PCIe X1 IP Core - Lattice Radiant Software

User Guide

FPGA-IPUG-02091-1.3

December 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
AXI	Advanced Extensible Interface
ASPM	Active State Power Management
BAR	Base Address Register
DLLP	Data Link Layer Packet
DMA	Direct Memory Access
EP	Endpoint
FIFO	First In First Out
LMMI	Lattice Memory Mapped Interface
LTSSM	Link Training and Status State Machine
MSI	Message Signaled Interrupt
RTL	Register Transfer Language
PCIE	Peripheral Component Interconnect Express
PCS	Physical Coding Sublayer
PMA	Physical Medium Attachment
PM	Power Mangagement
RC	Root Complex
RP	Root Port
TLP	Transaction Layer Packet
UCFG	User Configuration Interface

1. Introduction

PCI Express is a high performance, fully scalable, well defined standard for a wide variety of computing and communications platforms. Being a packet based serial technology, PCI Express greatly reduces the number of required pins and simplifies board routing and manufacturing. PCI Express is a point-to-point technology, as opposed to the multi-drop bus in PCI. Each PCI Express device has the advantage of full duplex communication with its link partner to greatly increase overall system bandwidth. The basic data rate for a single lane is double that of the 32 bit/33 MHz PCI bus. A four lane link has eight times the data rate in each direction of a conventional bus.

The Lattice PCIe X1 Core provides a flexible, high-performance, easy-to-use Transaction Layer Interface to the PCI Express® Bus. The Lattice PCIe X1 Core implementation is a hardened IP with soft logic provided for interface conversion options. The hardened IP is an integration of PHY and Link Layer blocks from third party vendors (Analog Bits Inc., and Northwest Logic Inc.). Throughout this document, the Link Layer block refers to the entire logical block that consists of three sub layers namely, PHY logical layer (that is, LTSSM, PIPE, and other related function), Data Link Layer, and Transaction Layer. The PHY block refers to the PMA and PCS.

The Lattice PCIe X1 IP Core is supported in CrossLink™-NX and Certus™-NX FPGA device families and is available in Lattice Radiant™ Software.

1.1. Quick Facts

Table 1.1 provides quick facts about the Lattice PCIe X1 IP core.

Table 1.1. Lattice PCIe X1 IP Core Quick Facts

IP Requirements	Supported FPGA Families	CrossLink-NX, Certus-NX
Resource Utilization	Targeted Devices	LIFCL-40, LFD2NX-40
	Supported User Interfaces	APB, AHB-Lite, AXI4-Stream
	Resources	See Table 4.1.
Design Tool Support	Lattice Implementation	IP Core v1.1.x –Lattice Radiant Software 2.1
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro® for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide.

1.2. Features

Hard IP PHY Key features include:

- Transmitter
  - Configurable driver impedance, amplitude and 3-tap pre-emphasis
- Receiver
  - Configurable receiver impedance, Continuous Time Linear Equalizer (CTLE)gain, 1-Tap Decision Feedback Equalization (DFE), and support for equalizer adaptation.
  - Baud rate Eye Monitoring capability to map eye density at receiver post equalization
  - Bit skip feature to allow for adjusting received byte clock alignment
- PCS
  - Rate negotiation support
  - Selectable parallel data widths such as 5, 10 and 16
  - 8b/10b encoding at 2.5 Gbps and 5 Gbps
  - Test support features such as near-end loopback, PLL bypass modes, etc.
  - Protocol-compatible features such as LOS, squelch, power modes, etc
  - L1-substates and Special L1P2 support for PCIe2.0

Hard IP Link Layer Key features include:

