility Capable.
31:17	Reserved	RO	0	Reserved.

4.5.4.9 EC - Device Capabilities 2 Register

The Device Capabilities 2 register is used to communicate Extended Device capabilities to the software interface.

Mnemonic

Address Offset 0x06C

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	ec24_comp_to_ranges_supported	ROH	0xF	EC - Device Capabilities 2 Register [3:0] - Completion Timeout Ranges Support. This value indicates device support for the optional completion timeout programmability mechanism which allows system software to modify the completion timeout value. Four time value ranges are supported: • Range A = 50 µs to 10 ms • Range B= 10 ms to 250 ms • Range C= 250 ms to 4 s • Range D= 4S to 64 s Bits are set according to the table below to show timeout value ranges supported: • 4'b0000 = Completion Timeout programing not supported, the device must implement a timeout value in range 50uS to 50mS • 4'b0001 = Range A • 4'b0010 = Range B • 4'b0011 = Range A and Range B • 4'b0110 = Range B and Range C • 4'b0111 = Range A and Range B and Range C • 4'b1110 = Range B and Range C and Range D • 4'b1111 = All 4 Ranges All other values are reserved. The design supports all ranges.
04	ec24_comp_to_dis_support	ROH	1	EC - Device Capabilities 2 Register [4] - Completion Timeout Disable Support. When set, this input pin indicates the support for completion timeout disable mechanism.
05	ec24_ari_forwarding_supported	ROH	1	EC - Device Capabilities 2 Register [5] - ARI Forwarding Supported. When set, this input pin indicates the support for the ARI Forwarding mechanism.
15:06	Reserved	RO	0	Reserved.
16	10-bit Tag Completer Supported	RO	0	10-bit Tag Completer Supported. If this bit is set, the function supports 10-bit Tag Completer capability; otherwise, the function does not.

Bit	Field Mnemonic	Type	Reset Value	Description
17	10-bit Tag Requester Supported	RO	0	10-bit Tag Requester Supported. If this bit is set, the function supports 10-bit Tag Requester capability; otherwise, the function does not. This bit must not be set if the 10-bit Tag Completer Supported bit is clear. Note: 10-bit tag field generation must be enabled by the 10-bit Tag Requester Enable bit in the Device Control 2 register of the Requester Function before 10-bit tags can be generated by the requester.
31:06	Reserved	RO	0	Reserved.
	ec24_comp_to_ranges_supported	ROH	0xF	EC - Device Capabilities 2 Register [3:0] - Completion Timeout Ranges Support. This value indicates device support for the optional completion timeout programmability mechanism which allows system software to modify the completion timeout value. Four time value ranges are supported: • Range A = 50µs to 10 ms • Range B= 10 ms to 250 ms • Range C= 250 ms to 4 s • Range D= 4 s to 64 s Bits are set according to the table below to show timeout value ranges supported: • 4'b0000 = Completion Timeout programing not supported, the device must implement a timeout value in range 50uS to 50mS • 4'b0001 = Range A • 4'b0010 = Range B • 4'b0011 = Range A and Range B • 4'b0110 = Range B and Range C • 4'b0111 = Range A and Range B and Range C • 4'b1110 = Range B and Range C and Range D • 4'b1111 = All 4 Ranges All other values are reserved. The design supports all ranges.

4.5.4.10 EC - Device Control/Status 2 Register

The Device Control/Status register is used to control extended device functions.

Mnemonic

Address Offset 0x070

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	*cfgrco_ec28*_comp_to_value(3:0)	RW	0	EC - Device Control 2 Register [03:00] - Completion Timeout Value. In device Functions that support Completion Timeout programmability, this field allows system software to modify the Completion Timeout value. A function that does not support this optional capability must hardwire this field to 0000b and is required to implement a timeout value in the range 50 µs to 50 ms. Functions that support Completion Timeout programmability must support the values given below corresponding to the programmability ranges indicated in the Completion Timeout Ranges Supported field. Defined encodings: • 4'b0000 Default range: 16 ms • It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms. • 4'b0001 32 µs • 4'b0010 4 ms • 4'b0101 16 ms • 4'b0110 64 ms • 4'b1001 256 ms • 4'b1010 1 s • 4'b1101 4 s • 4'b1110 32 s • 4'b1111 reserved Values not defined above are reserved. Software is permitted to change the value in this field at any time. For requests already pending when the Completion Timeout Value is changed, hardware is permitted to use either the new or the old value for the outstanding requests, and is permitted to base the start time for each request either on when this value was changed or on when each request was issued. The default value for this field is 4'b0000.
04	cfgrco_ec28_comp_to_disable	RW	0	EC - Device Control 2 Register [4] - Completion Timeout Disable. When set, this bit disables the Completion Timeout mechanism. Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued. The default value for this bit is 0b.
05	cfgrco_ec28_ari_fwd_enable	RO	1	EC - Device Control 2 Register [5] - ARI Forwarding Enable. When set, the root port will forward Type0 Configuration requests to non-zero device numbers (representing extended ARI function numbers). Hard-coded to '1', always allowed in this design.

Bit	Field Mnemonic	Type	Reset Value	Description
11:06	Reserved	RO	0	EC - Device Control 2 Register [15:06] - Reserved.
12	10-Bit Tag Requester Enable	RO	0	10-bit Tag Requester Enable. This bit, in combination with the Extended Tag Field Enable bit in the Device Control register, determines how many tag field bits a requester is permitted to use. When the 10-bit Tag Requester Enable bit is set, the requester is permitted to use 10-bit tags. Software should not change the value of this bit while the function has outstanding nonposted requests; otherwise, the result is undefined. Functions that do not implement 10-bit Tag Requester capability are permitted to hardwire this bit to 0b. Default value of this bit is 0b.
31:13	Reserved	RO	0	EC - Device Status 2 Register [15:00] - Reserved.

4.5.4.11 EC - Link Capabilities 2 Register

The Link Capabilities 2 register is used to define extended link capabilities.

Mnemonic

Address Offset 0x074

Bit	Field Mnemonic	Type	Reset Value	Description
0	Reserved	RO	0	EC - Link Capabilities 2 Register [0] - reserved.
07:01	Supported Link Speeds Vector	RO	0b0001111	Supported Link Speeds Vector – This field indicates the supported link speed(s) of the associated Port. For each bit, a value of 1b indicates that the corresponding link speed is supported; otherwise, the link speed is not supported. Bit definitions within this field are: • Bit 0 2.5 GT/s • Bit 1 5.0 GT/s • Bit 2 8.0 GT/s • Bit 3 16.0 GT/s • Bits 6:4 RsvdP Multi-Function devices associated with an upstream port must report the same value in this field for all functions.

Bit	Field Mnemonic	Type	Reset Value	Description
8	Crosslink Supported	RO	0	Crosslink supported when set to 1b, this bit indicates that the associated port supports cross links. When set to 0b on a port that supports link speeds of 8.0 GT/s or higher, this bit indicates that the associated port does not support cross links. When set to 0b on a port that only supports link speeds of 2.5 GT/s or 5.0 GT/s, this bit provides no information regarding the port's level of crosslink support. It is recommended that this bit be set in any port that supports cross links even though doing so is only required for ports that also support operating at 8.0 GT/s or higher link speeds. Note: Software uses this bit when referencing fields whose definition depends on whether or not the port supports crosslinks. Multi-Function devices associated with an upstream port must report the same value in this field for all Functions.
22:09	Reserved	RO	0	EC - Link Capabilities 2 Register [22:09] - reserved.
23	Retimer Presence Detect Supported	RO	1	Retimer Presence Detect Supported. For a downstream port, when set to 1b, this bit indicates that the associated port supports detection and reporting of Retimer presence. This bit is valid for downstream ports. This bit is reserved for upstream ports.
24	Two Retimers Presence Detect Supported	RO	0	Two Retimers Presence Detect Supported. When set to 1b, this bit indicates that the associated port supports detection and reporting of two Retimers presence. This bit must be set to 1b in a port when the Supported Link Speeds Vector of the Link Capabilities 2 register indicates support for a link speed of 16.0 GT/s or higher. It is permitted to be set to 1b regardless of the supported link speeds if the Retimer Presence Detect Supported bit is also set to 1b.
30:25	Reserved	RO	0	EC - Link Capabilities 2 Register [31:00] - reserved.
31	Device Readiness Status Supported	RO	0	DRS Supported. When set, indicates support for the optional Device Readiness Status (DRS) capability. Must be set in downstream ports that support DRS. Must be set in downstream ports that support FRS. For upstream ports that support DRS, it is strongly recommended that this bit be set in function 0. For all other functions associated with an upstream port, this bit must be clear. Must be clear in functions that are not associated with a port. RsvdP in all other functions.

4.5.4.12 EC - Link Control/Status 2 Register

The Link Control/Status 2 register is used to control extended link functions.

Mnemonic

Address Offset 0x078

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	cfgrco_ec30_targetlinkspeed(3:0)	RWS	0x4	EC - Link Control 2 Register [3:0] - Target Link Speed. The reset value of this signals defaults to the value of this signal: pi_sys_ec0c_maxlinkspeed.
04	cfgrco_ec30_entercompliance	RWHS	0	EC - Link Control 2 Register [4] - Enter Compliance. 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. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clrentercompliance asserts to '1'.
05	cfgrco_ec30_hwautospeed_disable	RWS	0	EC - Link Control 2 Register [5] - hardware Autonomous Speed Disable. When set, this bit disables hardware from changing the link speed for device specific reasons other than attempting to correct unreliable link operation by reducing link speed. Initial transition to the highest supported common link speed is not blocked by this bit.
06	ec30_sel_deemphasis	RO	0	EC - Link Control 2 Register [6] - Selectable Deemphasis. Hard-coded to '0'. Value is driven from the 'REGB' logic instead of the CFG core.

Bit	Field Mnemonic	Type	Reset Value	Description
09:07	cfgrco_ec30_transmitmargin(2:0)	RWHS	0	EC - Link Control 2 Register [09:07] - Transmit Margin. Transmit Margin – This field controls the value of the non-deemphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see *Functional Description* on page 29 for details of how the Transmitter voltage level is determined in various states). Encodings: • 000b Normal operating range • 001b-111b Not all encodings are required to be implemented. For a Multi-Function device associated with an upstream port, the field in function 0 is of type RWS, and only function 0 controls the component's link behavior. In all other functions of that device, this field is of type RsvdP. Default value of this field is 000b. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 000b. This field is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clrtransmitmargin asserts to '1'.
10	cfgrco_ec30_entercompliance	RWS	0	EC - Link Control 2 Register [10] - Enter Modified Compliance. When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate.
11	cfgrco_ec30_compliancesos	RWS	0	EC - Link Control 2 Register [11] - Compliance SOS. When set, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns.

Bit	Field Mnemonic	Type	Reset Value	Description
15:12	cfgrco_ec30_compliancedeemphasis(3:0)	RWS	0	EC - Link Control 2 Register [12] - Compliance Preset/Deemphasis. Compliance Preset/Deemphasis: For 8.0 GT/s and higher data rate: This field sets the Transmitter Preset in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Results are undefined if a reserved preset encoding is used when entering Polling. Compliance in this way. For 5.0 GT/s Data Rate: This field sets the deemphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Defined Encodings are: • 0001b -3.5 dB • 0000b -6 dB When the link is operating at 2.5 GT/s, the setting of this field has no effect. Components that support only 2.5 GT/s speed are permitted to hardwire this field to 0000b. For a Multi-Function device associated with an upstream port, the field in function 0 is of type RWS, and only function 0 controls the component's link behavior. In all other functions of that device, this field is of type RsvdP. The default value of this field is 0000b. This field is intended for debug and compliance testing purposes. System firmware and software is allowed to modify this field only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value.
16	tldlpo_dl_ec30_currentdeemphasis	ROH	0	EC - Link Status 2 Register [0] - Current De-emphasis Level. When the link is operating at 5.0 GT/s speed, this bit reflects the level of deemphasis. Encodings: • 1b: -3.5 dB • 0b: -6 dB The value in this bit is undefined when the link is not operating at 5.0 GT/s speed. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For components that support speeds greater than 2.5 GT/s, Multi-Function devices associated with an upstream port must report the same value in this field for all functions of the port. With the M-PCIe PHY, this feature is unused and this bit is hard-wired to 0b.

Bit	Field Mnemonic	Type	Reset Value	Description
17	tldlpo_dl_ec30_eqcomplete	RW1CHS	0	EC - Link Status 2 Register [1] - Equalization 8.0 GT/s Complete. Equalization Complete – When set to 1b, this bit indicates that the Transmitter Equalization procedure at the 8.0 GT/s data rate has completed. The default value of this bit is 0b. For Multi-Function Upstream Port, this bit must be implemented in function 0 and RsvdZ in other functions. Components that only support speeds below 8.0 GT/s are permitted to hardwire this bit to 0b. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clreqstatus asserts to '1'.
18	tldlpo_dl_ec30_eqphase1	RW1CHS	0	EC - Link Status 2 Register [2] - Equalization 8.0 GT/s Phase 1 Successful. This signal is asserted to '1' when Phase 1 of an 8.0 GT/s equalization process is completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clreqstatus asserts to '1'.
19	tldlpo_dl_ec30_eqphase2	RW1CHS	0	EC - Link Status 2 Register [3] - Equalization 8.0 GT/s Phase 2 Successful. This signal is asserted to '1' when Phase 2 of an 8.0 GT/s equalization process is completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clreqstatus asserts to '1'.
20	tldlpo_dl_ec30_eqphase3	RW1CHS	0	EC - Link Status 2 Register [4] - Equalization 8.0 GT/s Phase 3 Successful. This signal is asserted to '1' when Phase 3 of an 8.0 GT/s equalization process is completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clreqstatus asserts to '1'.
21	tldlpo_dl_ec30_seteqrequest	RW1CHS	0	EC - Link Status 2 Register [5] - Link Equalization Request 8 GT/s Status This bit is set by hardware to request the 8.0 GT/s Link equalization process to be performed on the Link. Note: This value will clear to zeros when the internal signal tldlpo_dl_ec30_clreqrequest asserts to '1'.
22	tldlpo_dl_ec30_retimerpresence	RW1CHS	0	EC - Link Status 2 Register [6] - Retimer Presence Detected. When set to 1b, this bit indicates that a Retimer was present during the most recent link negotiation. The default value of this bit is 0b. This bit is required for ports that have the Retimer Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. Ports that have the Retimer Presence Detect Supported bit set to 0b are permitted to hardwire this bit to 0b.

Bit	Field Mnemonic	Type	Reset Value	Description
23	tldlpo_dl_ec30_tworetimerpresence (vA4.2) tldlpo_dl_ec30_eqcomplete_16 (vA4.1, moved to P16 space)	RW1CHS	0	EC - Link Status 2 Register [7] - Two Retimers Presence Detected. When set to 1b, this bit indicates that two Retimers were present during the most recent Link negotiation. The default value of this bit is 0b. This bit is required for Ports that have the Two Retimers Presence Detect Supported bit of the Link Capabilities 2 register set to 1b. Ports that have the Two Retimers Presence Detect Supported bit set to 0b are permitted to hardwire this bit to 0b.
24	Reserved (vA4.2) tldlpo_dl_ec30_eqphase1_16 (vA4.1, moved to P16 space)	RO	0	EC - Link Status 2 Register [8] - Reserved.
25	Reserved (vA4.2) tldlpo_dl_ec30_eqphase2_16 (vA4.1, moved to P16 space)	RO	0	EC - Link Status 2 Register [9] - Reserved.
26	Reserved (vA4.2) tldlpo_dl_ec30_eqphase3_16 (vA4.1, moved to P16 space)	RO	0	EC - Link Status 2 Register [10] - Reserved.
27	Reserved (vA4.2) tldlpo_dl_ec30_seteqrequest_16 (vA4.1, moved to P16 space)	RO		EC - Link Status 2 Register [11] - Reserved.
31:28	Reserved	RO	0	EC - Link Status 2 Register [15:12] - Reserved.

4.5.5 AER - Advanced Error Reporting Registers (0x100 to 0x144)

4.5.5.1 AER - PCI Express Enhanced Capability Header

This register contains PCI Express specific ID and version for advanced error reporting. This register also contains the pointer to the next capability structure.

Mnemonic

Address Offset 0x100

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	aer_ext_cap_id	RO	0x0001	AER - Extended Capability ID Register [15:00].
19:16	aer_ext_cap_ver	RO	0x1	AER - Extended Capability Version Register [03:00].
31:20	aer_next_cap_ptr	RO	0x148	AER - Extended Capability Version Register [15:04].

4.5.5.2 AER - Uncorrectable Error Status Register

This register reports error status of the individual uncorrectable error sources (non-fatal and fatal) on a PCI Express device.

Mnemonic

Address Offset 0x104

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	Reserved	RW1CHS	0	AER - Data Link Protocol Error Status.
4	tldlpo_dl_ec08_dllpe	RW1CHS	0	AER - Surprise Down Error Status.
5	tldlpo_dl_ec08_surprisedown	RO	0	Reserved.
11:06	Reserved	RW1CHS	0	AER - Poisoned TLP Status.
12	pi_err_rcv_poisoned_tlp	RW1CHS	0	AER - Flow Control Protocol Error Status.
13	tldlpo_tl_ec08_fcpe	RW1CHS	0	AER - Completion Timeout Status.
14	pi_err_sig_cmpl_timeout	RW1CHS	0	AER - Completer Abort Status.
15	pi_err_rcv_cpl_ca	RW1CHS	0	AER - Unexpected Completion Status.
16	pi_err_rcv_unexp_cpl	RW1CHS	0	AER - Receiver Overflow Status.
17	pi_err_rcv_overflow	RW1CHS	0	AER - Malformed TLP Status.
18	pi_err_rcv_malf_tlp	RW1CHS	0	AER - ECRC Error Status.
19	pi_err_rcv_ecrc_err	RW1CHS	0	AER - Unsupported Request Error Status.
20	pi_err_rcv_unsup_req	RO	0	Reserved.
31:21	Reserved	RW1CHS	0	AER - Data Link Protocol Error Status.

4.5.5.3 AER - Uncorrectable Error Mask Register

This register masks the individual uncorrectable error reporting sources (non-fatal and fatal) on a PCI Express device. A masked error will not be logged in the Header Log Register, does not update the First Error Pointer in the Capabilities and Control Register, and is not reported to the root complex.

Mnemonic

Address Offset 0x108

Bit	Field Mnemonic	Type	Reset Value	Description
03:00	Reserved	RO	0	Reserved.
4	tldlpo_dl_ec08_dllpe	RWS	0	AER - Data Link Protocol Error Mask.
5	tldlpo_dl_ec08_surprisedown	RWS	0	AER - Surprise Down Error Mask.
11:06	Reserved	RO	0	Reserved.
12	pi_err_rcv_poisoned_tlp	RWS	0	AER - Poisoned TLP Mask.
13	tldlpo_tl_ec08_fcpe	RWS	0	AER - Flow Control Protocol Error Mask.
14	pi_err_sig_cmpl_timeout	RWS	0	AER - Completion Timeout Mask.
15	pi_err_rcv_cpl_ca	RWS	0	AER - Completer Abort Mask.
16	pi_err_rcv_unexp_cpl	RWS	0	AER - Unexpected Completion Mask.
17	pi_err_rcv_overflow	RWS	0	AER - Receiver Overflow Mask.
18	pi_err_rcv_malf_tlp	RWS	0	AER - Malformed TLP Mask.
19	pi_err_rcv_ecrc_err	RWS	0	AER - ECRC Error Mask.
20	pi_err_rcv_unsup_req	RWS	0	AER - Unsupported Request Error Mask.
31:21	Reserved	RO	0	Reserved.

4.5.5.4 AER - Correctable Error Status Register

This register reports error status of the individual correctable error sources on a PCI Express device.

Mnemonic

Address Offset 0x110

Bit	Field Mnemonic	Type	Reset Value	Description
0	tldlpo_dl_ec08_receivererror	RW1CHS	0	AER - Receiver Error Status.
05:01	Reserved	RO	0	Reserved.
6	tldlpo_dl_ec08_badtlp	RW1CHS	0	AER - Bad TLP Status.
7	tldlpo_dl_ec08_baddllp	RW1CHS	0	AER - Bad DLLP Status.
8	tldlpo_dl_ec08_replayrollover	RW1CHS	0	AER - REPLAY_NUM Rollover Status.
11:09	Reserved	RO	0	Reserved.
12	tldlpo_dl_ec08_replaytimeout	RW1CHS	0	AER - Replay Timer Timeout Status.
31:13	Reserved	RO	0	Reserved.

4.5.5.5 AER - Capabilities and Control Register

The AER capabilities and control register contains the first error pointer register and also controls and reports the state of the ECRC generation capabilities of the PCI Express devices.

Mnemonic

Address Offset 0x118

Bit	Field Mnemonic	Type	Reset Value	Description
04:00	aer_capctl_first_error_ptr	ROS	0	AER - First Error Pointer Register. The First Error Pointer Register identifies the bit position of the first error reported in the uncorrectable Error Status Register. If multiple uncorrectable errors occur on the same PCLK 250 clock cycle, the PCIEXG2CFG core will prioritize errors in the following priority. s_uncorr_err_vec(4) ≤ tldlpo_dl_ec08_dllpe; s_uncorr_err_vec(5) ≤ tldlpo_dl_ec08_surprisedown; -- s_uncorr_err_vec(12) ≤ pi_err_rcv_poisoned_tlp; s_uncorr_err_vec(13) ≤ tldlpo_tl_ec08_fcpe; s_uncorr_err_vec(14) ≤ pi_err_sig_cmpl_timeout; s_uncorr_err_vec(15) ≤ pi_err_rcv_cpl_ca; s_uncorr_err_vec(16) ≤ pi_err_rcv_unexp_cpl; s_uncorr_err_vec(17) ≤ pi_err_rcv_overflow; s_uncorr_err_vec(18) ≤ pi_err_rcv_malf_tlp; s_uncorr_err_vec(19) ≤ pi_err_rcv_ecrc_err; s_uncorr_err_vec(20) ≤ pi_err_rcv_unsup_req;
5	aer_capctl_ecrc_gen_cap	RO	0x1	AER - ECRC Generation Capable.
6	cfgrco_aer18_ecrc_gen	RWS	0	AER - ECRC Generation Enable.
7	aer_capctl_ecrc_chk_cap	RO	0x1	AER - ECRC Check Capable.
8	cfgrco_aer18_ecrc_check	RWS	0	AER - ECRC Check Enable.
31:09	Reserved	RO	0	Reserved.

4.5.5.6 AER - Header Log Register #1

The header log registers capture the TLP header of the packet corresponding to the first logable non-masked uncorrectable error.

Mnemonic

Address Offset 0x11C

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	pi_aer28to1c_header_log(31:00)	ROHS	0	AER - Header Log Register #1. Contains TLP header bytes 0:3, where byte 3 is the least significant byte of the AER header log register. Note: Header bytes stored in reverse order, (31:00) = bytes 0,1,2,3. This is in accordance with PCIe specification.

4.5.5.7 AER - Header Log Register #2

The header log registers capture the TLP header of the packet corresponding to the first logable non-masked uncorrectable error.

Mnemonic

Address Offset 0x120

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	pi_aer28to1c_header_log(63:32)	ROHS	0	AER - Header Log Register #2. Contains TLP header bytes 4:7, where byte 7 is the least significant byte of the AER header log register. Note: Header bytes stored in reverse order, (63:32) = bytes 4,5,6,7. This is in accordance with PCIe specification.

4.5.5.8 AER - Header Log Register #3

The header log registers capture the TLP header of the packet corresponding to the first logable non-masked uncorrectable error.

Mnemonic

Address Offset 0x124

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	pi_aer28to1c_header_log(95:64)	ROHS	0	AER - Header Log Register #3. Contains TLP header bytes 8:11, where byte 11 is the least significant byte of the AER header log register. Note: Header bytes stored in reverse order, (95:64) = bytes 8,9,10,11. This is in accordance with PCIe specification.