- PCI Express™ Base Specification Revision 3.0 compliant including compliance with earlier PCI Express Specifications.
  - Backward compatible with PCI Express 2.x, 1.x
- x1 PCI Express Lane only
- 5.0 GT/s, and 2.5 GT/s line rate support
- Comprehensive application support:
  - Endpoint
  - Root Port
- Multi-Function support with 1-4 Physical Functions
- ECC RAM and Parity Data Path Protection
- 32-bit Core Data Width
- Complete error-handling support
  - AER, ECRC generation/checking, recovery from Parity and ECC errors
  - Supports detection of numerous optional errors, embedded simulation error checks/assertions
  - Simulation and hardware error injection features enable error testing
- Flexible core options allow for design complexity/feature tradeoffs:
  - Configurable Receive, Transmit, and Replay Buffer sizes
- Supports Polarity Inversion, Up/Down-configure, Autonomous Link Width/Speed changes
- Power Management
  - Supports L1, ASPM L0s, & ASPM L1
  - L1 PM Substates with CLKREQ
  - Power Budgeting
  - Dynamic Power Allocation
- Latency Tolerance Reporting
- Implements Type 0 Configuration Registers in Endpoint Mode
- Implements Type 1 Configuration Registers in Root Port Mode
  - Complete Root Port Configuration Register implementation
- Dual mode design supports EP or RP via register changes
- Easy to use
  - Decodes received packets to provide key routing information (BAR hits, Tag, and others.)
  - Implements all aspects of the required PCIe Configuration Space
  - Optionally consumes PCI Express Message TLPs or leaves them in band
  - Interfaces have consistent timing and function over all modes of operation
  - Provides a wealth of diagnostic information for superior system-level debug and link monitoring
- Implements all 3 PCI Express Layers (Transaction, Data Link, Physical)

Soft IP features include:

- AHB-Lite Data Interface Option
- AHB-Lite Data Interface Option with DMA support
- AXI4-Stream Data Interface Option
- APB Register Interface Option
- AHB-Lite Register Interface Option

1.3. Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

- _*n* are active low (asserted when value is logic 0)
- _*i* are input signals
- _*o* are output signals

2. Functional Description

2.1. Overview

2.1.1. Block Diagram

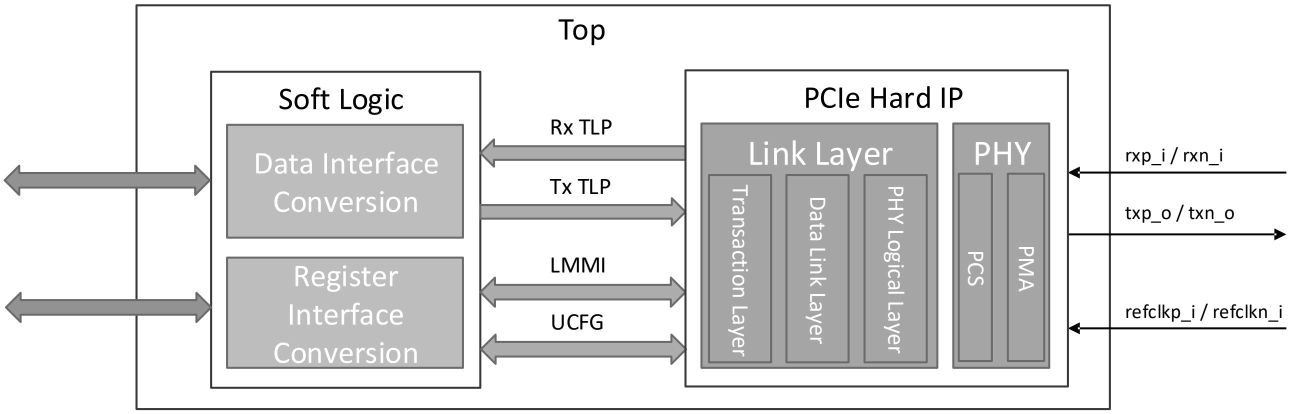


Figure 2.1. Lattice PCIe X1 Core Block Diagram

The Lattice PCIe X1 Core implements all three layers defined by the PCI Express Specification: Physical, Data Link, and Transaction. A soft logic is provided for optional interface conversion such as AHB-Lite, AXI4-stream, and APB. Details of the optional interface conversion are discussed in the section 2.1.10 and 0.