4.5.5.9 AER - Header Log Register #4

The header log registers capture the TLP header of the packet corresponding to the first logable non-masked uncorrectable error.

Mnemonic

Address Offset 0x128

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	pi_aer28to1c_header_log(127:96)	ROHS	0	AER - Header Log Register #4. Contains TLP header bytes 12:15, where byte 15 is the least significant byte of the AER header log register. The value in this register is undefined for 12 byte TLP headers that are logged. Note: Header bytes stored in reverse order, (127:96) = bytes 12,13,14,15. This is in accordance with PCIe specification.

4.5.5.10 AER - Root Error Status Register

The AER root status register reports the status of all errors (correctable, non-fatal, and fatal) in the PCI Express hierarchy of the Root Port device.

Mnemonic

Address Offset 0x130

Bit	Field Mnemonic	Type	Reset Value	Description
0	pi_aer30_corr_error_msg	RW1CHS	0	AER - Root Error Status Register [0] - ERR_CORR Received.
1	s_mult_err_cor	RW1CHS	0	AER - Root Error Status Register [1] - Multiple ERR_CORR Received.
2	*pi_aer30_non_fatal_error_msg or* pi_aer30_fatal_error_msg	RW1CHS	0	AER - Root Error Status Register [2] - NON_FATAL/FATAL Received.
3	s_mult_err_fat	RW1CHS	0	AER - Root Error Status Register [3] - Multiple NON_FATAL/FATAL Received.
4	s_first_is_fat	RW1CHS	0	AER - Root Error Status Register [4] - First Uncorrectable Fatal.
5	pi_aer30_non_fatal_error_msg	RW1CHS	0	AER - Root Error Status Register [5] - Non-Fatal Error Received.
6	pi_aer30_fatal_error_msg	RW1CHS	0	AER - Root Error Status Register [6] - Fatal Error Received.
31:07	Reserved	RO	0	Reserved.

4.5.5.11 AER - Error Source Identification Register

The AER error source identification register reports the source (Requester ID) of the first correctable and uncorrectable (non-fatal and fatal) errors reported in the AER root status register. Error messages from the port are given priority over internal errors if they occur in the same PCLK cycle. The assumption is that message based errors received on the port must have occurred earlier in time.

Mnemonic

Address Offset 0x134

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	pi_aer34_source_id_msg(15:0)	ROS	0	AER - ERR_CORR Source Identification Register [15:00].
31:16	pi_aer34_source_id_msg(15:0)	ROS	0	AER - ERR_FATAL/NONFATAL Source Identification Register [15:00].

4.5.6 SEC - Secondary PCI Express Extended Capability Registers (0x148 to 0x170)

The Secondary PCI Express Extended Capabilities register space was added in the PCIe Gen 3 specification. It is primarily used for link equalization controls and parameters.

4.5.6.1 SEC - PCI Express Extended Capability Header Register

PCI Express register.

Mnemonic

Address Offset 0x148

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	sec_ext_cap_id(15:0)	RO	0x0019	SEC - Extended Capability ID Register [15:00].
19:16	sec_ext_cap_ver(3:0)	RO	0x1	SEC - Extended Capability Version Register [03:00].
31:20	sec_next_cap_ptr(11:0)	RO	0x174	SEC - Extended Capability Version Register [15:04]. This value points to the P16 capability structure. (vA4.2).

4.5.6.2 SEC - Link Control 3 Register

PCI Express register.

Mnemonic

Address Offset 0x14C

Bit	Field Mnemonic	Type	Reset Value	Description
0	cfgrco_sec04_performeq	RWH	0	Perform Equalization, when this bit is 1b and a 1b is written to the Retrain Link bit with the Target Link Speed field set to 8.0 GT/s, the downstream port must perform link equalization. This bit is automatically cleared to '0' by the TLDLP core hardware when equalization completes via the signal: tldlpo_dl_sec04_clrperformeq. This function is typically not required for normal operation because equalization is automatically performed during link initialization and training. This function is used for debug or workaround purposes.
31:01	Reserved	RO	0	Reserved.

4.5.7 P16 - Physical Layer 16 GT/s Extended Capability Registers (0x174 to 0x19C)

The Physical Layer 16 GT/s Extended Capabilities register space was added in the PCIe Gen 4 specification. It is primarily used for link equalization controls and parameters.

4.5.7.1 P16 - Physical Layer 16 GT/s Extended Capability Header Register

PCI Express register.

Mnemonic

Address Offset 0x174

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	p16_ext_cap_id(15:0)	RO	0x0026	P16 - Extended Capability ID Register [15:00].
19:16	p16_ext_cap_ver(3:0)	RO	0x1	P16 - Extended Capability Version Register [03:00].
31:20	p16_next_cap_ptr(11:0)	RO	0x1A0	P16 - Extended Capability Version Register [15:04]. This value points to the LMR capability structure.

4.5.7.2 P16 - 16 GT/s Capabilities Register

PCI Express register.

Mnemonic

Address Offset 0x178

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	Reserved	RO	0	Reserved.

4.5.7.3 P16 - 16 GT/s Control Register

PCI Express register.

Mnemonic

Address Offset 0x17C

Bit	Field Mnemonic	Type	Reset Value	Description
31:00	Reserved	RO	0	Reserved.

4.5.7.4 P16 - 16 GT/s Status Register

PCI Express register.

Mnemonic

Address Offset 0x180

Bit	Field Mnemonic	Type	Reset Value	Description
00	tldlpo_dl_p160c_eqcomplete	RW1CHS	0	P16 – 16 GT/s Status Register [00] – Equalization 16.0 GT/s Complete. When set, this bit indicates that the 16.0 GT/s Transmitter Equalization procedure has completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_p160c_clreqstatus asserts to '1'.
01	tldlpo_dl_p160c_eqphase1	RW1CHS	0	P16 – 16 GT/s Status Register [01] – Equalization 16.0 GT/s Phase 1 Successful. When set to 1b, this bit indicates that Phase 1 of the 16.0 GT/s Transmitter Equalization procedure has successfully completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_p160c_clreqstatus asserts to '1.
02	tldlpo_dl_p160c_eqphase2	RW1CHS	0	P16 – 16 GT/s Status Register [02] – Equalization 16.0 GT/s Phase 2 Successful. When set to 1b, this bit indicates that Phase 2 of the 16.0 GT/s Transmitter Equalization procedure has successfully completed. Note: This value will clear to zeros when the internal signal 31:05 tldlpo_dl_p160c_clreqstatus asserts to '1'.
03	tldlpo_dl_p160c_eqphase3	RW1CHS	0	P16 – 16 GT/s Status Register [03] – Equalization 16.0 GT/s Phase 3 Successful. When set to 1b, this bit indicates that Phase 3 of the 16.0 GT/s Transmitter Equalization procedure has successfully completed. Note: This value will clear to zeros when the internal signal tldlpo_dl_p160c_clreqstatus asserts to '1'.
04	tldlpo_dl_p160c_seteqrequest	RW1CHS	0	P16 – 16 GT/s Status Register [04] – Link Equalization Request 16.0 GT/s. This bit is set by hardware to request the 16.0 GT/s link equalization process to be performed on the link. Note: This value will clear to zeros when the internal signal tldlpo_dl_p160c_clreqrequest asserts to '1'.
31:05	Reserved	RO	0	Reserved.

4.5.7.5 P16 - 16 GT/s Local Data Parity Mismatch Status Register

PCI Express register.

Mnemonic

Address Offset 0x184

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	tldlpo_dl_p1610_locparstatus	RW1CHS	0	Local Data Parity Mismatch Status. Each bit indicates if the corresponding lane detected a data parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding lane number.
31:16	Reserved	RO	0	Reserved.

4.5.7.6 P16 - 16 GT/s First Retimer Data Parity Mismatch Status Register

PCI Express register.

Mnemonic

Address Offset 0x188

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	ldlpo_dl_p1614_rt1parstatus	RW1CHS	0	First Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding lane detected a data parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding lane number. The value of this field is undefined when no Retimers are present.
31:16	Reserved	RO	0	Reserved.

4.5.7.7 P16 - 16 GT/s Second Retimer Data Parity Mismatch Status Register

PCI Express register.

Mnemonic

Address Offset 0x18C

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	tldlpo_dl_p1618_rt2parstatus	RW1CHS	0	Second Retimer Data Parity Mismatch Status. Each bit indicates if the corresponding lane detected a data parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding lane number. The value of this field is undefined when no Retimers are present or only one Retimer is present.
31:16	Reserved	RO	0	Reserved.

4.5.8 LMR - Lane Margining at the Receiver Extended Capability Registers (0x1A0 to 0x1E4)

The Lane Margining at the Receiver Extended Capabilities register space was added in the PCIe Gen 4 specification. It is primarily used for link equalization controls and parameters.

4.5.8.1 LMR - Margining Extended Capability Header Register

PCI Express register.

Mnemonic

Address Offset 0x1A0

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	lmr_ext_cap_id(15:0)	RO	0x0027	LMR - Extended Capability ID Register [15:00].
19:16	lmr_ext_cap_ver(3:0)	RO	0x1	LMR - Extended Capability Version Register [03:00].
31:20	lmr_next_cap_ptr(11:0)	RO	0x1E8	LMR - Extended Capability Version Register [15:04]. This value points to the DLF capability structure.

4.5.8.2 LMR - Margining Port Capabilities/Status Register

PCI Express register.

Mnemonic

Address Offset 0x1A4

Bit	Field Mnemonic	Type	Reset Value	Description
00	margin_uses_drvr_sw	RO	1	LMR – Margining Port Capabilities Register [00] - Margining uses driver software. If set, indicates that margining is partially implemented using device driver software. Margining Software Ready indicates when this software is initialized. If clear, margining does not require device driver software. In this case the value read from Margining Software Ready is undefined. This is hard-wired to '1'.
15:01	Reserved	RO	0	LMR – Margining Port Capabilities Register [15:01] - Reserved.
16	margining_ready	RO	0	LMR – Margining Port Status Register [00] - Margining Ready. Indicates when the margining feature is ready to accept margining commands. Behavior is undefined if this bit is clear and, for any lane, any of the Receiver Number, Margin Type, Usage Model, or Margin Payload fields are written. If Margining uses driver software is set, Margining Ready must be set no later than 100 ms after the later of Margining software ready becoming set or the link training to 16.0 GT/s. If Margining uses driver software is clear, Margining Ready must be set no later than 100 ms after the link trains to 16.0 GT/s. This is hard-wired to '0'.
17	margining_ready	RO	0	LMR – Margining Port Status Register [01] - Margining Software Ready. When Margining uses driver software is set, then this bit, when set, indicates that the required software has performed the required initialization. The value of this bit is undefined if Margining users driver software is clear. The default value of this bit is implementation specific. This is hard-wired to '0'.
31:18	Reserved	RO	0	Reserved.

4.5.8.3 LMR - Margining Lane Control/Status Register # 1

PCI Express register.

Mnemonic

Address Offset 0x1A8

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.4 LMR - Margining Lane Control/Status Register # 2

PCI Express register.

Mnemonic

Address Offset 0x1AC

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.5 LMR - Margining Lane Control/Status Register # 3

PCI Express register.

Mnemonic

Address Offset 0x1B0

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.6 LMR - Margining Lane Control/Status Register # 4

PCI Express register.

Mnemonic

Address Offset 0x1B4

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.7 LMR - Margining Lane Control/Status Register # 5

PCI Express register.

Mnemonic

Address Offset 0x1B8

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.8 LMR - Margining Lane Control/ Status Register # 6

PCI Express register.

Mnemonic

Address Offset 0x1BC

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.9 LMR - Margining Lane Control/Status Register # 7

PCI Express register.

Mnemonic

Address Offset 0x1C0

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.10 LMR - Margining Lane Control/Status Register # 8

PCI Express register.

Mnemonic

Address Offset 0x1C4

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.11 LMR - Margining Lane Control/Status Register # 9

PCI Express register.

Mnemonic

Address Offset 0x1C8

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.12 LMR - Margining Lane Control/Status Register # 10

PCI Express register.

Mnemonic

Address Offset 0x11CC

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.13 LMR - Margining Lane Control/Status Register # 11

PCI Express register.

Mnemonic

Address Offset 0x1D0

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.14 LMR - Margining Lane Control/Status Register # 12

PCI Express register.

Mnemonic

Address Offset 0x1D4

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.15 LMR - Margining Lane Control/Status Register # 13

PCI Express register.

Mnemonic

Address Offset 0x1D8

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.16 LMR - Margining Lane Control/Status Register # 14

PCI Express register.

Mnemonic

Address Offset 0x1DC

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.17 LMR - Margining Lane Control/Status Register # 15

PCI Express register.

Mnemonic

Address Offset 0x1E0

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.8.18 LMR - Margining Lane Control/Status Register # 16

PCI Express register.

Mnemonic

Address Offset 0x1E4

Bit	Field Mnemonic	Type	Reset Value	Description
02:00	mlc_rec_num	RO	0	LMR – Margining Lane Control Register [02:00] – Receiver Number.
05:03	mlc_margin_type	RO	0b111	LMR – Margining Lane Control Register [05:03] – Margin Type.
06	mlc_usage_model	RO	0	LMR – Margining Lane Control Register [06] – Usage Model.
07	Reserved	RO	0	LMR – Margining Lane Control Register [07] - Reserved.
15:08	mlc_margin_payload	RO	0x9C	LMR – Margining Lane Control Register [15:08] – Margin Payload.
18:16	mls_rec_num	RO	0	LMR – Margining Lane Status Register [02:00] – Receiver Number Status.
21:19	mls_margin_type	RO	0	LMR – Margining Lane Status Register [05:03] – Margin Type Status.
22	mls_usage_model	RO	0	LMR – Margining Lane Status Register [06] – Usage Model Status.
23	Reserved	RO	0	LMR – Margining Lane Status Register [07] – Reserved.
31:24	mls_margin_payload	RO	0	LMR – Margining Lane Status Register [15:08] – Margin Payload Status.

4.5.9 DLF - Data Link Feature Extended Capability Registers (0x1E8 to 0x1F0)

The Data Link Feature Extended Capabilities register space was added in the PCIe Gen 4 specification. It is primarily used for data link layer specific feature controls and status.

4.5.9.1 DLF - Data Link Feature Extended Capability Header Register

PCI Express register.

Mnemonic

Address Offset 0x1E8

Bit	Field Mnemonic	Type	Reset Value	Description
15:00	dlf_ext_cap_id(15:0)	RO	0x0025	DLF - Extended Capability ID Register [15:00].
19:16	dlf_ext_cap_ver(3:0)	RO	0x1	DLF - Extended Capability Version Register [03:00].
31:20	dlf_next_cap_ptr(11:0)	RO	0	DLF - Extended Capability Version Register [15:04]. This value will always return zeros since this is the last capability structure supported.

4.5.9.2 DLF - Data Link Layer Capabilities Register

PCI Express register.

Mnemonic

Address Offset 0x1EC

Bit	Field Mnemonic	Type	Reset Value	Description
22:00	dlf_feature_support	RO	0	Local Data Link Feature Supported. This field contains the Feature Supported value used when this Port sends a Data Link Feature DLLP. Defined features are: • Bit 0 – Local Scaled Flow Control Supported. • This bit indicates that this Port supports the Scaled Flow Control Feature. • Bits 22:1 RsvdP. • Bits associated with features that this Port is capable of supporting are HwInit, defaulting to 1b. Other bits in this field are RsvdP.
30:01	Reserved	RO	0	Reserved.
31	pi_sys_dlf_dlfeatureenable	RO	0	Data Link Feature Exchange Enable. If set, this bit indicates that this Port will enter the DL_Feature negotiation state.

4.5.9.3 DLF - Data Link Feature Status Register

PCI Express register.

Mnemonic

Address Offset 0x1F0

Bit	Field Mnemonic	Type	Reset Value	Description
22:00	ldlpo_dl_dlf08_remfeaturesupported	RO	0	Remote Data Link Feature Supported. These bits indicate that the remote port supports the corresponding Data Link Feature. These bits capture all information from the Feature Supported field of the Data Link Feature DLLP even when this port does not support the corresponding feature. This field is cleared on entry to state DL_Inactive. Features currently defined are: • Bit 0 – Remote Scaled Flow Control Supported. • This bit indicates that the remote port supports the Scaled Flow Control Feature. • Bits 22:1 – undefined.
23	tldlpo_dl_dlf08_remfeaturesupported_23	RO	0	Remote Data Link Feature Ack. This bit indicates that the remote port has received this Port's Data Link Feature DLLP. This bit captures the Feature Ack bit of the Data Link Feature DLLP. This bit is cleared on entry to state DL_Inactive.
30:24	Reserved	RO	0	Reserved.
31	pi_sys_dlf_dlfeatureenable	RO	0	Remote Data Link Feature Supported Valid. This bit indicates that the port has received a Data Link Feature DLLP in state DL_Feature and that the Remote Data Link Feature Supported field is meaningful. This bit is cleared on entry to state DL_Inactive. TLDLP core does not currently support this feature, this bit is hardwired to '0'.

4.6 REGB Registers

The PCIe macros have an internal register block called REGB. This block contains all PCIe specific control and status registers as well as PCIe specific error trap registers. These registers are mapped directly into the chip address space. Refer to the address map in *PHB4 Unified Address Space* on page 56. The REGB registers range from 0x1800 to 0x1FFF.

The REGB address map contains:

- PBL core specific registers.
- PCIe specific strapping registers and TLDLP core registers.
- Error status bits and controls.

MMIO Load/Store accesses to the REGB registers that do not exist are handled by ignoring MMIO Stores and returning data of all 1's for MMIO Loads. All REGB configuration register accesses via the AIB or SCOM bus must have a length of 8 bytes and be aligned on an 8-byte address boundary.

4.6.1 REGB Internal Register Address Map

Table 4-35. REGB Internal Register Address Map (Page 1 of 3)

Offset	Description	Page
PBL Core Registers		
0x1800	PBL - Control Register.	239
0x1808	Reserved (unused, will return all 1's when read).	
0x1810	PBL - Timeout Control Register.	240
0x1818	Reserved (unused, will return all 1's when read).	
0x1820	PBL - Nonposted Tag Enable Register.	241
0x1828	Reserved (unused, will return all 1's when read).	
0x1830	PBL - NBW Compare/Mask Register.	242
0x1838	PBL - SYS_LINK_INIT Register.	243
0x1840	PBL - Buffer Status Register.	245
0x1848 to 0x18F8	Reserved (unused, will return all 1's when read).	
0x1900	PBL - Error Status Register.	245
0x1908	PBL - First Error Status Register.	249
0x1910	PBL - Error Injection Register.	250
0x1918	Reserved (unused, will return all 1's when read).	
0x1920	PBL - Error INF Enable Register.	251
0x1928	PBL - Error ERC Enable Register.	251
0x1930	PBL - Error FAT Enable Register.	251
0x1938	Reserved (unused, will return all 1's when read).	
0x1940	PBL - Error Log Register 0.	252
0x1948	PBL - Error Log Register 1.	252
0x1950	PBL - Error Status Mask Register.	252

Table 4-35. REGB Internal Register Address Map (Page 2 of 3)

Offset	Description	Page
0x1958	PBL - First Error Status Mask Register.	252
0x1960 to 0x19F8	Reserved (unused, will return all 1's when read).	
PCIe Stack Registers		
0x1A00	PCIe - System Configuration Register.	254
0x1A08	PCIe - Bus Number Register.	256
0x1A10	PCIe - Core Reset Register.	257
0x1A18	PCIe - DLP Strapping Register.	258
0x1A20	PCIe - Hot Plug Status Register.	259
0x1A28	Reserved (unused, will return all 1's when read).	
0x1A30	PCIe - Link Management Register.	260
0x1A38	Reserved (unused, will return all 1's when read).	
0x1A40	PCIe - DLP Training Control Register.	261
0x1A48	PCIe - DLP Loopback Status Register.	263
0x1A50 to 0x1A68	Reserved (unused, will return all 1's when read).	
0x1A70	PCIe - DLP Fault Isolation Register.	264
0x1A78	PCIe - DLP Control Register.	265
0x1A80	PCIe - DLP Trace Read/Write Control Register.	270
0x1A88	PCIe - DLP Trace Read Data Register.	270
0x1A90 to 0x1A98	Reserved (unused, will return all 1's when read).	
0x1AA0	PCIe - DLP Error Log Register 1.	271
0x1AA8	PCIe - DLP Error Log Register 2.	271
0x1AB0	PCIe - DLP Error Status Register.	271
0x1AB8	PCIe - DLP Error Counters Register.	272
0x1AC0	Reserved (unused, will return all 1's when read).	
0x1AC8	PCIe - DLP Error Inject Control Register.	273
0x1AD0	PCIe - DLP LANEEQCONTROL Register 0 (Gen 3, 8Gbps).	275
0x1AD8	PCIe - DLP LANEEQCONTROL Register 1 (Gen 3, 8Gbps).	276
0x1AE0	PCIe - DLP LANEEQCONTROL Register 2 (Gen 3, 8Gbps).	277
0x1AE8	PCIe - DLP LANEEQCONTROL Register 3 (v3, 8Gbps).	278
0x1AF0	PCIe - DLP LANEEQCONTROL2 Register 0 (Gen 4, 16Gbps) (vA4.1). PCIe - DLP P1620-LANEEQCONTROL Register 0 (Gen 4, 16Gbps) (vA4.2).	279
0x1AF8	PCIe - DLP LANEEQCONTROL2 Register 1 (Gen 4, 16Gbps) (vA4.1). PCIe - DLP P1620-LANEEQCONTROL Register 1 (Gen 4, 16Gbps) (vA4.2).	280
0x1B00	PCIe - DLP LANEEQCONTROL2 Register 2 (Gen 4, 16Gbps) (vA4.1). Reserved (unused, will return all 1's when read) (vA4.2).	

Table 4-35. REGB Internal Register Address Map (Page 3 of 3)

Offset	Description	Page
0x1B08	PCIe - DLP LANEEQCONTROL2 Register 3 (Gen 4, 16Gbps) (vA4.1). Reserved (unused, will return all 1's when read) (vA4.2).	
0x1B10	PCIe - Cable Auxiliary Inputs Register. Reserved (unused, will return all 1's when read).	
0x1B18	PCIe - Cable Auxiliary Outputs Register. Reserved (unused, will return all 1's when read).	
0x1B20	PCIe - Trace Control Register.	281
0x1B28	Reserved (unused, will return all 1's when read).	
0x1B30	PCIe - Misc. Strapping Register.	284
0x1B38 to 0x1BF8	Reserved(unused, will return all 1's when read).	
Error Registers		
0x1C00	REGB Error Status Register.	285
0x1C08	REGB First Error Status Register.	289
0x1C10	REGB Error Injection Register.	290
0x1C18	Reserved (unused, will return all 1's when read).	
0x1C20	REGB Error INF Enable Register.	290
0x1C28	REGB Error ERC Enable Register.	291
0x1C30	REGB Error FAT Enable Register.	291
0x1C38	Reserved (unused, will return all 1's when read).	
0x1C40	REGB Error Log Register 0.	291
0x1C48	REGB Error Log Register 1.	292
0x1C50	REGB Error Status Mask Register.	292
0x1C58	REGB First Error Status Mask Register.	293
0x1C60 to 0x1FF8	Reserved (unused, will return all 1's when read).	

4.6.2 PBL Core Registers

4.6.2.1 PBL - Control Register

This register is specific to the PBL core block.

Mnemonic PBL_CTRL

Address Offset 0x1800

Bit	Field Mnemonic	Type	Reset Value	Description
0	pbl_cpl_ld_flush_enable	RW	1	PBL: PCIe Completion Table Linkdown Flush Enable. When enabled (='1') and the PCIe link is down, the PCIe Completion Table (PCT) is cleared for all outstanding PCIe nonposted requests. An error response is returned to the BLIF requester for all outstanding requests and the nonposted PCIe tags are released and allowed to be re-used for new outbound nonposted requests. No error will be reported by the PBL for the releasing of the PCT entries. This is a silent discard. When disabled (='0') and the PCIe link is down, the pending PCIe completions will not be cleared from the PCT (unless a timeout occurs and the timeout flush is enabled).
1	pbl_cpl_to_flush_enable	RW	1	PBL: PCIe Completion Timeout Flush Enable. When enabled (= '1') and a completion timeout is detected for an outstanding PCIe nonposted request, an error response is returned to the BLIF requester and the nonposted PCIe tag is released and allowed to be re-used for new outbound nonposted requests. When disabled (= '0') and a completion timeout is detected for an outstanding PCIe nonposted request, the PBL will continue report the timeout error but continues to wait for the PCIe response. It will not return an error response to BLIF and will not free the PCIe nonposted tag for reuse.
02:15	Reserved	RW	0	Reserved, but implemented register(s) for future use.
16:63	Reserved	RO	0	Reserved.

4.6.2.2 PBL - Timeout Control Register

This register is specific to the PBL core block.

Mnemonic PBL_TOCT

Address Offset 0x1810

Bit	Field Mnemonic	Type	Reset Value	Description
0	pbl_comp_to_en	RW	0	PBL: Application Defined Completion Timeout Enable PCIe specification defined Completion Timeout Value override control. When enabled (= '1') the PBL completion timer value is used for timing outstanding PCIe nonposted request completions and the PCIe specification defined completion timer value is bypassed. When disabled (= '0') the PCIe specification defined completion timer value is used for timing outstanding PCIe nonposted request completions.
02:07	pbl_comp_to_value(0:5)	RW	0x20	PBL: Application Defined Completion Timeout Value. This value is driven into the PCIe Completion timeout timer when the ap_comp_to_en is set, overriding the PCIe specification defined values. This allows more fine-grained timeout values than the broad ranges defined by PCI. Bit 2 = Timeout disable = '1'. Bits 3:7 = Timeout value. The configuration value of the timeout will always represent a minimum/maximum timeout range: • Minimum timeout value is: $2^{(\text{timeout value})}$ x 64ns. • Maximum timeout value is: $2^{(\text{timeout value})}$ x 96ns. Note: These equations do not change with the PCIe link speed. The equation is the same for all speeds. The internal logic automatically takes link speed into account.
08:09	Reserved	RW	0	Reserved, but implemented register(s) for future use.
10:15	pbl_out_trans_to_value(0:5)	RW	0x20	PBL: Outbound Transmission Timeout Value. This value controls the forward progress timer for outbound transactions in the PBL. All outbound transactions are timed from the moment they are presented at the output of the outbound buffer until they are pushed to the Replay buffer. Each type of transaction is time separately, meaning there are 3 individual timers, on each for posted, nonposted and completions. Bit 10 = Timeout disable = '1' Bits 11:15 = Timeout value The configuration value of the timeout will always represent a minimum/maximum timeout range: • Minimum timeout value is: $2^{(\text{timeout value})}$ x 64ns. • Maximum timeout value is: $2^{(\text{timeout value})}$ x 96ns. Note: These equations do not change with the PCIe link speed. The equation is the same for all speeds. The internal logic automatically takes link speed into account.
16:63	Reserved	RO	0	Reserved.

4.6.2.3 PBL - Nonposted Tag Enable Register

This register is specific to the PBL core block.

Mnemonic PBL_NPTAGEN

Address Offset 0x1820

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	NP_TAG_EN(0:7)	RW	0xFF	Directly controls the outbound nonposted tags issued by the PBL onto the PCIe link. Each bit controls and individual PCIe nonposted tag. '0' = Disabled, tag will not be issued on PCIe. '1' = Enabled, tag can be issued for a PCIe nonposted request. Bit 0 = Tag x"00" Bit 1 = Tag x"01" … Bit 7 = Tag x"07" Note: At least 1 tag must be enabled to allow nonposted requests to be issued.
08:63	Reserved	RO	0	Reserved.

4.6.2.4 PBL - NBW Compare/Mask Register

This register is specific to the PBL core block. This register supports the CAPI Pseudo-VC function in the PHB4. The PHB/Root Complex must decode a Non-Blocking Write (NBW) versus a normal write using address bits within the posted write command. This register specifies the compare and mask values to apply to inbound posted write addresses.

This is a firmware programmable register to match expected 16 bit value for NBW command.

- 16 bit compare value, CMP(63:48), 16 bit mask value, MSK(63:48).
- CMP and MSK values applied to upper 16 bits of TLP header address (addr(63:48)).
- First mask off address bits that do not contribute to compare, then compare.
- The final result will indicate a packet is a NBW='1' versus a normal write='0'.

Example:

```
CMP(63:48) = 0xAB00
MSK(63:48) = 0xFF00
Addr(63:48) = 0xAB13
and_mask(15:0) = (0xAB13 and 0xFF00) = 0xAB00
xor(15:0) = (0xAB00 xor 0xAB00) = 0x0000
NBW = not or_reduce(xor(15:0)) = ‘1’ (packet is a NBW)
```

Compare and Mask values chosen must match expected NBW bits driven by the CAPI device.

Mnemonic PBL_NBW_CMPM

Address Offset 0x1830

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	CMP(63:48)	RW	0	Compare Value.
16:31	MSK(63:48)	RW	0	Mask Value.
32:62	Reserved	RO	0	Reserved.
63	NBW Enable	RW	0	Set to '1' to enable NBW decoding. If '0', no NBW will be decoded and effectively the Pseudo-VC function is disabled.

4.6.2.5 PBL - SYS_LINK_INIT Register

This register is related to PCI Express functionality in the PCIe macro. The value for each field below should not be changed from their defaults without advanced knowledge of the consequences. The function will not be guaranteed if the defaults are changed.

This register defines the strap input buffer sizes attached to the PBL core. The PBL uses these sizes to specify PCIe credits after the link trains.

Note: The maximum allowed number of advertised header credits on PCIe is 127, not 128. Values above that indicate 128 header credits are actually advertised as 127 on the PCIe link.

Mnemonic PCIE_SYSLINIT

Address Offset 0x1838

Bit	Field Mnemonic	Type	Reset Value	Description
0	regbo_sys_link_init_en	RW	0b1	PCIe Link Credit Initialization Enable. When set to '1', the PBL core will use the strapping values in this register to specify the flow control credits for the PCIe link. If '0', flow control credits will never be negotiated and the link will not operate properly.
01:03	Reserved	RW	0	Reserved, but implemented register(s) for future use.
04:07	regbo_sys_link_init_rxch(0:3)	RW	0x0	RX Completion Header Credits (default: 0x0 = infinite). RX completion credits should always be infinite. Number of header credits encoding (same encoding for all header types): • x"0"; -- 0 = infinite • x"1"; -- 1 credit • x"2"; -- 2 credits • x"3"; -- 4 credits • x"4"; -- 8 credits • x"5"; -- 16 credits • x"6"; -- 32 credits • x"7"; -- 64 credits • x"8"; -- 128 credits Other values are undefined and results are not predictable.

Bit	Field Mnemonic	Type	Reset Value	Description
08:11	regbo_sys_link_init_rxcd(0:3)	RW	0x0	RX Completion Data Credits (default: 0x0 = infinite). RX completion credits should always be infinite. Number of data credits encoding (same encoding for all types): • x"0"; -- 0 = infinite • x"1"; -- 1 credit (16-byte) • x"2"; -- 2 credits (32-byte) • x"3"; -- 4 credits (64-byte) • x"4"; -- 8 credits (128-byte) • x"5"; -- 16 credits (256-byte) • x"6"; -- 32 credits (512-byte) • x"7"; -- 64 credits (1 KB) • x"8"; -- 128 credits (2 KB) • x"9"; -- 256 credits (4 KB) • x"A"; -- 512 credits (8 KB) • x"B"; -- 1024 credits (16 KB) • x"C"; -- 2048 credits (32 KB) • x"D"; -- 4096 credits (64 KB) Other values are undefined and results are not predictable.
12:15	regbo_sys_link_init_rxnh(0:3)	RW	0x8 (x16) 0x7 (x08)	RX Nonposted Header Credits (default: based on physical link width). x16 link = 0x8 = 128 credits. x08 link = 0x7 = 64 credits.
16:19	regbo_sys_link_init_rxph(0:3)	RW	0x8 (x16) 0x7 (x08)	RX Posted Header Credits (default: based on physical link width). x16 link = 0x8 = 128 credits. x08 link = 0x7 = 64 credits.
20:23	regbo_sys_link_init_rxpd(0:3)	RW	0xB (x16) 0xA (x08)	RX Posted Data Credits (default: based on physical link width). x16 link = 0xB = 1024 credits = 16KB. x08 link = 0xA = 512 credits = 8KB.
24:27	regbo_sys_link_init_txch(0:3)	RW	0x4	TX Completion Header Credits (default: 0x4 = 8 credits).
28:31	regbo_sys_link_init_txcd(0:3)	RW	0x6	TX Completion Header Credits (default: 0x6 = 32 credits = 512B).
32:35	regbo_sys_link_init_txnh(0:3)	RW	0x4	TX Nonposted Header Credits (default: 0x4 = 8 credits).
36:39	regbo_sys_link_init_txnd(0:3)	RW	0x2	TX Nonposted Header Credits (default: 0x2 = 2 credits = 32B).
40:43	regbo_sys_link_init_txph(0:3)	RW	0x4	TX Posted Header Credits (default: 0x4 = 8 credits).
44:47	regbo_sys_link_init_txpd(0:3)	RW	0x7	TX Posted Header Credits (default: 0x7 = 64 credits = 1KB).
48:63	Reserved	RO	0	Reserved.

4.6.2.6 PBL - Buffer Status Register

This register is specific to the PBL core block.

Mnemonic PBL_BUF_STAT

Address Offset 0x1840

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	pbl_rrb_status_ib_usage(0:15)	ROH	0	
16:19	Reserved	RO	0	Reserved.
20:31	pbl_rrb_status_ob_usage(0:11)	ROH	0	
32:63	Reserved	RO	0	Reserved.

4.6.2.7 PBL - Error Status Register

This register contains error status information for the PBL core.

Mnemonic PBL_ESR

Address Offset 0x1900

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	pb_err_p_fe_tlif_rx_par_e	RW1CHS	0	TLIF RX Parity Error Output, PCLK domain logic. Parity error detected on TLIF Receive interface.	Fatal
1	pb_err_p_fe_tlif_tx_par_e	RW1CHS	0	TLIF TX Parity Error Output, PCLK domain logic. Parity error detected on TLIF Transmit interface.	Fatal
2	pb_err_p_fe_blif_out_par_e	RW1CHS	0	BLIF outbound parity error.	Fatal
3	pb_err_p_fe_blif_in_par_e	RW1CHS	0	BLIF inbound parity error.	Fatal
4	pb_err_p_fe_int_par_e	RW1CHS	0	PBL internal logic parity error.	Fatal
5	pb_err_p_fe_toc_cred_e	RW1CHS	0	TOC credit error (parity and overflow/underflow).	Fatal
6	pb_err_p_fe_ocf_par_e	RW1CHS	0	OCF Interface Control Parity Error.	Fatal
7	pb_err_p_fe_ocf_prot_e	RW1CHS	0	OCF Interface Protocol Error.	Fatal
08:11	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
12	pb_err_p_fe_pct_erq_overflow_e	RW1CHS	0	PCT Error Queue Overflow Error.	Fatal
13	pb_err_p_fe_pct_erq_underflow_e	RW1CHS	0	PCT Error Queue Underflow Error.	Fatal

1. The INF class errors are a special error class. These errors are recorded in the PBL Error Status Registers, but they do not have any other side effects. The initialization sequence will force the INF enable bits for these bits to zero. For these errors, the PBL will send an error transaction over the BLIF interface to the ETU logic. The ETU logic will decode and handle the error in its logic.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
14	pb_err_p_fe_pct_onp_tags_rls_unused_e	RW1CHS	0	PCT observed a release of an unused Outbound Nonposted Tag error ITR dequeued an invalid tag!	Fatal
15	pb_err_p_fe_pct_onp_tags_used_notfree_e	RW1CHS	0	PCT observed an enqueue of an already used Outbound Nonposted Tag error OTD reused an in-flight tag!	Fatal
16	pb_err_p_fe_pct_onp_tags_used_unexp_e	RW1CHS	0	PBL internal PCT tag use error.	Fatal
17	pb_err_p_fe_bct_onp_tags_rls_unused_e	RW1CHS	0	BLIF Completion Table release of an unused tag (unexpected completion).	Fatal
18	pb_err_p_fe_bct_onp_tags_used_notfree_e	RW1CHS	0	BLIF Completion Table usage of used tag.	Fatal
19	pb_err_p_fe_ib_bct_rd_inv	RW1CHS	0	BLIF Completion Table Read Invalid Entry (unexpected Completion). The inbound Completion header buffer entry, when read from the IBCH pointed to a BCT entry that was not marked valid.	Fatal
20	pb_err_p_fe_ob_buffer_overflow_e	RW1CHS	0	Outbound Buffer Overflow Error.	Fatal
21	pb_err_p_fe_ob_buffer_underflow_e	RW1CHS	0	Outbound Buffer Underflow Error.	Fatal
22	pb_err_p_fe_ib_buffer_overflow_e	RW1CHS	0	Inbound Buffer Overflow Error.	Fatal
23	pb_err_p_fe_ib_buffer_underflow_e	RW1CHS	0	Inbound Buffer Underflow Error.	Fatal
24	pb_err_p_fe_ib_d_ecc_ue	RW1CHS	0	Inbound Data Buffer ECC Uncorrectable Error. One of the inbound data buffers (P/N/C) had an uncorrectable error on the read data when processing the TLP. Refer to the error log registers to determine which buffer was the cause.	Fatal
25	pb_err_p_fe_ib_h_ecc_ue	RW1CHS	0	Inbound Header Buffer ECC Uncorrectable Error One of the inbound header buffers (P/N/C) had an uncorrectable error on the read data when processing the header. Refer to the error log registers to determine which header buffer was the cause.	Fatal
26	pb_err_p_fe_ob_d_ecc_ue	RW1CHS	0	Outbound Data buffer ECC UE.	Fatal
27	pb_err_p_fe_ob_h_ecc_ue	RW1CHS	0	Outbound Header buffer ECC UE.	Fatal
28	pb_err_p_fe_ocf_ecc_ue	RW1CHS	0	OCF Interface Data Bus ECC UE.	Fatal
29:31	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
32	pb_err_p_fe_tx_pst_discard_e	RW1CHS	0	Outbound Posted Request discarded error indicator. An outbound posted request at the head of the queue was dropped and will not be transmitted over the PCIe link.	Fatal

1. The INF class errors are a special error class. These errors are recorded in the PBL Error Status Registers, but they do not have any other side effects. The initialization sequence will force the INF enable bits for these bits to zero. For these errors, the PBL will send an error transaction over the BLIF interface to the ETU logic. The ETU logic will decode and handle the error in its logic.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
33	pb_err_p_inf_tx_npst_discard_e	RW1CHS	0	Outbound Nonposted Request discarded error indicator. An outbound nonposted request at the head of the queue was dropped and will not be transmitted over the PCIe link. An error response is generated back to the BLIF requester.	INF[1]
34	pb_err_p_fe_nbw_tlp_e	RW1CHS	0	Inbound Non-Blocking Write TLP Error. Indicates a dropped inbound NBW TLP.	Fatal
35	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
36	pb_err_p_fe_pci_rcv_cpl_ca_e	RW1CHS	0	Completer Abort (CA) completion response received.	INF[1]
37	pb_err_p_fe_pci_rcv_cpl_crs_e	RW1CHS	0	Config Retry Status (CRS) completion response received.	INF[1]
38	pb_err_p_fe_pci_rcv_cpl_rsvd_e	RW1CHS	0	Reserved Completion response received. The response code does not match a known completion status value.	Fatal
39	pb_err_p_fe_pci_rcv_cpl_ur_e	RW1CHS	0	Unsupported Request Received Error. This is a summary error of all the individual Unsupported Request error log indicators for inbound TLP checks.	INF[1]
40	pb_err_p_fe_pci_rcv_ecrc_e	RW1CHS	0	ECRC error detected on inbound TLP (from PTL core).	Fatal
41	pb_err_p_fe_pci_rcv_malf_tlp_e	RW1CHS	0	Malformed TLP error. This is a summary error of all the individual malformed TLP error log indicators.	Fatal
42	pb_err_p_fe_pci_rcv_overflow_e	RW1CHS	0	Receiver overflow detected (from TLDLP core).	Fatal
43	pb_err_p_fe_pci_rcv_poisoned_tlp_e	RW1CHS	0	Received Inbound Poisoned TLP.	Fatal vA4.1 INF[1] vA4.2
44	pb_err_p_fe_pci_rcv_unexp_cpl_e	RW1CHS	0	Unexpected Completion Received Error. This is a summary error of all the individual unexpected completion error log indicators.	Fatal
45	pb_err_p_fe_pci_rcv_unsup_req_e	RW1CHS	0	Unsupported Request Received Error. This is a summary error of all the individual Unsupported Request error log indicators for inbound TLP checks.	INF[1]
46	pb_err_p_fe_pci_sig_cpl_abort_e	RW1CHS	0	Issued Completer Abort status outbound. PBL issued an outbound completion TLP with a completer abort ('CA') completion status.	INF[1]
47	pb_err_p_fe_pci_sig_cpl_timeout_e	RW1CHS	0	Completion Timeout Error. A timeout was detected for an outbound nonposted request which is waiting on a completion from PCI.	INF[1]

1. The INF class errors are a special error class. These errors are recorded in the PBL Error Status Registers, but they do not have any other side effects. The initialization sequence will force the INF enable bits for these bits to zero. For these errors, the PBL will send an error transaction over the BLIF interface to the ETU logic. The ETU logic will decode and handle the error in its logic.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
48	pb_err_p_fe_pci_sig_poisoned_tlp_e	RW1CHS	0	Signaled outbound poisoned TLP. PBL issued an outbound TLP with the poisoned bit ('EP') set in the header.	INF[1]
49:51	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
52	pb_err_p_inf_out_trans_to_pst_e	RW1CHS	0	Outbound Posted Request transmission timeout. Indicates the outbound posted request at the head of the posted queue was not able to be transmitted in the allotted time due to one of the following conditions: 1. Lack of PCIe credits for the request 2. Lack of Replay buffer space for the request. This is informational. The request is not dropped, it continues to wait for permission to proceed. This could indicate a credit return or DLL Ack problem from the PCIe link.	Fatal
53	pb_err_p_inf_out_trans_to_npst_e	RW1CHS	0	Outbound nonposted request transmission timeout. Indicates the outbound nonposted request at the head of the nonposted queue was not able to be transmitted in the allotted time due one of the following conditions: • Lack of PCIe credits for the request. • Lack of Replay buffer credits for the request. • Blocked by outbound Posted request due to PCIe ordering. This is informational. The request is not dropped, it continues to wait for permission to proceed. The nonposted timeout timer does not being if there are no available nonposted request tags to create the outbound request. The lack of request tags is timed via the completion timeout timers for previously issued nonposted requests.	Fatal
54	pb_err_p_inf_out_trans_to_cpl_e	RW1CHS	0	Outbound Completion transmission timeout. Indicates the outbound completion at the head of the completion queue was not able to be transmitted in the allotted time due to one of the following conditions: • Lack of PCIe credits for the request. • Lack of Replay buffer credits for the request. • Blocked by outbound posted request due to PCIe ordering. This is informational. The request is not dropped, it continues to wait for permission to proceed.	Fatal
55	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
56	pb_err_p_inf_ib_d_ecc_ce	RW1CHS	0	Inbound Data buffer ECC CE.	INF[1]
57	pb_err_p_inf_ib_h_ecc_ce	RW1CHS	0	Inbound Header buffer ECC CE.	INF[1]

1. The INF class errors are a special error class. These errors are recorded in the PBL Error Status Registers, but they do not have any other side effects. The initialization sequence will force the INF enable bits for these bits to zero. For these errors, the PBL will send an error transaction over the BLIF interface to the ETU logic. The ETU logic will decode and handle the error in its logic.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
58	pb_err_p_inf_ob_d_ecc_ce	RW1CHS	0	Outbound Data buffer ECC CE.	INF[1]
59	pb_err_p_inf_ob_h_ecc_ce	RW1CHS	0	Outbound Header buffer ECC CE.	INF[1]
60	pb_err_p_inf_ocf_ecc_ce	RW1CHS	0	OCF Interface Data Bus ECC CE.	INF[1]
61	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
62	PBL Bad Register Address Error	RW1CHS	0	PBL register access to non-implmented address/register in range 0x1800 to 0x1958.	Fatal
63	PBL Register Parity Error	RW1CHS	0	Parity Error detected in the PBL register set.	Fatal

1. The INF class errors are a special error class. These errors are recorded in the PBL Error Status Registers, but they do not have any other side effects. The initialization sequence will force the INF enable bits for these bits to zero. For these errors, the PBL will send an error transaction over the BLIF interface to the ETU logic. The ETU logic will decode and handle the error in its logic.

4.6.2.8 PBL - First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the PBL Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle.

A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic PBL_FESR

Address Offset 0x1908

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *PBL - Error Status Register* on page 245.	ROHS	0	Bits correspond to the error that occurred first.

4.6.2.9 PBL - Error Injection Register

This register controls error injection for the individual error bits defined in the Error Status Register.

Writing bits to a '1' in this register will inject the error corresponding to the bit that is written. The bits will automatically clear to '0' after the error is injected. The corresponding bit in the Error Status Register is set automatically when the error occurs.

Mnemonic PBL_EINJ

Address Offset 0x1910

Bit	Field Mnemonic	Type	Reset Value	Description
00:23	Reserved	RO	0	Reserved.
24	pb_err_p_fe_ib_d_ecc_ue	RWH	0	Inbound Data Buffer ECC Uncorrectable Error.
25	pb_err_p_fe_ib_h_ecc_ue	RWH	0	Inbound Header Buffer ECC Uncorrectable Error.
26	pb_err_p_fe_ob_d_ecc_ue	RWH	0	Outbound Data buffer ECC UE.
27	pb_err_p_fe_ob_h_ecc_ue	RWH	0	Outbound Header buffer ECC UE.
28:35	Reserved	RO	0	Reserved.
36	pb_err_p_fe_pci_rcv_cpl_ca_e	RWH	0	Completer Abort (CA) completion response received.
37	pb_err_p_fe_pci_rcv_cpl_crs_e	RWH	0	Config Retry Status (CRS) completion response received.
38	Reserved	RO	0	Reserved.
39	pb_err_p_fe_pci_rcv_cpl_ur_e	RWH	0	Unsupported Request Received Error.
40	pb_err_p_fe_pci_rcv_ecrc_e	RWH	0	ECRC error detected on inbound TLP (from PTL core).
41	pb_err_p_fe_pci_rcv_malf_tlp_e	RWH	0	Malformed TLP error.
42	pb_err_p_fe_pci_rcv_overflow_e	RWH	0	Receiver overflow detected (from TLDLP core).
43	pb_err_p_fe_pci_rcv_poisoned_tlp_e	RWH	0	Received Inbound Poisoned TLP.
44	pb_err_p_fe_pci_rcv_unexp_cpl_e	RWH	0	Unexpected Completion Received Error.
45	pb_err_p_fe_pci_rcv_unsup_req_e	RWH	0	Unsupported Request Received Error.
46	Reserved	RO	0	Reserved.
47	pb_err_p_fe_pci_sig_cpl_timeout_e	RWH	0	Completion Timeout Error.
48	pb_err_p_fe_pci_sig_poisoned_tlp_e	RWH	0	Signaled outbound poisoned TLP.
49:51	Reserved	RO	0	Reserved.
52	pb_err_p_inf_out_trans_to_pst_e	RWH	0	Outbound Posted Request transmission timeout.
53	pb_err_p_inf_out_trans_to_npst_e	RWH	0	Outbound Nonposted Request transmission timeout.
54	pb_err_p_inf_out_trans_to_cpl_e	RWH	0	Outbound Completion transmission timeout.
55:63	Reserved	RO	0	Reserved.

4.6.2.10 PBL - Error INF Enable Register

This register enables specific error bits to assert the Informative 'INF' output signals on the PBL block.

Mnemonic PBL_ERR_INF_EN

Address Offset 0x1920

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *PBL - Error Status Register* on page 245.	RW	0	Bits correspond to the INF error class. INF class errors will set a bit in the REGB Error Status Register, refer to page 285.

4.6.2.11 PBL - Error ERC Enable Register

This register enables specific error bits to assert the Informative 'DEV' output signals on the PBL block.

Mnemonic PBL_ERR_ERC_EN

Address Offset 0x1928

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *PBL - Error Status Register* on page 245.	RW	0	Bits correspond to the ERC error class. ERC class errors will set a bit in the REGB Error Status Register, refer to page 285.

4.6.2.12 PBL - Error FAT Enable Register

This register enables specific error bits to assert the informative 'FAT' output signals on the PBL block.

Note: The fatal 'FAT' output signals will feed into the logic that will 'kill' the PCIe link. When the fatal signals assert it will bring the PCIe link down.

Mnemonic PBL_ERR_FAT_EN

Address Offset 0x1930

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *PBL - Error Status Register* on page 245.	RW	0	Bits correspond to the FAT error class. FAT class errors will set a bit in the REGB Error Status Register, refer to page 285.

4.6.2.13 PBL - Error Log Register 0

This register contains error log information for errors that are active in the Error Status Register. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic PBL_ELR_0

Address Offset 0x1940

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log 0	ROHS	0	Error log contents.

4.6.2.14 PBL - Error Log Register 1

This register contains error log information for errors that are active in the Error Status Register. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic PBL_ELR_0

Address Offset 0x1940

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log 1	ROHS	0	Error log contents.

4.6.2.15 PBL - Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic PBL_ESMR

Address Offset 0x1950

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Status Mask	RWS	0	Each mask bit corresponds to an error bit in the PBL Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

4.6.2.16 PBL - First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic PBL_FESMR

Address Offset 0x1958

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	First Error Status Mask	RWS	0	Each mask bit corresponds to an error bit in the PBL First Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error. In addition to masking the status register, this mask will prevent an error from capturing logging information in the error log registers.

4.6.3 PCIe Stack Registers

4.6.3.1 PCIe - System Configuration Register

This register is related to PCI Express functionality in the PCIe macro.

Warning: Changing any of the values in this register from their reset values will have unpredictable results. Consult with the logic designers before changing any field.

Mnemonic PCIE_SCR

Address Offset 0x1A00

Bit	Field Mnemonic	Type	Reset Value	Description
00:03	SYS_EC00_PORTTYPE[3:0]	RW	0x4	Default to Root Complex. 4'b0000 - Endpoint. 4'b0100 - Root Complex. Can change to Endpoint to support PHB3 Gen 3 speed wrap mode. NOTE: This only affects the TLDLP core behavior.
04:09	SYS_EC0C_MAXLINKWIDTH[5:0]	RW	0x10 (x16) 0x08 (x08)	Maximum link width to the PCI Express capabilities link capabilities register. Valid encodings are: • 6'b000001 - x1 • 6'b000100 - x4 • 6'b001000 - x8 • 6'b010000 – x16 Default setting is based on build type, as x08 or x16 link width.
10:12	SYS_EC04_MAX_PAYLOAD[2:0]	RW	0x2	Max payload size support to the PCI Express capabilities device capabilities register. The PCI Express Specification v1.0a defines the following encoding: • 3'b000 - 128-byte max payload. • 3'b001 - 256-byte max payload. • 3'b010 - 512-byte max payload (default, this is the maximum supported size in the PHB4). • 3'b011 - 1024-byte max payload (not supported in PHB4, results are not predictable). • 3'b100 - 2048-byte max payload (not supported in PHB4, results are not predictable). • 3'b101 - 4096-byte max payload (not supported in PHB4, results are not predictable). • 3'b110 - Reserved. • 3'b111 - Reserved.
13:14	Reserved	RW	0	Reserved, but implemented register(s) for future use.
15	SYS_EC00_SLOT	RW	0	PCIe Slot is supported. This sets the EC00 slot status bit in the PCI CFG space.
16:30	Reserved	RW	0	Reserved, but implemented register(s) for future use.

Bit	Field Mnemonic	Type	Reset Value	Description
31	SYS_EC30_SELDEEMPHASIS	RW	0	TLDLP core strap input. Note: If loopback is entered from reset, the data rate of the link will be the highest common data rate supported by both components. If the data rate is 5.0 GT/s, then the transmitter setting (deemphasis) is controlled by the SYS_EC30_SELDEEMPHASIS input. If the data rate is 8.0 GT/s, then the transmitter settings (presets of each lane) are controlled by the SYS_SEC0C_LANEEQCONTROL input.
32:35	SYS_EC0C_MAXLINKSPEED[3:0]	RW	0x4	PCI Express Maximum Link Speed. Valid encodings are: • 4'0001 – Gen,1: only 2.5Gb/s link speed supported. • 4'0010 – Gen 2: Both 2.5Gb/s and 5.0Gb/s link speeds are supported. • 4'0011 – Gen 3: 2.5, 5.0, and 8.0 Gb/s link speeds are supported. • 4'0100 – Gen 4: 2.5, 5.0, 8.0, 16 Gb/s link speeds are supported (default). All other values are reserved and invalid and will break the link logic.
36:39	Reserved	RW	0	Reserved, but implemented register(s) for future use.
40:45	Physical Link Width[5:0]	RO	0x10 (x16) 0x08 (x08)	This is the physically built link width for this PCIe port, it is a read-only value. This is provided as a read-only reference for firmware in case it changes the SYS_EC0C_MAXLINKWIDTH[5:0] to a different value and wants to restore the value to the maximum physical width of the port. Valid encodings are: • 6'b000001 - x1 • 6'b000100 - x4 • 6'b001000 - x8 • 6'b010000 – x16
46:63	Reserved	RO	0	Reserved.

4.6.3.2 PCIe - Bus Number Register

This register is related to PCI Express functionality in the PCIe macro. This register reads the current PCIe Bus numbers without having to issue reads to CFG space.

Mnemonic PCIE_BNR

Address Offset 0x1A08

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	CFG_PCI18_PRIMARY_BUS[7:0]	ROH	0	Primary bus number register. This value is a shadow copy of the current value in the CFG space register 0x18, byte 0.
08:15	CFG_PCI18_SECONDARY_BUS[7:0]	ROH	0	Secondary bus number register. This value is a shadow copy of the current value in the CFG space register 0x18, byte 1 (or CFG address 0x19 to be accurate).
16:23	CFG_PCI18_SUBORDINATE_BUS[7:0]	ROH	0	Subordinate bus number register. This value is a shadow copy of the current value in the CFG space register 0x18, byte 2 (or CFG address 0x0A to be accurate).
24:63	Reserved	RO	0	Reserved.

4.6.3.3 PCIe - Core Reset Register

This register controls resets to the PCIe core blocks. This register is used in the REGB initialization sequence. Strap values are set in the sequence before resets are released to the cores.

Mnemonic PCIE_CRESET

Address Offset 0x1A10

Bit	Field Mnemonic	Type	Reset Value	Description
0	CFG Core Reset	RW	1	'0' = reset deasserted. '1' = assert reset on the internal PCI CFG core register block.
1	TLDLP and PTL Core Reset	RW	1	'0' = reset deasserted. '1' = assert reset on the internal TLDLP/PTL core blocks.
2	PBL Core Reset	RW	1	'0' = reset deasserted. '1' = assert reset on the internal PBL block.
3	hpo_perst_n	RWS	0	This bit will drive the active low reset signal for the Hot-Plug slot, fundamental reset. This reset will not affect other resets defined in this register or any other logic. It will merely drive the output and the chip output slot reset signal. This bit is sticky, meaning it will retain its value even if the PCIe macro is reset in any capacity. This reset is active low. Firmware writes this register to a '1' during initialization to release reset to the slot attached to the PCIe port.
4	pipeo_resetn	RW	0	This reset will drive the PIPE/PIE-8 interface reset on the PCS core. This bit will default to a 0, which is active. Firmware writes this register to a '1' during initialization to release reset to the PIPE interface on the PCS core.
05:63	Reserved	RO	0	Reserved.

4.6.3.4 PCIe - DLP Strapping Register

This register is related to PCI Express functionality in the PCIe macro.

Mnemonic PCIE_STR

Address Offset 0x1A18

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	regbo_sys_localnfts_25(7:0)	RW	0xFD	NFTS Value for Gen 1: 2.5 Gbps. Number of NFTS ordered sets required to exit L0s state. Note: The NFTS value also affects the Replay Timeout period. The time-out error signal from the TLDLP core is DL_EC08_REPLAYTIMEOUT.
08:15	regbo_sys_localnfts_50(7:0)	RW	0xFD	NFTS Value for Gen 2: 5 Gbps.
16:23	regbo_sys_localnfts_80(7:0)	RW	0xFD	NFTS Value for Gen 3: 8 Gbps.
24:31	regbo_sys_localnfts_16(7:0)	RW	0xFD	NFTS Value for Gen 4: 16 Gbps.
32	regbo_sys_eqpresetpolicy_80	RW	0	This signal controls how the Physical Layer handles 8.0 GT/s Transmitter Preset and Receiver Preset Hint information for an equalization process. This signal must not be changed after SYS_RESETN is deasserted.
33	regbo_sys_eqpresetpolicy_16	RW	0	This signal controls how the Physical Layer handles 16 GT/s Transmitter Preset and Receiver Preset Hint information for an equalization process. This signal must not be changed after SYS_RESETN is deasserted.
34	regbo_sys_eqtuningpolicy_80	RW	0	This signal controls how the Physical Layer handles the 8.0 GT/s remote transmitter equalization tuning process. This signal must not be changed after SYS_RESETN is deasserted.
35	regbo_sys_eqtuningpolicy_16	RW	0	This signal controls how the Physical Layer handles the 16 GT/s remote transmitter equalization tuning process. This signal must not be changed after SYS_RESETN is deasserted.
36:63	Reserved	RO	0	Reserved.

4.6.3.5 PCIe- Hot Plug Status Register

This register is used to read the current card slot presence status.

Mnemonic PCIE_HPSTAT

Address Offset 0x1A20

Bit	Field Mnemonic	Type	Reset Value	Description
00:02	Reserved	RW	0	Reserved
3	Simspeed diagnostic bit	RW	0	The hot plug inputs are debounced digitally by using counters. Under normal conditions, these counters have very high values that translate to signal debounce times of 10ms or more. Setting this bit reduces the counter times to very small values to add in diagnostics and simulation. There is currently only one hot-plug input signal, hpi_present_n. This bit is for diagnostics only. For normal hot plug function, this bit must always be zero. Note: If simulation code changes this bit (Simspeed diagnostic bit) it should also write bit '9' so that the input status signal will re-sample correctly.
04:08	Reserved	RO	0	Reserved.
9	Force re-sample of hot-plug input signals	WO	0	Force re-sample of hpi_present_n input (write-only). Note: If simulation code changes bit 3 (Simspeed diagnostic bit) it should also write this bit to '1' so the input status signal will re-sample correctly.
10	hpi_present_n	ROH	1	The firmware team needs this signal and uses this status signal in the firmware code. Current state of hot-plug card presence detect signal. This signal is active low and reads as '0' when a card is plugged into the PCIe slot.
11	Reserved	RO	0	Reserved.
12	TL_EC10_LINKACTIVE	ROH	0	This signal is asserted when a physical link is established and VC0 flow control negotiation has completed. Also used to drive the Data Link Layer Link Active bit of the link status register.
13:63	Reserved	RO	0	Reserved.

4.6.3.6 PCIe - Link Management Register

This register is related to PCI Express functionality in the PCIe macro.

Mnemonic PCIE_LMR

Address Offset 0x1A30

Bit	Field Mnemonic	Type	Reset Value	Description
0	SYS_CHANGELINKWIDTH	WO	0	When this bit is set to '1' when the register is written, a link width change request will be issued. The SYS_CHANGELINKWIDTH signal is asserted for one clock cycle. This bit is always '0' when read.
1	CFG_EC10_RETRAINLINK	WO	0	A link retrain can be initiated by driving this signal to '1'. The internal logic will drive this input on the TLDLP core for 1 PCLK cycle. It can be generated from the Retrain Link bit in the PCI CFG space Link Control register. Note: This signal must not be asserted when SYS_EC00_PORTTYPE defines the core to be part of an upstream facing port (ex-endpoint port).
02:07	SYS_TARGETLINKWIDTH[5:0]	RW	0x10 (x16) 0x08 (x08)	This signal defines a target link width for the core to negotiate with directed by SYS_CHANGELINKWIDTH. Defined encodings are: • 000001 - x1 • 000010 - x2 • 000100 - x4 • 001000 - x8 • 010000 - x16 (this is the maximum width) All other encodings are reserved.
08	TL_EC10_LINKACTIVE	ROH	0	This signal is asserted when a physical link is established and VC0 flow control negotiation has completed. Also used to drive the Data Link Layer Link Active bit of the Link Status Register.
09	DL_WIDTHCHANGEPENDING	ROH	0	When the link is active, this signal is asserted one cycle after the signal SYS_CHANGELINKWIDTH is asserted. It is deasserted when the requested link width change is complete, or if no action was taken.
10	DL_UPCONFIGURECAPABLE	ROH	0	This signal is asserted when both Ports on the Link are capable of link width up configuration. It's value is valid only when the link is active.
11	SYS_AUTOLINKSPEEDEN	RW	0	Set to '1' to allow the TLDLP core to make automatic link speed changes.
12:15	SYS_AUTOLINKSPEED[3:0]	RW	0	Set the maximum allowed automatic link speed. Valid encodings are: • 4'0001 - Only 2.5 Gb/s link speed supported. • 4'0010 - Both 2.5 Gb/s and 5.0 Gb/s link speeds are supported. • 4'0011 - 2.5, 5.0, and 8.0 Gb/s link speeds are supported (default).
16	CFG_EC10_HWAUTO_WIDTH_DISABLE	ROH	0	Status bit. When asserted, the hardware has been disabled from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width.
17:18	Reserved	RW	0	Reserved, but implemented register(s) for future use.
19	CFG_EC30_HWAUTO_SPEED_DISABLE	ROH	0	If this bit is set in the PCI CFG space, the SYS_AUTOLINKSPEED mechanism will not function.

Bit	Field Mnemonic	Type	Reset Value	Description
20:23	Reserved	RW	0	Reserved, but implemented register(s) for future use.
24:63	Reserved	RO	0	Reserved.

4.6.3.7 PCIe - DLP Training Control Register

This register is related to PCI Express functionality in the PCIe macro.

Mnemonic PCIE_DLP_TCR

Address Offset 0x1A40

Bit	Field Mnemonic	Type	Reset Value	Description
0	SYS_TCTX_RESET	RW	0	Assert this signal when the core is configured as a downstream facing port (root port, and so forth) and the hierarchy below the port is to be reset. Normally, this signal is driven by the Secondary Bus Reset bit of the Bridge Control register.
1	SYS_TCTX_DISABLE	RW	0	Assert this signal when the core is configured as a downstream facing port (root port, and so forth) and the hierarchy below the port is to be disabled.
2	SYS_TCTX_SCRAMBLEOFF	RW	0	Assert this signal when the core is configured as either an upstream or downstream facing port and scrambling should be disabled on the link. The source of this signal is application specific. It must be asserted before the link is configured to be effective.
3	SYS_TCTX_LOOPBACK	RW	0	Assert this signal when the core is configured as either an upstream or down stream facing port and loop back mode is desired. The source of this signal is application specific. It must be asserted before the link is configured to be effective.
04:11	Reserved	RW	0	Reserved, but implemented register(s) for future use.
12	SYS_DISABLEDETECT	RW	0	Set to '1' to disable the link bringup sequence.
13:14	Reserved	RW	0	Reserved, but implemented register(s) for future use.
15	SYS_TCRX_RESET	ROH	0	Status output. Asserted when the link is in the reset state.
16	DL_TCRX_DISABLE	ROH	0	Status output. Asserted when the link is in the disabled state.
17	DL_TCRX_SCRAMBLEOFF	ROH	0	Status output. Asserted when a link has been established with scrambling disabled.
18	DL_TCRX_LOOPBACK	ROH	0	Status output. Asserted when the link is in the loopback state.
19	tldlpo_pl_inbandpresence	ROH	0	Status output. This signal is the physical layer's In-Band Presence status. It can be used to generate the Presence Detect State bit of the Slot Status register.
20	DL_EC10_LINKTRAINING	ROH	0	This signal is asserted when link training is in progress.
21	tldlpo_pl_linkup	ROH	0	Status output. This signal is the physical layer's LinkUp status.
22	DL_PGRESET	ROH	0	Status output. Power-Good Reset. This signal is active high. This is a version of the SYS_RESETN and is used by the TLDLP core. When '1' it indicates the IOP/PMA/PHY core is still held in reset. When '0' it indicates the IOP/PMA/PHY core is operational and out of reset.

Bit	Field Mnemonic	Type	Reset Value	Description
23	TL_EC10_LINKACTIVE	ROH	0	Status output. Asserted when the core has finished credits initialization. The link is now ready to transmit / receive TLPs. This is the third, and last, phase of a “link up” sequence.
24:27	tldlpo_dl_tracerddata(3:0)	ROH	0x0	TLDLP Link Control State Machine State. The following states are defined: • LTSSM_RESET = '0000' (0) • LTSSM_DETECT = '0001' (1) • LTSSM_POLLING = '0010' (2) • LTSSM_CONFIG = '0011' (3) • LTSSM_L0 = '0100' (4) • LTSSM_RECOVERY = '0101' (5) • LTSSM_L1 = '0110' (6) • LTSSM_L2 = '0111' (7) • LTSSM_HOTRESET = '1000' (8) • LTSSM_DISABLED = '1001' (9) • LTSSM_LOOPBACK = '1010' (10) • States not listed are Reserved Note: This field only represents the link states when the TLDLP trace interface is in its default state or configured specifically to display the link state in these bits. See *PCIe - DLP Trace Read/Write Control Register* on page 270.
28	tldlpo_dl_up	ROH	0	Status output. This signal is the Data Link Layer’s DL_Up status.
29	tldlpo_dl_ec30_retimerpresence	ROH	0	Status output. Retimer Presence Detected. For a downstream port, when set to 1b, this bit indicates that a Retimer was present during the most recent Link negotiation. The default value of this bit is 0b. This bit is required for downstream ports that have the Retimer Presence Detect Supported bit Set. Downstream ports that have the Retimer Presence Detect Supported bit of the Link Capabilities 2 register set to 0b are permitted to hardwire this bit to 0b. This bit is Reserved for upstream ports.
30:35	DL_EC10_NEGLINKWIDTH(5:0)	ROH	0	This signal is the current Link width. The width after link is negotiation and trained.
36:39	DL_EC10_CURRENTLINKSPEED(3:0)	ROH	0	This signal is the current Link speed. The speed after link is negotiated and trained.
40:63	Reserved	RO	0	Reserved.

4.6.3.8 PCIe - DLP Loopback Status Register

This register is related to PCI Express functionality in the PCIe macro.

Mnemonic PCIE_DLP_LSR

Address Offset 0x1A48

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	DL_LB_ACTIVE(0:15)	ROH	0	This signal is asserted per lane when the lane is in loopback.
16:31	DL_LB_ERROR(0:15)	ROH	0	This signal is asserted per lane for one PCLK cycle each time an error is detected in the received loopback data stream. This is the current state of the error signal and does not hold.
32:47	DL_LB_ACTIVE(0:15)	ROH	0	Copy of same DL_LB_ACTIVE bits as above.
48:63	Trapped DL_LB_ERROR(0:15)	RW1CH	0	This signal is asserted per lane for one PCLK cycle each time an error is detected in the received loopback data stream. This is the 'trapped' error state. The value holds until it is cleared.

4.6.3.9 PCIe - DLP Fault Isolation Register

This register is related to PCI Express functionality in the PCIe macro.

Mnemonic PCIE_DLP_FAULTISO

Address Offset 0x1A70

Bit	Field Mnemonic	Type	Reset Value	Description
0	regbo_sys_faultisoenable	RW	0	Fault Isolation Enable. When set to '1', enables transmission of status reporting format SKP Ordered Sets.
01:08	Reserved	RW	0	Reserved, but implemented register(s) for future use.
09:31	regbo_sys_faultisotxdata(22:0)	RW	0	Fault Isolation TX Data. This defines the content of transmitted status reporting format SKP Ordered Sets. Example setting that matches the proposed Fault Isolation specification in the PHY Logical subteam is: 23'b000000_00000011_00000011.
32:47	dl_sec_rt1errorstatus(0:15) (vA4.1) dl_p1614_rt1parstatus(0:15) (vA4.2)	RW1CH	0	Fault Isolation 'RT1' Error Status (vA4.1). One bit per lane. Each bit is driven to 1 for one PCLK cycle to indicate that the respective lane of the first Retimer along the path detected a SKP OS Data Parity mismatch. First Retimer Data Parity Mismatch Status (vA4.2) Each bit indicates if the corresponding lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding lane number.
48:63	dl_sec_rt2errorstatus(0:15) (vA4.1) dl_p1618_rt2parstatus(0:15) (vA4.2)	RW1CH	0	Fault Isolation 'RT2' Error Status (vA4.1) One bit per lane. Each bit is driven to 1 for one PCLK cycle to indicate that the respective lane of the second Retimer along the path detected a SKP OS Data Parity mismatch. Second Retimer Data Parity Mismatch Status (vA4.2) Each bit indicates if the corresponding lane detected a Data Parity mismatch. A value of 1b indicates that a mismatch was detected on the corresponding lane number.

4.6.3.10 PCIe - DLP Control Register

This register is related to PCI Express functionality in the PCIe macro. Do not change the default value for each field below without advanced knowledge of the consequences.

Mnemonic PCIE_DLPCTL

Address Offset 0x1A78

Bit	Field Mnemonic	Type	Reset Value	Description
00:03	SYS_DLLPTLPRATIO(0:3)	RW	0x3	Strap input that defines the ratio of DLLPs to TLPs transmitted over the link. Default is 3 DLLPs to 1 TLP.
4	regbo_sys_eqbypassphase2_80	RW	0	This signal, when asserted, directs the physical layer to alter its execution of Phase 2 of the 8.0 GT/s data rate equalization process.
5	regbo_sys_eqbypassphase3_80	RW	0	This signal, when asserted, directs the physical layer to alter its execution of Phase 3 of the 8.0 GT/s data rate equalization process (EQ). The EQ process is as follows for downstream-facing (DSF) Ports and upstream-facing (USF) Ports: 1: DSF in Phase 1, USF in Phase 0 2: DSF in Phase 1, USF in Phase 1. DSF will end the EQ process here if SYS_EQBYPASSPHASE2 = 1. 3: DSF in Phase 2, USF in Phase 1 4: DSF in Phase 2, USF in Phase 2. USF will proceed to Phase 3 (and not adjust the DSF's transmitter) if SYS_EQBYPASSPHASE2 = 1. 5: DSF in Phase 2, USF in Phase 3. 6: DSF in Phase 3, USF in Phase 3. DSF will end the EQ process (and not adjust the USF's transmitter) if SYS_EQBYPASSPHASE3 = 1. 7: DSF in Recovery.RcvrLock, USF in Phase 3. 8: DSF in Recovery.RcvrLock, USF in Recovery.RcvrLock. Normal Link retrain process. A Root Port is a DSF Port. An Endpoint is an USF Port. An embedded Link will never get to "8 GB" so the inputs can be set to any value. In steps 4 and 6 above, "bypass" simply means that the TX adjustment is bypassed (the state exit condition of “all lanes are operating properly” is satisfied immediately) and the port proceeds to the next state.
6	SYS_EQEIEOSCOUNT	RW	0	The setting of this signal depends on the requirements of the PHY being used. See the Physical Layers documentation for guidance. This signal, when asserted, directs the PHY to request an EIEOS transmission interval of 65,536 Blocks (instead of 32 Blocks) from the remote transmitter when adjusting the remote transmitter’s settings during the 8.0 GT/s data rate equalization process. This signal must not be changed after SYS_RESETN is deasserted.

Bit	Field Mnemonic	Type	Reset Value	Description
7	regbo_sys_eqcontrol	RW	0	This signal controls when the core starts to evaluate information received from the PHY after a data rate change to 8.0 GT/s or 16.0 GT/s. When ‘0’, the core starts to evaluate information after completing the data rate change. When ‘1’, the core starts to evaluate information 2ms after completing the data rate change. This feature is intended to allow the PHY’s receiver additional time to adjust to the link signal after a data rate change to 8.0 GT/s or 16.0 GT/s. Many PHYs implement DFE, and additional adjustment time might significantly improve the bit error rate. It must not be changed after SYS_RESETN is deasserted. When TIE0_SIMSPEED = 1, the core starts to evaluate information 2µs after completing the data rate change.
8	Lane Swapping Bit	RW	0	Controls the PCIe lane swapping. 0 = normal lane assignment 1 = lanes are swapped Note: This is lane swapping, not lane reversal. The hardware design does not support true lane reversal. This swaps the physical/logical lane relationship across the full width. Note the hardware (wrapper outside the TLDLP core) takes bifurcation into account automatically and will swap within the bifurcation width only. The final lane swap signal sent to the TLDLP core is the logical OR of this register bit and the pervasive input signal. This register bit is only be used for debug purposes (typically always set to '0'). The primary method of controlling lane swapping should be via the pervasive signal. regbo_sys_laneswap(to TLDLP) = ('this register bit' OR tc_pe_swap_dc);
9	SYS_FORCEDETECT	RW	0	This signal, when asserted, causes the receiver detect results from the PHY to be overridden by the SYS_FORCEDETECTRESULTS input. This signal must not be changed after SYS_RESETN is deasserted. This normally does not need to be changed.
10	regbo_sys_eqbypassphase2_16	RW	0	This signal, when asserted, directs the PHY to change how it executes Phase 2 of the 16.0 GT/s data rate equalization process. When the port is downstream-facing, Phases 2 and 3 are not executed. When the port is upstream-facing, Phase 2 is not executed. It is treated as though it were executed and all lanes achieved their optimal settings. It must not be changed after SYS_RESETN is deasserted.

Bit	Field Mnemonic	Type	Reset Value	Description
11	regbo_sys_eqbypassphase3_16	RW	0	This signal, when asserted, directs the PHY to change how it executes phase 3 of the 16.0 GT/s data rate equalization process. When the port is downstream-facing, phase 3 is not executed. It is treated as though it were executed and all lanes achieved their optimal settings. This signal has no effect on an upstream-facing port. It must not be changed after SYS_RESETN is deasserted.
12:13	SYS_EQTIMER1(1:0)	RW	0	Strap input that defines the EQ Timer 1 value to the TLDLP core. This signal defines the time the port waits for a requested remote transmitter equalization setting to take effect. The encodings are: • 2'b00 - 2 µs (recommended default setting) • 2'b01 - 4 µs • 2'b10 - 6 µs • 2'b11 - 8 µs This input must not be set to a value of 8µs if SYS_EQTIMER2 is set to a value of 8µs. When TIE0_SIMSPEED = 1, these times are reduced to 200, 400, 600, and 800 ns.
14:15	SYS_EQTIMER2(1:0)	RW	0	Strap input that defines the EQ Timer 2 value to the TLDLP core. This signal defines the time the port waits for the remote transmitter to respond to a requested equalization setting change. The encodings are: • 2'b00 - 8 µs (recommended default setting) • 2'b01 - 16 µs • 2'b10 - 24 µs • 2'b11 - 32 µs When TIE0_SIMSPEED = 1, these times are reduced to 800, 1600, 2400, and 3200 ns.
16	regbo_sys_upconfigurecapable	RW	1	Enable PCIe core logic support for link width upconfiguration. This signal, when asserted, enables the Port's support for link width upconfiguration. It must not be changed after SYS_RESETN is deasserted.
17	regbo_sys_tctx_comprec	RW	0	Assert this signal when the core is configured as either an upstream or downstream-facing port and SYS_TCTX_LOOPBACK is asserted to modify the master loopback operation. This signal must not be changed while SYS_TCTX_LOOPBACK is asserted.
18	regbo_sys_srismode	RW	0	Set this signal to '0' when using a common reference clock architecture of a Separate Reference Clocks with No SSC (SRNS) architecture. Set this input to '1' when using a Separate Reference Clocks with Independent SSC (SRIS) architecture. It must not be changed after SYS_RESETN is deasserted.
19:31	SYS_ENTERL0STIME(0:12)	RW	0x0FA	Strap input that defines the L0s link power state entry time. This is the number of PCIe clock cycles that the link is idle before entering the L0s state. The default is 250d (0x0FA) clock cycles.

Bit	Field Mnemonic	Type	Reset Value	Description
32:47	SYS_FORCEDETECTRESULTS(15:0)	RW	0	This signal specifies the receiver detect results when SYS_FORCEDETECT is asserted. Bit 0 represents lane 0, and bit 15 represents lane 15. This signal must not be changed after SYS_RESETN is deasserted.
48:49	regbo_sys_ackpolicy(1:0)	RW	0b11	This signal controls the Ack transmission scheduling behavior. The TLDLP core schedules an Ack DLLP for transmission when it forwards a received TLP to the application logic for processing, and will then transmit an Ack DLLP as defined by the SYS_ACKPOLICY input. The policies are: • '00' - Transmit an Ack DLLP after allowing time for multiple TLPs to be received. • '01' - Transmit an Ack DLLP after allowing time for multiple TLPs to be received, but less time than policy '00'. • '10' - Transmit an Ack DLLP after allowing time for multiple TLPs to be received, but more time than policy '00'. • '11' - Transmit an Ack DLLP as soon as possible after receiving a TLP. The SYS_ACKPOLICY input can be used to tune the performance of your system. It is possible for links with large transmission latencies to experience TLP data bandwidth limitations due to unexpectedly small Transmit Retry Buffer sizes, and using policy '01' to reduce the Ack DLLP transmission time might improve the situation. It is also possible that the Transmit Retry Buffers are not a limitation, and using policy '10' to increase the Ack DLLP transmission time can increase Link bandwidth available for TLP data by reducing the number of Ack DLLPs transmitted. Default to policy '11', transmit Ack DLLP as soon as possible.

Bit	Field Mnemonic	Type	Reset Value	Description
50:51	regbo_sys_rxcreditpolicy_vc0(1:0)	RW	0b11	This signal controls the receiver flow control update transmission scheduling behavior for VC0. UpdateFC DLLPs for non-infinite credit types are scheduled for transmission based on the setting of the SYS_RXCREDITPOLICY_VCx core inputs. The policies are: • '00' - Transmit an UpdateFC DLLP after allowing time for the application logic to process multiple TLPs. • '01' - Transmit an UpdateFC DLLP after allowing time for the application logic to process multiple TLPs, but less time than policy '00'. • '10' - Transmit an UpdateFC DLLP after allowing time for the application logic to process multiple TLPs but more time than policy '00'. • '11' - Transmit an UpdateFC DLLP as soon as possible after the application logic processes a TLP. The SYS_RXCREDITPOLICY_VCx inputs can be used to tune the performance of your system. It is possible for links with large transmission latencies to experience TLP data bandwidth limitations due to unexpectedly small TLP buffer sizes, and using policy '01' to reduce the UpdateFC DLLP transmission time might improve the situation. It is also possible that the TLP buffers are not a limitation, and using policy '10' to increase the UpdateFC DLLP transmission time can increase Link bandwidth available for TLP data by reducing the number of UpdateFC DLLPs transmitted. The core will automatically schedule transmission of UpdateFC DLLPs at the interval of 30µs (-0%/+50%) required by the PCI Express Base Specification. Default to policy '11', transmit UpdateFC DLLP as soon as possible.
52	regbo_sys_16geieos	RW	0	This signal controls the pattern used for the Electrical Idle Exit Ordered Set (EIEOS) when operating at 16.0 GT/s. When '0', the pattern is sixteen 0s followed by sixteen 1s. When '1', the pattern is eight 0s followed by eight 1s. Set to '1' for operation with components designed to the PCI Express Base Specification 4.0 draft 0.5 and earlier revisions. Default '0' is the recommended setting to cover current and future PCIe Gen 4 designs after draft 0.5.
53:55	Reserved	RW	0	Reserved, but implemented register(s) for future use.
56:63	Reserved	RO	0	Reserved.

4.6.3.11 PCIe - DLP Trace Read/Write Control Register

This register controls the tracing behavior in the TLDLP core.

Mnemonic PCIE_DLP_TRWCTL

Address Offset 0x1A80

Bit	Field Mnemonic	Type	Reset Value	Description
0	regbo_sys_tracerden	RW	0	TLDLP core trace read-enable signal. Set to '1' to capture trace information from the TLDLP core. This signal also acts as a clock gating signal, set to '0' when not tracing from the TLDLP to save power.
1	regbo_sys_tracewren	WO	0	TLDLP Trace Write Enable Pulse. Single cycle PCLK pulse, write-enable signal sent to TLDLP core. This write-only pulse will update the trace control information in the TLDLP core using the write address and write data information from this register.
02:07	Reserved	RW	0	Reserved, but implemented register(s) for future use.
08:15	regbo_sys_tracerdaddr(7:0)	RW	0	TLDLP Trace Read Address. This selects the source of information to be reflected on the DL_TRACERDDATA output.
16:23	regbo_sys_tracewraddr(7:0)	RW	0	TLDLP Trace Write Address. This input is used to control the setup of the internal trace logic.
24:31	regbo_sys_tracewrdata(7:0)	RW	0	TLDLP Trace Write Data. This input is used to control the setup of the internal trace logic.
32:63	Reserved	RO	0	Reserved.

4.6.3.12 PCIe - DLP Trace Read Data Register

This register controls the tracing behavior in the TLDLP core.

Mnemonic PCIE_DLP_TRCRDDATA

Address Offset 0x1A88

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	tldlpo_dl_tracerddata(63:0)	ROH	0	TLDLP core trace read data output. This register provides a read-only, current value of the TLDLP core trace output bus.

4.6.3.13 PCIe - DLP Error Log Register 1

This register contains error status bits for error indicators internal to the PCIe cores.

Mnemonic PCIE_DLP_EL1

Address Offset 0x1AA0

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	DL_INTERNALERROR(64:95)	RW1CHS	0	Trap register for TLDLP error bits.
32:63	Reserved	RO	0	Reserved.

4.6.3.14 PCIe - DLP Error Log Register 2

This register contains error status bits for error indicators internal to the PCIe cores.

Mnemonic PCIE_DLP_EL2

Address Offset 0x1AA8

Bit	Field Mnemonic	Type	Reset Value	Description
00:31	DL_INTERNALERROR(64:95)	RW1CHS	0	Trap register for TLDLP error bits.
32:63	Reserved	RO	0	Reserved.

4.6.3.15 PCIe - DLP Error Status Register

This register contains error status bits for error detected by the TLDLP core. These errors are raw status bits only. They have no effect on the logic in the ETU.

Mnemonic PCIE_DLP_ERR_STAT

Address Offset 0x1AB0

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	DL_SEC08_LANEERROR STATUS(0:15)	RW1CHS	0	This signal reports errors specific to a PCIe lane. Lane numbers match bit position, 0 to 15.
16:63	Reserved	RO	0	Reserved.

4.6.3.16 PCIe - DLP Error Counters Register

This register contains error counters for error status bits for error detected by the TLDLP core.

Note: Counters count low to high assertion edges only. They should not count when the level is high. These signals come from the PCLK domain and the counts are in the BCLK domain. The PCLK domain signals can have a larger pulse width than the BCLK so the counters would count too many times if they were to count the levels.

Mnemonic PCIE_DLP_ERR_CNT

Address Offset 0x1AB8

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	DL_EC08_BADDLLP Count(0:7)	RWH	0	This is an 8-bit counter that saturates at a value of 0xFF.
08:15	DL_EC08_BADTLP Count(0:7)	RWH	0	This is an 8-bit counter that saturates at a value of 0xFF.
16:23	DL_EC08_RECEIVERERROR Count(0:7)	RWH	0	This is an 8-bit counter that saturates at a value of 0xFF. The intent of this counter is to monitor the “health” of the link. Firmware will use this counter information in its algorithms to degrade the link width and speed as it deems necessary.
24:31	DL_EC08_DLLPE Count(0:7)	RWH	0	This is an 8-bit counter that saturates at a value of 0xFF.
32:63	Reserved	RO	0	Reserved.

4.6.3.17 PCIe - DLP Error Injection Control Register

This register contains error status bits for error indicators internal to the PCIe cores. This register must be used with extreme care, as error injection to the core can cause the entire logic to enter an unspecified/locked state, depending on con-current packet traffic.

Mnemonic PCIE_DLP_EIC

Address Offset 0x1AC8

Bit	Field Mnemonic	Type	Reset Value	Description
0	SYS_EI Enable Bit	RW	0	Bit must be set for the selected SYS_EI_* action to occur.
01:03	Reserved	RW	0	Reserved, but implemented register(s) for future use.
4	SYS_EI_FORCE_BAD_TLP	RW	0	When this bit is set, REGB hardware will issue a single clock cycle pulse that will cause the next TLP packet received to be reported as a “bad” TLP. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. Note: Firmware can use this injector to emulate RX LCRC errors. This bit must be set to zero for PCIe compliant operation.
5	SYS_EI_BLOCK_ACK_NAK TRANSMIT	RW	0	When this bit is set, the TLDLP core will stop scheduling Ack and Nak DLLPs for transmission. This affects the other component on the link, and not TLDLP. For example, if a TLP is received, the TLDLP core will not transmit a confirmation of its reception and eventually the sender’s Replay Timer will expire. The sender will set its Replay Timer Timeout Status and transmit the TLP again. This bit must be set to zero for PCIe compliant operation.
6	SYS_EI_BLOCK_ACK_NAK RECEIVE	RW	0	When this bit is set, the TLDLP core will discard received Ack and Nak DLLPs without action. This action does affect TLDLP transmit activity. All DLLPs will continue to be checked for correct CRC, and they can still be reported as Bad DLLPs. However, received Ack and Nak DLLPs will not remove any TLPs from the TLDLP replay buffer and will not reset the Replay Timer. Eventually, an outstanding transmitted TLP will cause the TLDLP core’s Replay Timer to time out and the TLDLP core will assert the EC08_REPLAYTIMEOUT output and initiate a replay of all outstanding TLPs. If the input is asserted long enough, a Replay Rollover also occurs, causing the TLDLP core to assert EC08_REPLAYROLLOVER and retrain the link before initiating a replay. This bit must be set to zero for PCIe compliant operation.
7	SYS_EI_RXRECEIVERERROR	RW	0	When this bit is set the TLDLP will simulate detection of a receiver error, DL_EC08_RECEIVERERROR. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. This bit must be set to zero for PCIe compliant operation.

Bit	Field Mnemonic	Type	Reset Value	Description
8	SYS_EI_TXDLLP	RW	0	When this bit is set the TLDLP will transmit a single DLLP packet with a CRC-16 error. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. This bit must be set to zero for PCIe compliant operation.
9	SYS_EI_TXTLP	RW	0	When this bit is set the TLDLP will transmit a single TLP packet with an LCRC error. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. This bit must be set to zero for PCIe compliant operation.
10	SYS_EI_TXFRAMING	RW	0	When this bit is set the TLDLP will transmit a single DLLP packet with incorrect framing. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. This bit must be set to zero for PCIe compliant operation.
11	regbo_ptl_rx_dat_ecrc_einj	RW	0	PTL Receive Data ECRC Error Injector. Set to '1' to inject an ECRC error on the next inbound TLP. This bit will stay set until software clears it by writing zero. Software must clear this bit, then set it again to force another bad TLP error injection. This bit must be set to zero for PCIe compliant operation.
12:15	Reserved	RW	0	Reserved, but implemented register(s) for future use.
16:31	PIPE TXELECIDLE(0:15)	RW	0	Forces the pipei_txelecidle vector bits to a '1' for each bit set in the vector. This injector can be used to emulate one or more lanes being disconnected. This is a continuous injection. It will remain asserted until firmware/software clears this register.
32:47	PIPE RXELECIDLE(0:15)	RW	0	Forces the pipei_rxelecidle vector bits to a '1' for each bit set in the vector. This injector can be used to emulate one or more lanes being disconnected. This is a continuous injection. It will remain asserted until firmware/software clears this register.

Bit	Field Mnemonic	Type	Reset Value	Description
48:63	PIPE RX Lane Error Inject(0:15)	RW	0	Error inject bit vector, one bit per lane. Forces the following PIPE interface vector bits to a '0' for each bit set in the vector: • pipei_rxvalid • pipei_rxdatavalid (relevant for Gen 3 mode, 128/130 bit encoding (8G) only) • pipei_rxstartblock (relevant for Gen 3 mode, 128/130 bit encoding (8G) only) This injector can be used to emulate one or more lanes having a high bit error rate condition. This is a continuous injection. It will remain asserted until firmware/software clears this register.

4.6.3.18 PCIe - DLP LANEEQCONTROL Register 0 (Gen 3, 8Gbps)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: *regbo_sys_sec0c_laneeqcontrol*. This sets the initial value in the PCI CFG space reg: Lane Equalization Control Register(s).

Mnemonic PCIE_DLP_LANEEQCTL_0

Address Offset 0x1AD0

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	regbo_sys_sec0c_laneeqcontrol(15:00)	RW	0	PCIe Lane 0. This is the same bit definition for all PCIe lanes. There is a 16-bit control value for each lane. Refer to the latest PCIe Gen 3 specification for detailed contents of this register: Lane Equalization Control Register. 15 - reserved. 14:12 - Upstream Port Receiver Preset Hint. 11:08 - Upstream Port Transmitter Preset. 7 - reserved. 06:04 - Downstream Port Receiver Preset Hint. 03:00 - Downstream Port Transmitter Preset.
16:31	regbo_sys_sec0c_laneeqcontrol(31:16)	RW	0	PCIe Lane 1.
32:47	regbo_sys_sec0c_laneeqcontrol(47:32)	RW	0	PCIe Lane 2.
48:63	regbo_sys_sec0c_laneeqcontrol(63:48)	RW	0	PCIe Lane 3.

4.6.3.19 PCIe - DLP LANEEQCONTROL Register 1 (Gen 3, 8Gbps)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: regbo_sys_sec0c_laneeqcontrol. This sets the initial value in the PCI CFG space reg: Lane Equalization Control Register(s).

Mnemonic PCIE_DLP_LANEEQCTL_1

Address Offset 0x1AD8

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	regbo_sys_sec0c_laneeqcontrol(79:64)	RW	0	PCIe Lane 4. This is the same bit definition for all PCIe lanes. There is a 16-bit control value for each lane. Refer to the latest PCIe Gen 3 specification for detailed contents of this register: Lane Equalization Control Register. 15 - reserved. 14:12 - Upstream Port Receiver Preset Hint. 11:08 - Upstream Port Transmitter Preset. 7 - reserved. 06:04 - Downstream Port Receiver Preset Hint. 03:00 - Downstream Port Transmitter Preset.
16:31	regbo_sys_sec0c_laneeqcontrol(95:80)	RW	0	PCIe Lane 5.
32:47	regbo_sys_sec0c_laneeqcontrol(111:96)	RW	0	PCIe Lane 6.
48:63	regbo_sys_sec0c_laneeqcontrol(127:112)	RW	0	PCIe Lane 7.

4.6.3.20 PCIe - DLP LANEEQCONTROL Register 2 (Gen 3, 8Gbps)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: *regbo_sys_sec0c_laneeqcontrol*. This sets the initial value in the PCI CFG space reg: Lane Equalization Control Register(s).

Mnemonic PCIE_DLP_LANEEQCTL_2

Address Offset 0x1AE0

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	regbo_sys_sec0c_laneeqcontrol(143:128)	RW	0	PCIe Lane 8. This is the same bit definition for all PCIe lanes. There is a 16-bit control value for each lane. Refer to the latest PCIe Gen 3 specification for detailed contents of this register: Lane Equalization Control Register. 15 - reserved. 14:12 - Upstream Port Receiver Preset Hint. 11:08 - Upstream Port Transmitter Preset. 7 - reserved. 06:04 - Downstream Port Receiver Preset Hint. 03:00 - Downstream Port Transmitter Preset.
16:31	regbo_sys_sec0c_laneeqcontrol(159:144)	RW	0	PCIe Lane 9.
32:47	regbo_sys_sec0c_laneeqcontrol(175:160)	RW	0	PCIe Lane 10.
48:63	regbo_sys_sec0c_laneeqcontrol(191:176)	RW	0	PCIe Lane 11.

4.6.3.21 PCIe - DLP LANEEQCONTROL Register 3 (Gen 3, 8Gbps)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: regbo_sys_sec0c_laneeqcontrol. This sets the initial value in the PCI CFG space reg: Lane Equalization Control Register(s).

Mnemonic PCIE_DLP_LANEEQCTL_3

Address Offset 0x1AE8

Bit	Field Mnemonic	Type	Reset Value	Description
00:15	regbo_sys_sec0c_laneeqcontrol(207:192)	RW	0	PCIe Lane 12. This is the same bit definition for all PCIe lanes. There is a 16-bit control value for each lane. Refer to the latest PCIe Gen 3 specification for detailed contents of this register: Lane Equalization Control Register. 15 - reserved. 14:12 - Upstream Port Receiver Preset Hint. 11:08 - Upstream Port Transmitter Preset. 7 - reserved. 06:04 - Downstream Port Receiver Preset Hin.t 03:00 - Downstream Port Transmitter Preset.
16:31	regbo_sys_sec0c_laneeqcontrol(223:208)	RW	0	PCIe Lane 13.
32:47	regbo_sys_sec0c_laneeqcontrol(239:224)	RW	0	PCIe Lane 14.
48:63	regbo_sys_sec0c_laneeqcontrol(255:240)	RW	0	PCIe Lane 15.

4.6.3.22 PCIe - DLP P1620-LANEEQCONTROL Register 0 (Gen 4, 16Gbps) (vA4.2)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: regbo_sys_p1620_laneeqcontrol.

Mnemonic PCIE_DLP_P1620_LANEEQCTL_0

Address Offset 0x1AF0

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	regbo_sys_p1620_laneeqcontrol(07:00)	RW	0	PCIe Lane 0. This is the same bit definition for all PCIe lanes. There is an 8-bit control value for each lane. Refer to the latest PCIe Gen 4 specification for detailed contents of this register. 07:04 - Upstream Port Transmitter Preset. 03:00 - Downstream Port Transmitter Preset.
08:15	regbo_sys_p1620_laneeqcontrol(15:08)	RW	0	PCIe Lane 1.
16:23	regbo_sys_p1620_laneeqcontrol(23:16)	RW	0	PCIe Lane 2.
24:31	regbo_sys_p1620_laneeqcontrol(31:24)	RW	0	PCIe Lane 3.
32:39	regbo_sys_p1620_laneeqcontrol(39:32)	RW	0	PCIe Lane 4.
40:47	regbo_sys_p1620_laneeqcontrol(47:40)	RW	0	PCIe Lane 5.
48:55	regbo_sys_p1620_laneeqcontrol(55:48)	RW	0	PCIe Lane 6.
56:63	regbo_sys_p1620_laneeqcontrol(63:56)	RW	0	PCIe Lane 7.

4.6.3.23 PCIe - DLP P1620-LANEEQCONTROL Register 1 (Gen 4, 16Gbps) (vA4.2)

This register drives the initial PCI CFG space value for the lane equalization presets. It drives the internal signal: regbo_sys_p1620_laneeqcontrol.

Mnemonic PCIE_DLP_P1620_LANEEQCTL_1

Address Offset 0x1AF8

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	regbo_sys_p1620_laneeqcontrol(71:64)	RW	0	PCIe Lane 8. This is the same bit definition for all PCIe lanes. There is an 8-bit control value for each lane. Refer to the latest PCIe Gen 4 specification for detailed contents of this register. 07:04 - Upstream Port Transmitter Preset. 03:00 - Downstream Port Transmitter Preset.
08:15	regbo_sys_p1620_laneeqcontrol(79:72)	RW	0	PCIe Lane 9.
16:23	regbo_sys_p1620_laneeqcontrol(87:80)	RW	0	PCIe Lane 10.
24:31	regbo_sys_p1620_laneeqcontrol(95:88)	RW	0	PCIe Lane 11.
32:39	regbo_sys_p1620_laneeqcontrol(103:96)	RW	0	PCIe Lane 12.
40:47	regbo_sys_p1620_laneeqcontrol(111:104)	RW	0	PCIe Lane 13.
48:55	regbo_sys_p1620_laneeqcontrol(119:112)	RW	0	PCIe Lane 14.
56:63	regbo_sys_p1620_laneeqcontrol(127:120)	RW	0	PCIe Lane 15.

4.6.3.24 PCIe - Trace Control Register

This register controls the traced signals in the PCIe block. The traced signals are used for PCIe debug purposes.

Mnemonic PCIE_TCR

Address Offset 0x1B20

Bit	Field Mnemonic	Type	Reset Value	Description
0	regbo_pclk_trc_enable	RW	0	Trace Enable bit. Set to '1' to enable tracing from the PCIe macro. This signal also acts as a clock gating signal, set to '0' when not tracing from the PCIe macro to save power.
01:03	Reserved	RW	0	Reserved, but implemented register(s) for future use.
04:07	regbo_pclk_trc_sel(0:3)	RW	0	4 bit field used to select 1 of 16 possible groups to place on the trace mux output *pcie_etu_trc_vec(0:87).*
04:63	Reserved	RO	0	Reserved.

Table 4-36. REGB Trace Groups

Groups	Name	Description	Pages
0	PCIe TLP Inbound #1	Inbound Receive TLP Header (Format #1)	281
1	PCIe TLP Inbound #2	Inbound Receive TLP Header (Format #2)	282
2	PCIe TLP Outbound #1	Outbound Transmit TLP Header (Format #1)	282
3	PCIe TLP Outbound #2	Outbound Transmit TLP Header (Format #2)	283
4	tldlpo_dl_tracerddata RX	tldlpo_dl_tracerddata and DL_MON RX Interface	283
5	tldlpo_dl_tracerddata TX	tldlpo_dl_tracerddata and DL_MON TX Interface	283
6 to 15	unused	unused group, all zeros trace output	

Table 4-37. Trace Group: PCIe TLP Inbound #1

Bit	Facility	Comment
0	tlif_trace_rxhdr_val	
001:007	tlif_trace_rxhdr(001 to 007)	Byte 0 (first reserved bit replaced with tlif_trace_rxhdr_val).
008:015	tlif_trace_rxhdr(016 to 023)	Byte 2 (we skip Byte 1, it only has traffic class info in it).
016:023	tlif_trace_rxhdr(024 to 031)	Byte 3.
024:031	tlif_trace_rxhdr(032 to 039)	Byte 4.
032:039	tlif_trace_rxhdr(040 to 047)	Byte 5.
040:047	tlif_trace_rxhdr(048 to 055)	Byte 6.
048:055	tlif_trace_rxhdr(056 to 063)	Byte 7.
056:063	tlif_trace_rxhdr(064 to 071)	Byte 8.

Table 4-37. Trace Group: PCIe TLP Inbound #1

Bit	Facility	Comment
064:071	tlif_trace_rxhdr(072 to 079)	Byte 9.
072:079	tlif_trace_rxhdr(080 to 087)	Byte 10.
080:087	tlif_trace_rxhdr(088 to 095)	Byte 11.

Table 4-38. Trace Group: PCIe TLP Inbound #2

Bit	Facility	Comment
0	tlif_trace_rxhdr_val	
001:007	tlif_trace_rxhdr(001 to 007)	Byte 0 (first reserved bit replaced with tlif_trace_rxhdr_val).
008:015	tlif_trace_rxhdr(016 to 023)	Byte 2 (we skip Byte 1, it only has traffic class info in it).
016:023	tlif_trace_rxhdr(024 to 031)	Byte 3.
024:031	tlif_trace_rxhdr(032 to 039)	Byte 4.
032:039	tlif_trace_rxhdr(040 to 047)	Byte 5.
040:047	tlif_trace_rxhdr(048 to 055)	Byte 6.
048:055	tlif_trace_rxhdr(056 to 063)	Byte 7.
056:063	tlif_trace_rxhdr(096 to 103)	Byte 12.
064:071	tlif_trace_rxhdr(104 to 111)	Byte 13.
072:079	tlif_trace_rxhdr(104 to 111)	Byte 14.
080:087	tlif_trace_rxhdr(120 to 127)	Byte 15.

Table 4-39. Trace Group: PCIe TLP Outbound #1

Bit	Facility	Comment
0	tlif_trace_txhdr_val	
001:007	tlif_trace_txhdr(001 to 007)	Byte 0 (first reserved bit replaced with tlif_trace_rxhdr_val).
008:015	tlif_trace_txhdr(016 to 023)	Byte 2 (we skip Byte 1, it only has traffic class info in it).
016:023	tlif_trace_txhdr(024 to 031)	Byte 3.
024:031	tlif_trace_txhdr(032 to 039)	Byte 4.
032:039	tlif_trace_txhdr(040 to 047)	Byte 5.
040:047	tlif_trace_txhdr(048 to 055)	Byte 6.
048:055	tlif_trace_txhdr(056 to 063)	Byte 7.
056:063	tlif_trace_txhdr(064 to 071)	Byte 8.
064:071	tlif_trace_txhdr(072 to 079)	Byte 9.
072:079	tlif_trace_txhdr(080 to 087)	Byte 10.
080:087	tlif_trace_txhdr(088 to 095)	Byte 11.

Table 4-40. Trace Group: PCIe TLP Outbound #2

Bit	Facility	Comment
0	tlif_trace_txhdr_val	
001:007	tlif_trace_txhdr(001 to 007)	Byte 0 (first reserved bit replaced with tlif_trace_rxhdr_val).
008:015	tlif_trace_txhdr(016 to 023)	Byte 2 (we skip Byte 1, it only has traffic class info in it).
016:023	tlif_trace_txhdr(024 to 031)	Byte 3.
024:031	tlif_trace_txhdr(032 to 039)	Byte 4.
032:039	tlif_trace_txhdr(040 to 047)	Byte 5.
040:047	tlif_trace_txhdr(048 to 055)	Byte 6.
048:055	tlif_trace_txhdr(056 to 063)	Byte 7.
056:063	tlif_trace_txhdr(096 to 103)	Byte 12.
064:071	tlif_trace_txhdr(104 to 111)	Byte 13.
072:079	tlif_trace_txhdr(104 to 111)	Byte 14.
080:087	tlif_trace_txhdr(120 to 127)	Byte 15.

Table 4-41. Trace Group: tldlpo_dl_tracerddata RX

Bit	Facility	Comment
000:063	tldlpo_dl_tracerddata(63:0)	
64	tldlpo_pl_inbandpresence	
65	tldlpo_pl_linkup	
66	TL_EC10_LINKACTIVE	
67	'0'	Unused, always zero.
068:069	DL_MON_RXDLLP(1:0)	
070:071	DL_MON_RXTLP(1:0)	
72	'0'	Unused, always zero.
073:083	DL_MON_RXTLPSIZE(10:0)	
84	all zeros	Unused, always zero.
085:087	DL_MON_RXRECEIVERERROR(2:0)	

Table 4-42. Trace Group: tldlpo_dl_tracerddata TX

Bit	Facility	Comment
000:063	tldlpo_dl_tracerddata(63:0)	
64	tldlpo_pl_inbandpresence	
65	tldlpo_pl_linkup	
66	TL_EC10_LINKACTIVE	
67	'0'	Unused, always zero.

Table 4-42. Trace Group: tldlpo_dl_tracerddata TX

Bit	Facility	Comment
69	DL_MON_TXDLLP	
70	'0'	Unused, always zero.
71	DL_MON_TXTLP	
72	DL_MON_TXTLPNULLIFY	
073:083	DL_MON_TXTLPSIZE(10:0)	
084:087	all zeros	Unused, always zero.

4.6.3.25 PCIe - Misc. Strapping Register

This register is related to PCI Express functionality.

Mnemonic PCIE_MISC_STRAP

Address Offset 0x1B30

Bit	Field Mnemonic	Type	Reset Value	Description
00:20	Reserved	RW	0	Reserved, but implemented register(s) for future use.
21	pcie_err_kill_link_en	RW	1	Force the link to be 'killed' when an error class Fatal (FAT) is detected in the REGB Error Status Registers. This will force the link to drop when a REGB fatal error is detected.
22	pcie_err_link_down_kill_en	RW	1	Force the link to be 'killed' when the PCIe Link Down error bit is set in the REGB Error Status Registers. This will force the link to remain in a downed state until the error is cleared by firmware.
23:31	Reserved	RW	0	Reserved, but implemented register(s) for future use.
32:63	Reserved	RO	0	Reserved.

4.6.4 Error Registers

Each of the error status bits are assigned a class. A class represents the recommended method of how the error should be handled and which signal will be asserted to notify components outside the REGB and PCIe macro blocks when the error occurs. Refer to the following enable registers that represent the three error classes, Informative (INF), Endpoint Recoverable (ERC), and Fatal (FAT):

- REGB Error INF Enable Register (INF),page 290.
- REGB Error ERC Enable Register (ERC), page 291.
- REGB Error FAT Enable Register (FAT), page 291.

4.6.4.1 REGB Error Status Register

This register contains error status information for the REGB register block.

Mnemonic REGB_ESR

Address Offset 0x1C00

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
0	REGB Internal Register Parity Error	RW1CHS	0	A parity error occurred in one of the internal registers: REGB/PCIE specific.	Fatal
1	PBL Internal Register Parity Error	RW1CHS	0	A parity error occurred in one of the internal registers: REGB/PBL specific.	Fatal
2	Invalid Address Decode Error	RW1CHS	0	An address was decoded that targeted no defined registers in REGB.	INF
3	Register Access Invalid Address+Size Error	RW1CHS	0	A register access was attempted with an invalid address+size. PCI CFG accesses must be 4-byte size and 4-byte address aligned. All other register addresses must be 8-byte size and 8-byte address aligned.	Fatal
4	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
5	Register State Machine or Other Internal Error	RW1CHS	0	An internal and fatal error was detected in the REGB control logic. bus_err_2nd_req_e bus_err_rsp_no_req_e	Fatal
6	PCI CFG Core Registers Parity Error	RW1CHS	0	A parity error was detected in the PCI CFG space registers.	Fatal
7	Register access to CFG core while in reset error.	RW1CHS	0	An attempt was made to access registers in the CFG space regs while it was held in reset. The REGB logic will return all ones on a read and flag this error. Refer to *PCIe - Core Reset Register* on page 257.	INF

1. Bit 9, PCIe Link Up is classified as INF. However, Power Systems design does not want to be notified when this occurs. So, the interrupt enable bit is turned off in the PHB4 initialization sequence for this bit.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
8	PCIE Link Down	RW1CHS	0	This bit detects the high to low transition of the TL_EC10_LINKACTIVE signal. This indicates the link has gone down after it had been active. See the pcie_err_link_down_kill_en enable bit in the register on *PCIe - Misc. Strapping Register* on page 284. This enable bit will force the link to be held in a down state as long as this error bit is a '1'.	Fatal
9	PCIE Link Up	RW1CHS	0	This bit detects the low to high transition of the TL_EC10_LINKACTIVE signal. This indicates the link came up and has trained and exchanged flow control credits successfully. Note: This bit was added for the ICB usage. The firmware can require this indication for its cable pull recovery scenarios.	INF[1]
10	PCIE Link Auto Bandwidth Event Status	RW1CHS	0	Asserted when the TLDLP core reports a Link Autonomous Bandwidth event. This event occurs with software/firmware instructs the TLDLP core to perform a dynamic link speed change event. (tldlpo_tl_ec10_linkactive and SYS_AUTOLINKSPEEDEN and tldlpo_dl_ec10_linkautobw)	INF
11	PCIE Link BW Management Event Status (Silent Retrain)	RW1CHS	0	Asserted when the TLDLP core reports Link Bandwidth Management event. This event can occur when the TLDLP core in the root complex detects reliability issues on the link (or a lane simply drops) and must change the link speed or width by doing a 'silent' retrain of the PCIe link to obtain a more stable link operation. This often degrades the link speed and/or width from the original values on the initial link train. Software needs to notified in such cases to be aware of a change in the link health. (tldlpo_tl_ec10_linkactive and tldlpo_dl_ec10_linkbwmgmt)	INF
12:24	Reserved	RO	0	Reserved	Fatal
25	PBL Error Trap: INF Error	RW1CHS	0	PBL Error Trap Registers signaled an INF error. Refer to the PBL Error Trap and INF Enable Registers.	INF
26	PBL Error Trap: ERC Error	RW1CHS	0	PBL Error Trap Registers signaled an ERC error. Refer to the PBL Error Trap and ERC Enable Registers.	INF
27	PBL Error Trap: FAT Error	RW1CHS	0	PBL Error Trap Registers signaled a FAT error. Refer to the PBL Error Trap and FAT Enable Registers.	Fatal

1. Bit 9, PCIe Link Up is classified as INF. However, Power Systems design does not want to be notified when this occurs. So, the interrupt enable bit is turned off in the PHB4 initialization sequence for this bit.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
28	tldlpo_dl_mon_rxreceivererror(0)	RW1CHS	0	DL_MON_RXRECEIVERERROR provides additional information about unexpected events detected by the PHY and by this core when the core is evaluating the receiver information for packets. Bit [0] is driven to '1' for one cycle when the PHY reported an unexpected event on one or more Lanes with the RxStatus and RxValid signals. Some examples of PHY-reported events follow: • An RxStatus of 3'b100 indicates that the PHY detected an 8-bit/10-bit decode error (when the link is operating at 2.5 GT/s or 5.0 GT/s). • An RxValid transition from '1' to '0' indicates that the PHY detected a loss of Symbol or Block alignment.	INF
29	tldlpo_dl_mon_rxreceivererror(1)	RW1CHS	0	Bit [1] is driven to '1' for one cycle when a Lane-to-Lane deskew error is detected.	INF
30	tldlpo_dl_mon_rxreceivererror(2)	RW1CHS	0	Bit [2] is driven to '1' for one cycle when the Link is operating at 8.0 GT/s or 16.0 GT/s and the core detects that an Ordered Set Block was received without an EDS Token preceding it or that a Data Block was received following an EDS Token.	INF
31	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
32	DL_EC08_BADDLLP	RW1CHS	0	TLDLP core detected a bad DLLP packet. This error is considered correctable on PCIe. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
33	DL_EC08_BADTLP	RW1CHS	0	TLDLP core detected a bad TLP packet. This error is considered correctable on PCIe. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
34	DL_EC08_DLLPE	RW1CHS	0	TLDLP core detected a Data Link Layer Protocol Error. This error is considered uncorrectable on PCIe. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
35	DL_EC08_RECEIVERERROR	RW1CHS	0	TLDLP core detected a Receiver Error on the link. This error is considered correctable on PCIe. This is a typical error in the case where there is noise, signal integrity, or other transmission line issues with the PCIe link. Refer to *PCIe - DLP Error Counters Register* on page 272. See error bits tldlpo_dl_mon_rxreceivererror(2:0) above.	INF
36	DL_EC08_ REPLAYROLLOVER	RW1CHS	0	TLDLP core detected a wrap of the REPLAY_NUM counter. The TLDLP will initiate a Link Retrain after this condition occurs. This error is considered correctable on PCIe. This is the case where TLP is getting repeatedly NAK'd at the link level because the target sees bad LCRC on the packet every time it is received. This error indicates that the TLDLP has attempted to resend or replay the packet too many times and had yet to obtain a positive ACK from the target before giving up. This error is often in the presence of excessive noise and other signal integrity issues on the link.	INF

1. Bit 9, PCIe Link Up is classified as INF. However, Power Systems design does not want to be notified when this occurs. So, the interrupt enable bit is turned off in the PHB4 initialization sequence for this bit.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
37	DL_EC08_REPLAYTIMEOUT	RW1CHS	0	This signal is asserted for one PCLK cycle when the TLDLP core has not received an Ack or Nak DLLP for the REPLAY_TIMER timeout period defined in the PCI Express Base Specification. The timeout period varies with the negotiated Link width, and the CFG_EC08_MAX_PAYLOAD_SIZE, CFG_EC10_EXTENDEDSYNCH, and SYS_LOCALNFTS inputs.	INF
38	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
39	DL_INTERNALERROR	RW1CHS	0	Logical OR of all internal error bits detected in the TLDLP core. An error is reported through the REGB Error Status Registers, 0x1AA0-0x1AA8. See those register for the specific error that was reported. tldlpo_dl_internalerror	Fatal
40	DL_LB_ERROR	RW1CHS	0	This signal is asserted any time an error is detected in the received loopback data stream for any lane.	INF
41	DL_RX_MALFORMED	RW1CHS	0	TLDLP core indicates it received a TLP that was malformed.	Fatal
42	DL_RX_NULLIFY	RW1CHS	0	TLDLP core indicates it received a TLP that was nullified by the transmitter.	INF
43	DL_RX_OVERFLOW	RW1CHS	0	Asserted to indicate that the TLP in-progress has overflowed its receiver buffer as determined by the available receiver credits.	Fatal
44	DL_TX_CORRERROR	RW1CHS	0	TLDLP core detects a correctable ECC error on the Replay Buffer read data.	INF
45	DL_TX_UNCORRERROR	RW1CHS	0	TLDLP core detects an un-correctable ECC error on the Replay Buffer read data.	Fatal
46	TL_EC08_FCPE	RW1CHS	0	This signal is asserted for one PCLK cycle when a flow control protocol error is detected. The following conditions are flow control protocol errors: • Reception of a credit update of more than 2047 data credits or 127 header credits. • Reception of a non-infinite credit update when the type was initialized as infinite.	Fatal
47	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal
48	Replay ECC Correctable Error (CE)	RW1CHS	0	Replay wrapper logic detects a correctable ECC error on the Replay Buffer read data. Note: This will typically be asserted before or coincident with the DL_TX_CORRERROR bit from the TLDLP core.	INF
49	Replay ECC UnCorrectable Error (UE)	RW1CHS	0	Replay wrapper logic detects an un-correctable ECC error on the Replay Buffer read data. Note: This will typically be asserted before or coincident with the DL_TX_UNCORRERROR bit from the TLDLP core.	Fatal
50	Bad DLLP Error Count Saturated	RW1CHS	0	The Bad DLLP Error Counter value saturated at 0xFF. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
51	Bad TLP Error Count Saturated	RW1CHS	0	Bad TLP Error Counter value saturated at 0xFF. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
52	Receiver Error Count Saturated	RW1CHS	0	Receiver Error Counter value saturated at 0xFF. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF

1. Bit 9, PCIe Link Up is classified as INF. However, Power Systems design does not want to be notified when this occurs. So, the interrupt enable bit is turned off in the PHB4 initialization sequence for this bit.

Bit	Field Mnemonic	Type	Reset Value	Description	Error Class
53	DLLPE Error Count Saturated	RW1CHS	0	DLLPE Error Counter value saturated at 0xFF. Refer to *PCIe - DLP Error Counters Register* on page 272.	INF
54:57	Reserved	RO	0	Reserved.	Fatal
58	pbl_ptl_dl_al_rx_initcredit_p_e	RW1CHS	0	PBL to PTL credit signal parity error. Checks parity on these signals: • pbl_ptl_dl_al_rx_initcreditvalid_vc0 • pbl_ptl_dl_al_rx_initcredit_vc0(59:0)	Fatal
59	pbl_ptl_dl_al_rx_updatecredit_p_e	RW1CHS	0	PBL to PTL credit signal parity error. Checks parity on these signals: • pbl_ptl_dl_al_rx_updatecreditvalid_vc0(2:0) • pbl_ptl_dl_al_rx_updatecredit_vc0(59:0)	Fatal
60	PTL Core DLIF Protocol Error	RW1CHS	0	The PTL Core detected a protocol error on its DLIF interface.	Fatal
61	PTL Core TLIF Protocol Error	RW1CHS	0	The PTL Core detected a protocol error on its TLIF interface.	Fatal
62	PTL Core Internal Parity Error	RW1CHS	0	An internal parity error detected in the PTL core. pt_err_p_fe_rx_dat_par_e pt_err_p_fe_rx_int_par_e pt_err_p_fe_tx_dat_ctl_par_e pt_err_p_fe_tx_dat_queuesize_par_e pt_err_p_fe_tx_int_par_e	Fatal
63	Reserved	RW1CHS	0	Reserved, but implemented register(s) for future use.	Fatal

1. Bit 9, PCIe Link Up is classified as INF. However, Power Systems design does not want to be notified when this occurs. So, the interrupt enable bit is turned off in the PHB4 initialization sequence for this bit.

4.6.4.2 REGB First Error Status Register

This register contains first error status information. The register bits correspond directly with the bits defined in the REGB Error Status Register. If there are multiple error bits set in the Error Status Register, this register will show which error occurred first chronologically. It is possible to have multiple bits set in this register if several errors occur on the same clock cycle. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic REGB_FESR

Address Offset 0x1C08

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *REGB Error Status Register* on page 285.	ROHS	0	Bits correspond to the error that occurred first.

4.6.4.3 REGB Error Injection Register

This register controls error injection for the individual error bits defined in the Error Status Register. Writing bits to a '1' in this register will inject the error corresponding to the bit that is written. The bits will automatically clear to '0' after the error is injected. The corresponding bit in the Error Status Register is set automatically when the error occurs.

Mnemonic REGB_EINJ

Address Offset 0x1C10

Bit	Field Mnemonic	Type	Reset Value	Description
00:47	Reserved	RO	0	Reserved; read zeros.
48	regbo_replay_err_inj_ce	RWH	0	Inject a Correctable ECC error (single bit flip) into the Replay Buffer. Note: There must be outbound PCIe packets otherwise an injection will not occur and no errors will be flagged.
49	regbo_replay_err_inj_ue	RWH	0	Inject an Uncorrectable ECC error (double bit flip) into the Replay Buffer. Note: There must be outbound PCIe packets otherwise an injection will not occur and no errors will be flagged.
50:63	Reserved	RO	0	Reserved; read zeros.

4.6.4.4 REGB Error INF Enable Register

This register enables specific error bits to assert the Informative 'INF' output signals on the REGB block.

Mnemonic REGB_ERR_INF_EN

Address Offset 0x1C20

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *REGB Error Status Register* on page 285.	RW	0	Bits correspond to the errors that will set the following signals on an error: pcie_etu_regb_err_inf.

4.6.4.5 REGB Error ERC Enable Register

This register enables specific error bits to assert the Informative 'DEV' output signals on the REGB block.

Mnemonic REGB_ERR_ERC_EN

Address Offset 0x1C28

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *REGB Error Status Register* on page 285.	RW	0	Bits correspond to the errors that will set the following signals on an error: pcie_etu_regb_err_erc.

4.6.4.6 REGB Error FAT Enable Register

This register enables specific error bits to assert the informative 'FAT' output signals on the REGB block.

Note: The Fatal 'FAT' output signals will feed into the logic that will 'kill' the PCIe link. When the fatal signals assert it will bring the PCIe link down.

Mnemonic REGB_ERR_FAT_EN

Address Offset 0x1C30

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	See *REGB Error Status Register* on page 285.	RW	0	Bits correspond to the errors that will set the following signals on an error: pcie_etu_regb_err_fat.

4.6.4.7 REGB Error Log Register 0

This register contains error log information for errors that are active in the Error Status Register. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic REGB_ELR_0

Address Offset 0x1C40

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log 0	ROHS	0	Error log contents.

4.6.4.8 REGB Error Log Register 1

This register contains error log information for errors that are active in the Error Status Register. A write of any value to this register will set the entire contents of the register to all zeros.

Mnemonic REGB_ELR_1

Address Offset 0x1C48

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Log 1	ROHS	0	Error log contents.

4.6.4.9 REGB Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It is used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic REGB_ESMR

Address Offset 0x1C50

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	Error Status Mask	RWS	0	Each mask bit corresponds to an error bit in the REGB Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error.

4.6.4.10 REGB First Error Status Mask Register

This register is used to prevent errors from setting their corresponding bits in the Error Status Register. It should be used to disable rogue errors or for debug purposes only. This register is not required to be accessed during initialization unless there is an issue with a particular error bit.

Mnemonic REGB_FESMR

Address Offset 0x1C58

Bit	Field Mnemonic	Type	Reset Value	Description
00:63	First Error Status Mask	RWS	0	Each mask bit corresponds to an error bit in the REGB First Error Status Register. If a bit in the mask register is set to a '1' it will prevent the corresponding error from being logged in the status register. It does not prevent the side effects of the error. In addition to masking the status register, this mask will prevent an error from capturing logging information in the error log registers.

4.7 ETU/RSB UV Registers

This section describes the internal configuration registers specific to a PHB4. These registers are used for initialization, control, and debug of the PHB4 function. This section also describes the Ultra-Visor accessible register set. These registers are not accessible from the chip MMIO address space. They are only accessible through the SCOM interface. Refer to the SCOM interface registers for details.

The Ultra-visor registers are intended to be only accessible by the Ultra-visor firmware, but the PHB4 hardware does not enforce Ultra-visor only access to this registers. The protection is enforced in the Power nest hardware outside the PHB4.

Note: The SEID checking function and the UV registers are non-functional in POWER9/Nimbus DD1 (vA4.1). The UV registers have no effect on the logic.

Note: These registers have been updated for POWER9 DD2 (vA4.2). They are functional and support the updated function for secure address checking.

4.7.1 ETU/RSB UV Register Address Map

Table 4-43. ETU/RSB UV Register Address Map

Offset	Description	Page
0x2000	UV - Secure Address Exclude CMP/MSK Register	295
0x2008	Reserved (unused, will return all 1’s when read)	
0x2010	UV - Secure Address Include CMP Register	296
0x2018	UV - Secure Address Include MSK Register	297
0x2020 to 0x2FF8	Reserved (unused, will return all 1’s when read)	

Note: The offsets listed here are for internal use only. These registers are not accessible via the MMIO address space and are only accessible via the SCOM interface and its register space.

4.7.2 UV Register Set

4.7.2.1 UV - Secure Address Exclude CMP/MSK Register

Ultra-visor specific logic configuration register. This register should be programmed to prevent access to specific system address ranges. Commands matching this range are not allowed to be issued over the AIB interface.

Secure Exclude Check Example:

```
CMP(8:23) = 0xAF0
MSK(8:23) = 0x0FF0
TestAddr(8:23) = 0xAAF9 (this is always a system address, never a PCIe address or other address)
and_mask(8:23) = (0xAAF9 and 0x0FF0) = 0x0AF0
xor(8:23) = (0x0AF0 xor 0x0AF0) = 0x0000
Secure Range Hit = not or_reduce(0x0000) = '1' (TestAddr(8:23) is within the secure range)
```

Note the bit range 08:63 correlates to the 56-bit system address range. The CMP/MSK bits match one for one with the system address bits. The range checks only compare a small subset of the most significant bits of the system address bit range.

Mnemonic UV_SEC_EXCL

Address Offset 0x2000

Bit	Field Mnemonic	Type	Reset Value	Description
0	Enable Bit	RW	0	Enable/disable bit: • Enabled = 1 • Disabled = 0 (default) All include commands pass with no check.
01:07	Reserved	RO	0	Reserved.
08:11	CMP(08:11)	RO	0	Compare value, hard-coded and always zeros.
12:51	CMP(12:51)	RW	0	Compare value, programmable bits. Address is aligned on a 4K page boundary.
52:63	Reserved	RO	0	Reserved.

4.7.2.2 UV - Secure Address Include CMP Register

Ultra-visor specific logic configuration register.

This register should be programmed to grant specific commands access to specific system address ranges. If enabled, those commands must match the defined range in this register else they cannot be issued over the AIB interface.

Secure Include Check Example:

```
CMP(8:51) = 0x03_4567_8000_0 (from register 0x2010)
MSK(8:51) = 0x0F_FFFF_F000_0 (from register 0x2018)
TestAddr(8:51) = 0x23_4567_8FEE_D (this is always a system address, never a PCIe address or other address)
and_mask(8:51) = (0x23_4567_8FEE_D and 0x0F_FFFF_F000_0) = 0x03_4567_8000_0
xor(8:51) = (0x03_4567_8000_0 xor 0x03_4567_8000_0) = 0x00_0000_0000_0
Secure Range Hit = not or_reduce(0x00_0000_0000_0) =  e1 f (TestAddr(8:51) is within the secure range)
```

Note the bit range 08:63 correlates to the 56-bit system address range. The CMP/MSK bits match one for one with the system address bits. The include range checks compare the more bits than the exclude checks, up to the specific 4K page boundary if necessary.

Mnemonic UV_SEC_INCL_CMP

Address Offset 0x2010

Bit	Field Mnemonic	Type	Reset Value	Description
o	Enable Bit	RW	0	Enable/disable bit: • Enabled = 1 • Disabled = 0 (default) All include commands pass with no check.
00:07	Reserved	RO	0	Reserved.
08:11	CMP(08:11)	RO	0	Compare value, hard-coded and always zeros.
12:51	CMP(12:51)	RW	0	Compare value, programmable bits. Address is aligned on a 4K page boundary.
52:63	Reserved	RO	0	Reserved.

4.7.2.3 UV - Secure Address Include MSK Register

Ultra-visor specific logic configuration register. This register should be programmed to grant specific commands access to specific system address ranges. If enabled, those commands must match the defined range in this register else they cannot be issued over the AIB interface.

Mnemonic UV_SEC_INCL_MSK

Address Offset 0x2018

Bit	Field Mnemonic	Type	Reset Value	Description
00:07	Reserved	RO	0	Reserved.
08:11	MSK(08:11)	RO	0	Mask value, hard-coded and always zeros.
12:51	MSK(12:51)	RW	0	Mask value, programmable bits. Address is aligned on a 4K page boundary.
52:63	Reserved	RO	0	Reserved.

4.8 PHB4 Register Initialization

The AIB interface is used for all initialization sequence steps unless stated otherwise. System specific value (<sys specific value>) must be supplied by the software or firmware.

RdL access type means to read in a loop, look for data value specified (can have a mask too).

mXXXX_XXXX_XXXX_XXXX is a bitwise mask for altering data. A '1' indicates a data bit to modify, '0' means do not modify the data bit.

4.8.1 PHB4 Initialization Sequence

Table 4-44. PHB4 Initialization Sequence Table (Page 1 of 9)

Step	Acc	Addr	Write Data	Comment
Init_1	nop	n/a	n/a	Apply PHB4 synchronous resets.
• Assumption at the start of the sequence is the PHB4 has been reset via the PB register that drives the resets inputs on the PHB4 unit and all other PB registers have been set up as required. See the PEC/PB specification for details				
Init_2	Write	x0C18	0xFFFFFFFF_FFFFFFFF	LEM Error Mask Register.
• Disable all LEM errors until the end of the init sequence.				
Init_3	nop	x0868	0xFFFFFFFF_FFFFFFFF	PHB4 – TCE Tag Enable Register.
• Skip these steps if default value is acceptable. • Enable all TCE tags. • TCE tags are used for every internal transaction as well as TCE read requests.				
Init_4	Write	x1A00	0x42100000_40000000 (x08) 0x44100000_40000000 (x16)	PCIe - System Configuration Register.
• Skip these steps if default value is acceptable. • This is the default value out of reset. This register can be modified to change the following fields if needed: • Bits 04:09 - SYS_EC0C_MAXLINKWIDTH[5:0] - The default link width is x16. This can be reduced to x1, x4, or x8, if needed. • Bits 10:12 - SYS_EC04_MAX_PAYLOAD[2:0] - The default max payload size is 512-byte. This can be reduced to the allowed ranges from 128-byte to 4KB if needed.				
Init_5	Write	x1A10	0x60000000_00000000	PCIE - Core Reset Register
• This will deassert reset for the PCI CFG core only. • This step done here so the sticky status bits for link status can be cleared in a later step and before the TLDLP core reset is released and the link begins to train.				
Init_6	Write	x1AD0	<sys specific value>	PCIE - DLP LANEEQCONTROL Register 0 (Gen 3, 8Gbps).
Init_7	Write	x1AD8	<sys specific value>	PCIE - DLP LANEEQCONTROL Register 1 (Gen 3, 8Gbps).
Init_8	Write	x1AE0	<sys specific value>	PCIE - DLP LANEEQCONTROL Register 2 (Gen 3, 8Gbps).
Init_9	Write	x1AE8	<sys specific value>	PCIE - DLP LANEEQCONTROL Register 3 (Gen 3, 8Gbps).
Init_10	Write	x1AF0	<sys specific value>	PCIE - DLP P1620-LANEEQCONTROL Register 0 (Gen 4, 16Gbps) (vA4.2).
Init_11	Write	x1AF8	<sys specific value>	PCIE - DLP P1620-LANEEQCONTROL Register 1 (Gen 4, 16Gbps) (vA4.2).
Init_12	nop	n/a	n/a	Removed (vA4.2).
Init_13	nop	n/a	n/a	Removed (vA4.2).
• These steps can be skipped unless there are recommended values to write to these registers. • These registers drive the initial values for the lane equalization presets for Gen 3,4 equalization. • The values in this register are somewhat arbitrary. The values will eventually be determined empirically during system bring-up and are highly dependent on the physical layer and transmission line parameters. • These registers have settings for 16 lanes, therefore a x8 configuration would not need to set values in all four registers. **Note:** These preset values are for the physical lane number. When lane swapping is enabled, the physical/logical lane relationship is swapped, this has to be taken in account when setting these values.				
Init_14	Write	x1078	0x0400FE07	EC – Link Control / Status 2 Register (size=4B).

Table 4-44. PHB4 Initialization Sequence Table (Page 2 of 9)

Step	Acc	Addr	Write Data	Comment
• Clear link training status bits. • Set bits 3:0 = 'sys_ec0c_maxlinkspeed(3:0)' = 0x4 (Gen 4).				
Init_15	Write	x1A10	0x18000000_00000000	PCIE - Core Reset Register.
• This will deassert reset for the PCIe cores, including the PCS/PMA (PHY/HSS) macros. • The TLDLP core will begin link training shortly after this register is written. • Firmware will verify in a later step whether the PCIe link has been established.				
Init_16	RdL	x1A40	x0000_0000_0000_0000 m0000_0200_0000_0000	PCIE - DLP Training Control Register.
• Firmware must poll this register to check bit 22 (DL_PGRESET). • DL_PGRESET should be polled until it is inactive with a value of '0'. The recommended polling frequency is once every 1ms. Firmware should poll at least 200 attempts before giving up. • When '0' it indicates the IOP/PMA/PHY core is operational and out of reset.				
Init_17	Write	x0810	<sys specific value>	PHB - Control Register.
• Skip this step if default values are acceptable. • Interrupt Page Size Bit, bit 11. firmware can select 4K (default) or 64K page size for interrupts.				
Init_18	Write	x0300	<sys specific value>	Interrupt Notify Base Address Register.
Init_19	Write	x0308	<sys specific value>	Interrupt Notify Base Index Register.
• Set up interrupt notification base registers.				
Init_20	Write	x0168	<sys specific value>	RTT Base Address Register.
Init_21	Write	x0188	<sys specific value>	PELT-V Base Address Register.
• Set up various memory BAR tables.				
Init_22	Write	x01A0	<sys specific value>	M32 Starting Address Register.
• Firmware must determine the correct values to write to this register. • See M32 Address Translation, *Figure 3-2* on page 39.				
Init_23	Write	x01A8	<sys specific value>	PEST Base Address Register.
• Set up PEST table in memory. • This is the start location of the error log table.				
Init_24	Write	x01B0	<sys specific value>	CRW Base Address Register.
• Skip this step if NBW support is not enabled. • Firmware must determine the value to write to this register.				
Init_25	Write	x01C0	<sys specific value>	ASN Compare/Mask Register.
• Skip this step if ASN support is not enabled. • Firmware must determine the value to write to this register.				
Init_26	Write	x01C8	<sys specific value>	CAPI Compare/Mask Register.
• Skip this step if CAPI support is not enabled. • Firmware must determine the value to write to this register.				
Init_27	Write	x01F0	<sys specific value>	PCIe Outbound Request Upper Address Register.
• Firmware must determine the value to write to this register. • See also M64 Address Translation *Figure 3-3* on page 40.				
Init_28	Write	x0160	x00020000_00000000	PHB4 Configuration Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 3 of 9)

Step	Acc	Addr	Write Data	Comment
• Bit 8 - 32-bit MSI enable bit. Set this bit to '1' to enable 32-bit MSI interrupts (default not enabled). • Bit 14 - 64-bit MSI enable bit. Set this bit to '1' to enable 64-bit MSI interrupts (default not enabled).				
Init_29	nop	n/a	n/a	IODA Table SRAM Initialization Wait State.
• Firmware (and simulation) must wait at least 512 cycles (256ns in this design) before attempting any access of the IODA tables. This is to ensure the post-reset initialization state machines have had adequate time to zero out the SRAMs, otherwise bad side effects can occur.				
Init_30	Write	x0220	x00070000_00000000	IODA Table Address Register.
Init_31	Write	x0228	x00000000_00000000	IODA Table Data Register.
• The writes will write all zeros to the first entry of the PESTA table as part of the workaround.				
Init_32	Write	x0220	x80020000_00000000	IODA Table Address Register.
• This sets up the MIST Table (MIST) to be written in 'auto-increment' mode.				
Init_33	Write	x0228 (1024,x0)	<sys specific value>	IODA Table Data Register.
• These are writes to MIST entries 0 to 1023 (PHBX16 = 1024, PHBX08 = 512). • Firmware must determine the correct information to write. • The P/Q state bits should be set to zeros for the initial state. • The PE Number(0:11) bits must be set to the correct PE for the interrupt source.				
Init_34	Write	x0220	x80060000_00000000	IODA Table Address Register.
• This sets up the Migration Register Table (MRT) to be written in 'auto-increment' mode.				
Init_35	Write	x0228 (16,x0)	<sys specific value>	IODA Table Data Register.
• These are writes to MRT entries 0 to 15 (PHBX16 = 16, PHBX08 = 8). • Firmware must determine the correct information to write for memory migration pointers. **Note:** Firmware can update these values during runtime as well.				
Init_36	Write	x0220	x80090000_00000000	IODA Table Address Register.
This sets up the TVT Table (TVT) to be written in 'auto-increment' mode.				
Init_37	Write	x0228 (512,x0)	<sys specific value>	IODA Table Data Register.
• These are writes to TVT entries 0 to 511 (PHBX16 = 512, PHBX08 = 256). • Firmware must determine the correct information to write for TVT entries.				
Init_38	Write	x0220	x80100000_00000000	IODA Table Address Register.
• This sets up the MMIO BAR Table (MBT) to be written in 'auto-increment' mode.				
Init_39	Write	x0228 (32,x0)	<sys specific value>	IODA Table Data Register.
• These are writes to MBT entries 0 to 31 (PHBX16 = 32, PHBX08 = 16). • Firmware must determine the correct information to write to this register.				
Init_40	Write	x0220	x80110000_00000000	IODA Table Address Register.
• This sets up the MMIO Domain Table (MDT) to be written in 'auto-increment' mode.				
Init_41	Write	x0228 (512,x0)	<sys specific value>	IODA Table Data Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 4 of 9)

Step	Acc	Addr	Write Data	Comment
• These are writes to MDT entries 0 to 511 (PHBX16 = 512, PHBX08 = 256). • Each entry holds the PE number corresponding the memory segment of its associated BAR region. • Firmware must determine the correct information to write to this register.				
Init_42	Write	x1AA0	0xFFFFFFFF_FFFFFFFF	PCIE - DLP Error Log Register 1.
Init_43	Write	x1AA8	0xFFFFFFFF_FFFFFFFF	PCIE - DLP Error Log Register 2.
Init_44	Write	x1AB0	0xFFFFFFFF_FFFFFFFF	PCIE - DLP Error Status Register.
Init_45	Write	x1AB8	0x00000000_00000000	PCIE - DLP Error Counters Register.
• Clear all sticky error bits for TLDLP core error vectors. • Clear all link error counters.				
Init_46	Write	x1018	<sys specific value>	PCI - Primary / Secondary / Subordinate Bus Number Register (Type-1) (size=4-byte).
• Firmware must determine the correct information to write to this register. • Example typical simulation value is 0x0001FF00.				
Init_47	Write	x101C	0x000000FF	PCI - IO Base/Limit and Secondary Status Register (size=4-byte).
• Clear error status bits.				
Init_48	Write	x1050	0x40000F00	EC - Device Control / Status Register (size=4-byte).
• Enable a max payload size of 512B. • Clear any detected errors.				
Init_49	Write	x1070	0x05000000	EC - Device Control / Status Register 2 (size=4-byte).
• Set PCIe completion timeout for MMIO Loads to 16ms. • See *PBL - Timeout Control Register* on page 240.				
Init_50	Write	x1104	0xFFFFFFFF	AER - Uncorrectable Error Status Register (size=4-byte).
• AER - Uncorrectable Error Status Register. • Clear all error bits.				
Init_51	Write	x1108	0x00D00000	AER - Uncorrectable Error Mask Register (size=4-byte).
• Set bit 12, Poisoned TLP Mask. • Set bit 14, Completion Timeout Mask. • Set bit 15, Completer Abort Mask. • Clear all other error mask bits.				
Init_52	Write	x1110	0xFFFFFFFF	AER - Correctable Error Status Register (size=4-byte).
• Clear all error bits.				
Init_53	Write	x1118	0x40010000	AER - Capabilities and Control Register (size=4-byte).
• Enable ECRC generation bit 6. • Enable ECRC checking bit 8.				
Init_54	Write	x1130	0xFFFFFFFF	AER - Root Error Status Register (size=4-byte).
• Clear all error bits.				
Init_55	Write	x1900	0xFFFFFFFF_FFFFFFFF	PBL - Error Status Register.
Init_56	Write	x1908	0x00000000_00000000	PBL - First Error Status Register.
Init_57	Write	x1920	0x00000000_4D1780F8	PBL - Error INF Enable Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 5 of 9)

Step	Acc	Addr	Write Data	Comment
Init_58	Write	x1928	0x00000000_00000000	PBL - Error ERC Enable Register.
Init_59	Write	x1930	0xFFFFFFFF_B2F87F07 (vA4.1) 0xFFFFFFFF_B2E87F07 (vA4.2)	PBL - Error FAT Enable Register.
Init_60	Write	x1940	0x00000000_00000000	PBL - Error Log Register 0.
Init_61	Write	x1948	0x00000000_00000000	PBL - Error Log Register 1.
Init_62	Write	x1950	0x00000000_00000000	PBL - Error Status Mask Register.
Init_63	Write	x1958	0x00000000_00000000	PBL - First Error Status Mask Register.
PBL Error Trap Registers: • Clear all sticky error bits. • Enable class specific error bits enables. • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_64	Write	x1C00	0xFFFFFFFF_FFFFFFFF	REGB Error Status Register.
Init_65	Write	x1C08	0x00000000_00000000	REGB First Error Status Register.
Init_66	Write	x1C20	0x2130006E_FCA8BC00	REGB Error INF Enable Register.
Init_67	Write	x1C28	0x00000000_00000000	REGB Error ERC Enable Register.
Init_68	Write	x1C30	0xDE8FFF91_035743FF	REGB Error FAT Enable Register.
Init_69	Write	x1C40	0x00000000_00000000	REGB Error Log Register 0.
Init_70	Write	x1C48	0x00000000_00000000	REGB Error Log Register 1.
Init_71	Write	x1C50	0x00000000_00000000	REGB Error Status Mask Register.
Init_72	Write	x1C58	0x00400000_00000000	REGB First Error Status Mask Register.
REGB Error Trap Registers: • Clear all sticky error bits. • Enable class specific error bits enables. • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_73	Write	x0D00	0xFFFFFFFF_FFFFFFFF	TXE Error Status Register.
Init_74	Write	x0D08	0x00000000_00000000	TXE First Error Status Register.
Init_75	Write	x0D18	0xFFFFFF0F_FFFFFFFF	TXE Error LEM Report Enable Register.
Init_76	Write	x0D28	0x0000400A_00000000	TXE Error EEH Freeze Enable Register.
Init_77	Write	x0D30	0xDFF7FD01_F7DDFFF0 0xDFF7FF0B_F7DDFFF0 (CAPI)	TXE Error AIB Fence Enable Register.
Init_78	Write	x0D40	0x00000000_00000000	TXE Error Log Register 0.
Init_79	Write	x0D48	0x00000000_00000000	TXE Error Log Register 1.
Init_80	Write	x0D50	0x00000000_00000000	TXE Error Status Mask Register.
Init_81	Write	x0D58	0x00000000_00000000	TXE First Error Status Mask Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 6 of 9)

Step	Acc	Addr	Write Data	Comment
TXE Error Trap Registers: • Clear all sticky error bits. • Enable error bits that are allowed to set LEM FIR bits. • Enable error bits that are allowed to set the EEH freeze state (ER class). • Note: When a CAPI device is attached to the PHB4, for all ER class errors the corresponding AIB fence enable bit should also be set. This forces the AIB to fence for these errors which eventually leads to a PEC freeze. This allows the error to be detected in the CAPP unit. The CAPP unit cannot detect when an ER error occurs. • Enable error bits that are allowed to fence the AIB bus (Fatal class). • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_82	Write	x0D80	0xFFFFFFFF_FFFFFFFF	RXE_ARB Error Status Register.
Init_83	Write	x0D88	0x00000000_00000000	RXE_ARB First Error Status Register.
Init_84	Write	x0D98	0xFFFFFFFF_FBFFFFFF	RXE_ARB Error LEM Report Enable Register.
Init_85	Write	x0DA8	0xC00000B8_01000060 (vA4.1) 0xC00008B8_01000060 (vA4.2)	RXE_ARB Error EEH Freeze Enable Register.
Init_86	Write	x0DB0	0xFBFFD703_FE7FBF8F (vA4.1) 0x3BFFD703_FE7FBF8F (vA4.2) 0xFBFFD7BB_FF7FBFEF (CAPI)	RXE_ARB Error AIB Fence Enable Register.
Init_87	Write	x0DC0	0x00000000_00000000	RXE_ARB Error Log Register 0.
Init_88	Write	x0DC8	0x00000000_00000000	RXE_ARB Error Log Register 1.
Init_89	Write	x0DD0	0x00000000_00000000	RXE_ARB Error Status Mask Register.
Init_90	Write	x0DD8	0x00000000_00000000	RXE_ARB First Error Status Mask Register.
RXE_ARB Error Trap Registers: • Clear all sticky error bits. • Enable error bits that are allowed to set LEM FIR bits. • Enable error bits that are allowed to set the EEH freeze state (ER class). • Note: When a CAPI device is attached to the PHB4, for all ER class errors the corresponding AIB fence enable bit should also be set. This forces the AIB to fence for these errors which eventually leads to a PEC freeze. This allows the error to be detected in the CAPP unit. The CAPP unit cannot detect when an ER error occurs. • Enable error bits that are allowed to fence the AIB bus (Fatal class). • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_91	Write	x0E00	0xFFFFFFFF_FFFFFFFF	RXE_MRG Error Status Register.
Init_92	Write	x0E08	0x00000000_00000000	RXE_MRG First Error Status Register.
Init_93	Write	x0E18	0xFFFFFFFF_FFFFFFFF	RXE_MRG Error LEM Report Enable Register.
Init_94	Write	x0E28	0x00006000_00000000	RXE_MRG Error EEH Freeze Enable Register.
Init_95	Write	x0E30	0xFFFF9EFF_FF7FFF57 0xFFFFFEFF_FF7FFF57 (vA4.1) 0xFFFFFEFF_FF7FFF57 (CAPI)	RXE_MRG Error AIB Fence Enable Register.
Init_96	Write	x0E40	0x00000000_00000000	RXE_MRG Error Log Register 0.
Init_97	Write	x0E48	0x00000000_00000000	RXE_MRG Error Log Register 1.
Init_98	Write	x0E50	0x00000000_00000000	RXE_MRG Error Status Mask Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 7 of 9)

Step	Acc	Addr	Write Data	Comment
Init_99	Write	x0E58	0x00000000_00000000	RXE_MRG First Error Status Mask Register.
RXE_MRG Error Trap Registers: • Clear all sticky error bits. • Enable error bits that are allowed to set LEM FIR bits. • Enable error bits that are allowed to set the EEH freeze state (ER class). • NOTE: When a CAPI device is attached to the PHB4, for all ER class errors the corresponding AIB fence enable bit should also be set. This forces the AIB to fence for these errors which eventually leads to a PEC freeze. This allows the error to be detected in the CAPP unit. The CAPP unit cannot detect when an ER error occurs. • Enable error bits that are allowed to fence the AIB bus (Fatal class). • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_100	Write	x0E80	0xFFFFFFFF_FFFFFFFF	RXE_TCE Error Status Register.
Init_101	Write	x0E88	0x00000000_00000000	RXE_TCE First Error Status Register.
Init_102	Write	x0E98	0xFFFFFFFF_FFFFFFFF	RXE_TCE Error LEM Report Enable Register.
Init_103	Write	x0EA8	0x60000000_00000000 (vA4.1) 0x60000000_C0000000 (vA4.2)	RXE_TCE Error EEH Freeze Enable Register.
Init_104	Write	x0EB0	0x9FAEFFAF_3FFFFFFF 0xFFAEFFAF_FFFFFFFF (CAPI)	RXE_TCE Error AIB Fence Enable Register.
Init_105	Write	x0EC0	0x00000000_00000000	RXE_TCE Error Log Register 0.
Init_106	Write	x0EC8	0x00000000_00000000	RXE_TCE Error Log Register 1.
Init_107	Write	x0ED0	0x00000000_00000000	RXE_TCE Error Status Mask Register.
Init_108	Write	x0ED8	0x00000000_00000000	RXE_TCE First Error Status Mask Register.
RXE_TCE Error Trap Registers: • Clear all sticky error bits. • Enable error bits that are allowed to set LEM FIR bits. • Enable error bits that are allowed to set the EEH freeze state (ER class). • NOTE: When a CAPI device is attached to the PHB4, for all ER class errors the corresponding AIB fence enable bit should also be set. This forces the AIB to fence for these errors which eventually leads to a PEC freeze. This allows the error to be detected in the CAPP unit. The CAPP unit cannot detect when an ER error occurs. • Enable error bits that are allowed to fence the AIB bus (Fatal class). • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_109	Write	x0C80	0xFFFFFFFF_FFFFFFFF	PHB4 Error Status Register.
Init_110	Write	x0C88	0x00000000_00000000	PHB4 First Error Status Register.
Init_111	Write	x0C98	0xFFFFFFFF_FFFFFFFF	PHB4 Error LEM Report Enable Register.
Init_112	Write	x0CA8	0x00000040_00000000	PHB4 Error EEH Freeze Enable Register.
Init_113	Write	x0CB0	0x35777033_FF000000 0x35777073_FF000000 (CAPI)	PHB4 Error AIB Fence Enable Register.
Init_114	Write	x0CC0	0x00000000_00000000	PHB4 Error Log Register 0.
Init_115	Write	x0CC8	0x00000000_00000000	PHB4 Error Log Register 1.
Init_116	Write	x0CD0	0x00000000_00000000	PHB4 Error Status Mask Register.
Init_117	Write	x0CD8	0x00000000_00000000	PHB4 First Error Status Mask Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 8 of 9)

Step	Acc	Addr	Write Data	Comment
PHB4 Error Trap Registers: • Clear all sticky error bits. • Enable error bits that are allowed to set LEM FIR bits. • Enable error bits that are allowed to set the EEH freeze state (ER class). • NOTE: When a CAPI device is attached to the PHB4, for all ER class errors the corresponding AIB fence enable bit should also be set. This forces the AIB to fence for these errors which eventually leads to a PEC freeze. This allows the error to be detected in the CAPP unit. The CAPP unit cannot detect when an ER error occurs. • Enable error bits that are allowed to fence the AIB bus (Fatal class). • Clear all sticky error logging information. • Clear all sticky error mask bits. • The error status mask bits should always be set to all zeros unless there is a particular error bit that needs to be masked.				
Init_118	Write	x0C00	0x00000000_00000000	LEM FIR Accumulator Register.
Init_119	Write	x0C30	0xFFFFFFFF_FFFFFFFF	LEM Action 0 Register.
Init_120	Write	x0C38	0xFFFFFFFF_FFFFFFFF	LEM Action 1 Register.
Init_121	Write	x0C40	0x00000000_00000000	LEM WOF Register.
• LEM Error Registers: • Clear all sticky LEM FIR bits. • Set up all LEM actions to 'No Action'; Action[0:1] = '11'. • Clear all sticky LEM WOF bits.				
Init_122	RdL	x1A30	x0080_0000_0000_0000 m0080_0000_0000_0000	PCIE – Link Management Register.
• Firmware must poll this register to check bit 8 to see if the PCIe link is up (TL_EC10_LINKACTIVE). • Recommended firmware polling frequency of once every 2ms (link configuration hardware loop time) with a timeout of about 200 ms. • If firmware times-out polling for the linkup status it is recommended that the link be reset (via PCIe Core Reset Register x1A10, bit 3 (hpo_perst_n) and another attempt be made. At least 3 attempts should be made before giving up. • PCIe Core Reset Register, bit 3 is the PERST# bit (PCI Fundamental Reset). Firmware must set and clear this bit, it does not automatically clear itself. • MMIO Stores to the link are silently dropped by the PBL core if the link is down. • MMIO Loads to the link will be dropped by the PBL core and will eventually time-out and will return an all ones response if the link is down.				
Init_123	Rd	x1A40	n/a	PCIE - DLP Training Control Register.
• Check negotiated link width in bits 30:35 (DL_EC10_NEGLINKWIDTH(5:0)). • Check negotiated/current link speed in bits 36:39 (DL_EC10_CURRENTLINKSPEED(3:0))				
Init_124	Write	x1830	<sys specific value>	PBL - NBW Compare/Mask Register.
• Skip this step if NBW decoding should not be enabled. • Firmware must determine the value to write to this register.				
Init_125	Write	x1004	0x06000000	PCI - Command/Status Register (size=4B).
• Enable Memory space bit 1. • Enable Bus Master bit 2. • Refer to the PCIe specification for the description of these bits. **Note:** These bits mean different things on Root Ports versus Endpoints.				
Init_126	Write	x0CA0	0xCA8880CC_00000000	PHB4 Error System Interrupt Enable Register.
Init_127	Write	x0D20	0x2008400E_08200000	TXE Error System Interrupt Enable Register.
Init_128	Write	x0DA0	0xC40028FC_01804070	RXE_ARB Error System Interrupt Enable Register.
Init_129	Write	x0E20	0x00006100_008000A8	RXE_MRG Error System Interrupt Enable Register.

Table 4-44. PHB4 Initialization Sequence Table (Page 9 of 9)

Step	Acc	Addr	Write Data	Comment
Init_130	Write	x0EA0	0x60510050_00000000 (vA4.1) 0x60510050_C0000000 (vA4.2)	RXE_TCE Error System Interrupt Enable Register.
Init_131	Write	x0C18	0x00000000_00000000	LEM Error Mask Register.
• Enable error trap interrupts for INF and ER errors, set bits to '1'. • Clear all mask bits for Fatal class errors as well as Reserved bits, clear bits to '0'. • Clear all LEM mask bits to allow LEM WOF to capture first error of any error class.				
Init_132	Write	x0218	0xF0000000_00000000	Speculation Control Register.
• Set bits 00:03 to enable all DMA read/write TLP address speculation.				
Init_133	Write	x0878	0x00181500_00200000	PHB4 - Timeout Control Register 1.
• BLIF Forward Progress timeout set to ~268-402 ms. • MMIO Read Request timeout set to ~33-50 ms. • CFG Request timeout disabled.				
Init_134	Write	x0880	0x00001817_00000000	PHB4 - Timeout Control Register 2.
• TCE Request timeout set to ~268-402 ms. • AIB Transmit timeout set to ~134-201 ms.				
Init_135	Write	x1810	0x20150000_00000000	PBL - Timeout Control Register.
• PBL: Outbound Transmission Timeout Value set to ~134-201 ms.				
Init_136	nop	n/a	n/a	End of Initialization Sequence.

4.9 PHB4 Operating Sequences

The AIB interface is to be used for all sequence steps unless stated otherwise. System specific value (<sys specific value>) must be supplied by the software or firmware.

RdL access type means to read in a loop and look for the data value specified (can have a mask too).

mXXXX_XXXX_XXXX_XXXX is a bitwise mask for altering data. A '1' indicates a data bit to modify, '0' means do not modify the data bit.

4.9.1 Hot-Reset Sequences

The Hot-Reset Sequences can be part of a larger sequence. *Table 4-45* and *Table 4-46* describe only what is necessary to send a hot-reset from the PHB4 toward the endpoints.

Table 4-45. Hot-Reset (Assert) sequence table

Step	Acc	Addr	Write Data	Comment
Init_1	Write	x1B30	0x00000000_00000000	PCIE – Misc. Strapping Register.
• Clear bits 21:22, pcie_err_kill_link_en and pcie_err_link_down_kill_en.				
Init_2	Write	x1C20	0x00000000_00000000	REGB Error INF Enable Register.
Init_3	Write	x1C28	0x00000000_00000000	REGB Error ERC Enable Register.
Init_4	Write	x1C30	0x00000000_00000000	REGB Error FAT Enable Register.
• Disable all PCIE/REGB error status indicators that might trigger an interrupt from the PHB/ETU.				
Init_5	Write	x103C	x00004000 m00004000	PCI - Bridge Control Register (size=4-byte).
• Set bit 22 = 1 to turn on the hot-reset.				
Init_6	nop	n/a	n/a	Delay for 1 second.
• Firmware must wait 1 second for the hot-reset to take affect before proceeding.				
Init_7	nop	n/a	n/a	End of Hot-Reset (Assert) Sequence.

Table 4-46. Hot-Reset (De-Assert) sequence table (Page 1 of 2)

Step	Acc	Addr	Write Data	Comment
Init_1	Write	x103C	x00000000 m00004000	PCI - Bridge Control Register (size=4-byte).
• Clear bit 22 = 0 to turn off the hot-reset.				
Init_2	Write	x101C	0x000000FF	PCI - IO Base/Limit and Secondary Status Register (size=4-byte).
Init_3	Write	x1050	x00000F00 m00000F00	EC - Device Control / Status Register (size=4-byte).
Init_4	Write	x1104	0xFFFFFFFF	AER - Uncorrectable Error Status Register (size=4-byte).
Init_5	Write	x1110	0xFFFFFFFF	AER - Correctable Error Status Register (size=4-byte).
Init_6	Write	x1130	0xFFFFFFFF	AER - Root Error Status Register (size=4-byte)
Init_7	Write	x1900	0xFFFFFFFF_FFFFFFFF	PBL - Error Status Register.
Init_8	Write	x1908	0x00000000_00000000	PBL - First Error Status Register.
Init_9	Write	x1940	0x00000000_00000000	PBL - Error Log Register 0.
Init_10	Write	x1948	0x00000000_00000000	PBL - Error Log Register 1.
Init_11	Write	x1C00	0xFFFFFFFF_FFFFFFFF	REGB Error Status Register.
Init_12	Write	x1C08	0x00000000_00000000	REGB First Error Status Register.
Init_13	Write	x1C40	0x00000000_00000000	REGB Error Log Register 0.

Table 4-46. Hot-Reset (De-Assert) sequence table (Page 2 of 2)

Step	Acc	Addr	Write Data	Comment
Init_14	Write	x1C48	0x00000000_00000000	REGB Error Log Register 1.
• Clear any spurious errors. • Clear error log contents.				
Init_15	Write	x1C20	<init sequence value>	REGB Error INF Enable Register.
Init_16	Write	x1C28	<init sequence value>	REGB Error ERC Enable Register.
Init_17	Write	x1C30	<init sequence value>	REGB Error FAT Enable Register.
• Re-enable PCIE/REGB error status indicators based on init sequence values.				
Init_18	Write	x1B30	0x00000600_00000000	PCIE – Misc. Strapping Register.
• Set bits 21:22, pcie_err_kill_link_en and pcie_err_link_down_kill_en.				
Init_19	nop	n/a	n/a	End of Hot-Reset (De-Assert) Sequence.

4.9.2 PCIe Power Reduction Sequences

The following sequences describe sequences that can help reduce the power consumption in the PCIe stack. The main areas of control are:

- PCIe Link Width
- PCIe Link Speed

The largest power reduction will come from reducing the link width. The smaller, narrower the width the better. As a general rule, reduction in the link speed is not as effective at reducing power consumption as compared to reducing the link width. These reduction options can be set statically at initialization time after reset or dynamically at runtime while traffic is running. The register access sequences that enable these reduction options will be described in the following sections.

The trade-off with the power reduction options is the overall performance degradation of the port and link. Reducing the link width and speed will reduce the port's link bandwidth significantly.

Note the PCIe architected link power states are not supported in the design and should not be enabled. These are the Active State Power Management (ASPM) link states such as L0s, L1, L2, and L3.

4.9.2.1 Static Power Reduction Settings

The static power settings are configured during the PHB4 initialization sequence. They do not require a separate defined sequence. All the options are controlled via register 0x600, PCIe - System Configuration Register. Firmware can control the following parameters in this register where it appears in the PHB4 initialization sequence:

- SYS_EC0C_MAXLINKWIDTH[5:0], bits 04:09
- SYS_EC0C_MAXLINKSPEED[3:0], bits 32:35

4.9.2.2 Dynamic Link Width Change Sequence

Firmware can increase or decrease the link width while traffic is running. Reducing the link width will save power, but it can be increased as well to restore it to a prior higher level. The width can be changed to and from any width, up or down. It does not have to be changed to a consecutive amount up or down. Firmware can skip from x16 to x1, for example. *Table 4-47* describes the sequence for how this can be achieved. The sequence applies to any change to the link width, up or down.

Table 4-47. Dynamic link width change sequence table

Step	Acc	Addr	Write Data	Comment
Init_1	Write	x0630	x8100_0000_0000_0000 mBF00_0000_0000_0000	PCIE – Link Management Register.
• Set the new target link width in this step. • Set bit 0, SYS_CHANGELINKWIDTH. • Set bits 2:7, SYS_TARGETLINKWIDTH. The above example value is set for a x1 link.				
Init_2	RdL	x0630	x0040_0000_0000_0000 m0040_0000_0000_0000	PCIE – Link Management Register.
• Poll and check when bit 9 is set to '1', DL_WIDTHCHANGEPENDING.				
Init_3	RdL	x0630	x0000_0000_0000_0000 m0040_0000_0000_0000	PCIE – Link Management Register.
• Poll and check when bit 9 is clear to '0', DL_WIDTHCHANGEPENDING.				
Init_4	Rd	x0640	n/a	PCIE - DLP Training Control Register.
• Check bits 30:35 to verify new link width, DL_EC10_NEGLINKWIDTH(5:0).				
Init_5	nop	n/a	n/a	End of Dynamic Link Width Change Sequence.

4.9.2.3 Dynamic Link Speed Change Sequence

Firmware can increase or decrease the link speed while traffic is running. Reducing the link speed will save power, but it can be increased as well to restore it to a prior higher level. The speed can be changed to and from any value, up or down. It does not have to be changed to a consecutive amount up or down. Firmware can skip from 8G to 2.5G, for example. *Table 4-48* on page 309 describes the sequence for how this can be achieved. The sequence applies to any change to the link speed, up or down.

Table 4-48. Dynamic link speed change sequence table (Page 1 of 2)

Step	Acc	Addr	Write Data	Comment
Init_1	Read	x0630	x0000_0000_0000_0000 m0000_1000_0000_0000	PCIE – Link Management Register.
• Verify that bit 19 is '0', CFG_EC30_HWAUTO_SPEED_DISABLE. • If not '0', firmware must clear bit 5 in PCI CFG register 0x78, else this procedure will not work. • Refer to *EC - Link Control/Status 2 Register* on page 208.				
Init_2	Write	x0630	x0011_0000_0000_0000 m001F_0000_0000_0000	PCIE – Link Management Register.

Table 4-48. Dynamic link speed change sequence table (Page 2 of 2)

Step	Acc	Addr	Write Data	Comment
• Set the new target link speed in this step. • Set bit 11, SYS_AUTOLINKSPEEDEN. • Set bits 12:15, SYS_AUTOLINKSPEED[3:0]. The above example value is set for a 2.5G speed.				
Init_3	RdL	x0640	x0000_0000_0100_0000 m0000_0000_0F00_0000	PCIE - DLP Training Control Register.
• Poll bits 36:39 to verify new link speed, DL_EC10_CURRENTLINKSPEED(3:0). • In this example, the polling is checking for a change to a 2.5G speed.				
Init_4	nop	n/a	n/a	End of Dynamic Link Speed Change Sequence.

Glossary

Ack	Acknowledgment
ACM	Access Control Monitor
AER	Advanced error reporting
AIB	ASIC interconnect bus
ARI	Alternate Requester ID
ASB	Advanced system bus
ASN	ASB System Notification
ASPM	Active State Power Management
ATS	Address translation services
BAR	Base address register
BIST	Built-in self test
BLIF	Buffer Layer Interface
BT	Bar table
BW	Bandwidth
CA	Completer abort
CAM	Content-addressable memory
CAPI	Coherent Accelerator Processor Interface
CAPP	Coherent accelerator processor proxy
CE	Error correcting
CFG	Configuration
CFGOP	Configuraton operation
CI	Cache-inhibited
CMD	Command
CMP	Compare
CQ	Common queue block
CRC	Cyclic redundancy check
CRS	Root Control Register

CRW	Credit Return Write
DEC	Decode block
DLIF	Data Link Interface
DLL	Delay-locked loop
DLPAR	Dynamic logical partitioning
DMA	Direct memory access
Downbound	The direction and flow of transactions and data from the GX bus and the processor toward the I/O devices and the PCIe link.
DRS	Device Readiness Status
DSF	Downstream-facing
ECC	Error-correcting code
ECRC	End-to-end cyclic redundancy check
EEH	Extended Error Handling
eff	Efficiency
EIEOS	Electrical Idle Exit Ordered Set
EP	Endpoint
EQ	equalization process
ER	Endpoint recoverable
ESB	Event state buffer
ETU	Express transaction unit
FA	Fundamental Registers Set A
FIFO	First in, first out.
FIR	Fault Isolation Register
FMTC	Firmware Managed TCE Coherency
GT/s	Gigatransfers per second
HDL	Hardware description language
HV	Hypervisor
I/O	Input/output
ICB	Instruction cache block
ID	Identification

Inbound	The direction and flow of transactions and data from the PCIe link.
INF	Informative
IOA	I/O adapter
IODA	I/O device architecture
IOV	I/O virtualization
IPD	Inbound Posted Data
IST	Interrupt State Table
IVC	Interrupt vector cache
IVE	Interrupt vector entry
IVSE	Interrupt Virtualization Source Engine
LBE	last byte enable
LBS	least-significant byte
LCB	Local clock buffer
LCRC	Link cyclic redundancy check
LED	Light-emitting diode
LEM	Local error macro
LMB	Logical memory block
LRU	Least recently used
LSB	least-significant byte
LSI	Level sensitive interrupt
LXIVT	LSI XIVT Table
MBT	MMIO BAR Table
MDT	MMIO domain table
MIST	MSI Interrupt State Table
MMIO	Memory-mapped I/O
MRT	Migration register table
MSB	Most-significant byte
MSI	Message signalled interrupt
MSK	Mask

NAK	negative-acknowledgement
NBW	Non-Blocking Write
nop	No operation
OCF	Outbound Completion Forward
Outwardbound	The direction and flow of transactions and data to the PCIe link.
P2P	peer-to-peer
PAPR	Power Architecture Platform Reference
PB	Processor bus
PBL	Packet buffer layer
PCI	Peripheral Component Interface
PCIe	Peripheral Component Interface Express
PCLK	Processor clock
PCS	Physical coding sublayer
PCT	PCI completion timeout PCIe Completion Table
PE	Partionable endpoint
PEC	PCI Express Controller
PEEV	PE error vector table
PELTV	Partitionable-endpoint state table
PERR	Parity Error Enable
PERST	PCIe reset
PEST	PE state table
PHB	IBM® Power Systemsä Host Bridge
PHY	Physical layer
PM	Power Management
PMC	performance monitor counter
PME	Power Management Event
PTL	PCIE Transaction Layer
RAM	Random access memory

RAS	Reliability, availability, and serviceability
RBA	Reject bit array
RCAM	RID translation CAM
RID	Requestor ID
RO	Read-only
ROM	Read-only memory
RPN	Real page number
RRB	Remote Register Block
RSB	Register service block
RTC	RID translation cache
RTE	RID translation entry
RTT	RID translation table
RW	Read- write
RW1C	Read-Write 1 to Clear
RWOR	Read-Only, Write 1 to Set.
RX	Receive
RXCPLD	PCIe RX, Completion Data Buffer Size
RXE	Receive engine
RXNPH	PCIe RX, Nonposted Header Credits
RXPD	PCIe RX, Posted Data Credits
RXPH	PCIe RX, Posted Header Credits
SCOM	Serial communications
SEID	Secure Identifier
SERR	System error
SRAM	Static random-access memory
SSA	Serial Storage Architecture
SSC	Spread spectrum clocking
SUE	Special uncorrectable ECC
TB	Terabyte

TCAM	TCE CAM
TCE	Translation control entry
TDR	TCE data RAM
THASH	TCE Hash Table
TLDLP	Transaction, data link, and logical physical
TLP	Transaction layer packet
TTA	TCE table address
TVE	Translation validation entry
TVT	TCE validation table
TX	Transmit
TXCPLD	PCIe TX, Completion Data Buffer Size
TXCPLH	PCIe TX, Completion Slots
TXE	Transmit engine
TXPD	PCIe TX, Posted Data Buffer Size
TXPH	PCIe TX, Posted Header Slots
UE	Uncorrectable error
USF	Upstream-facing
Upwardbound	The direction and flow of transactions and data from the I/O devices and the PCIe link toward the GX bus and the processor.
UR	Unsupported request
UTL	Upper transaction layer
UV	Ultra-Visor
Verilog	Hardware description language standardized as IEEE 1364
VC	Virtual channel
VHDL	VHSIC hardware description language
VHSIC	Very High Speed Integrated Circuit
WO	Write only
WOF	Who’s on first