The Lattice PCIe X1 Core Hard IP implementation integrates the Northwest Logic's Expresso Core (Link Layer), and AnalogBits' PMA and PCS Core (PHY).

The Lattice PCIe X1 Hard IP has the following interfaces as shown in Figure 2.2. The details of each interface are discussed in the succeeding sections.

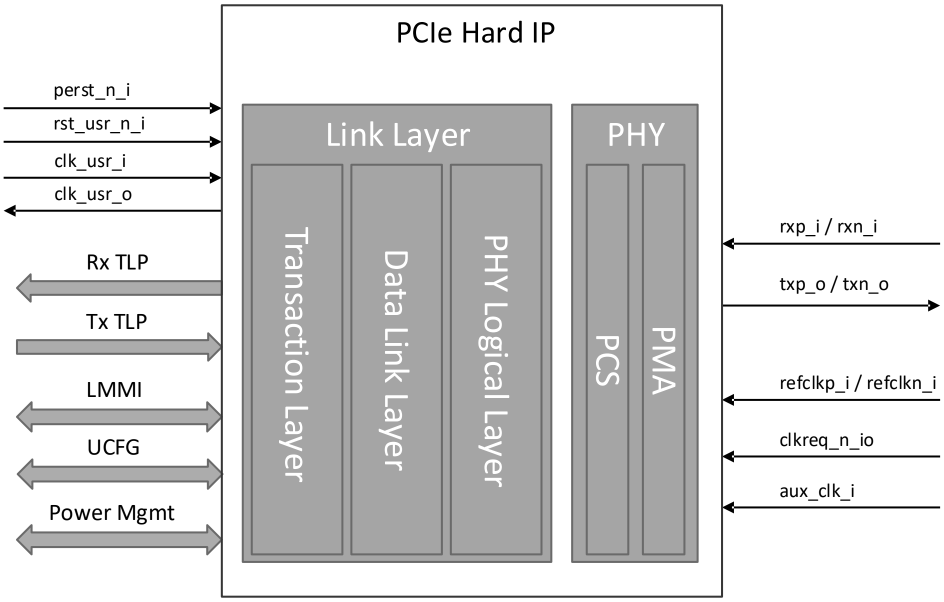


Figure 2.2. Lattice PCIe X1 Core Hard IP

- Clock and Reset Interface
  - The user domain interface can be clocked using the PHY PCLK output (clk_usr_i=clk_usr_o) or by user generated clock. The fundamental reset (perst_n_i) resets the core (PHY and Link layer blocks) except for the core configuration registers. Another reset (rst_usr_n_i) is provided to reset only the Link layer block.
- PHY Interface
  - High Speed Serial Interface supports maximum rate of 5GTs
- TLP Receive Interface
  - Receive TLPs from the PCIe link partner
  - High bandwidth interface
- TLP Transmit Interface
  - Transmit TLPs to the PCIe link partner
  - High bandwidth interface
- Power Management Interface
  - Ports for implementing power management capabilities
- UCFG – User Configuration Space Register Interface
  - Enables customers to access the PCIe Configuration Space Registers
- LMMI – Configuration and Status Register (CSR) Interface
  - This interface is used to write and read the Core configuration and status registers. A typical customer application requires changing only a small number of the default values such as Vendor ID, Device ID, and BAR configuration.

2.1.2. Endpoint/Root Port Configurations

The Lattice PCIe X1 Core supports Endpoint and Root Port applications:

- Endpoint
- PCI Express Interface for an I/O device (network/storage/data capture/etc.)
- Root Port
- Implements the top node of a PCI Express hierarchy; part of the Root Complex
- For use with an embedded processor running an Operating System

Root Port operation is detailed in Section 2.1.6.

The different applications supported by the Lattice PCIe X1 Core (gray blocks) are shown in Figure 2.3.

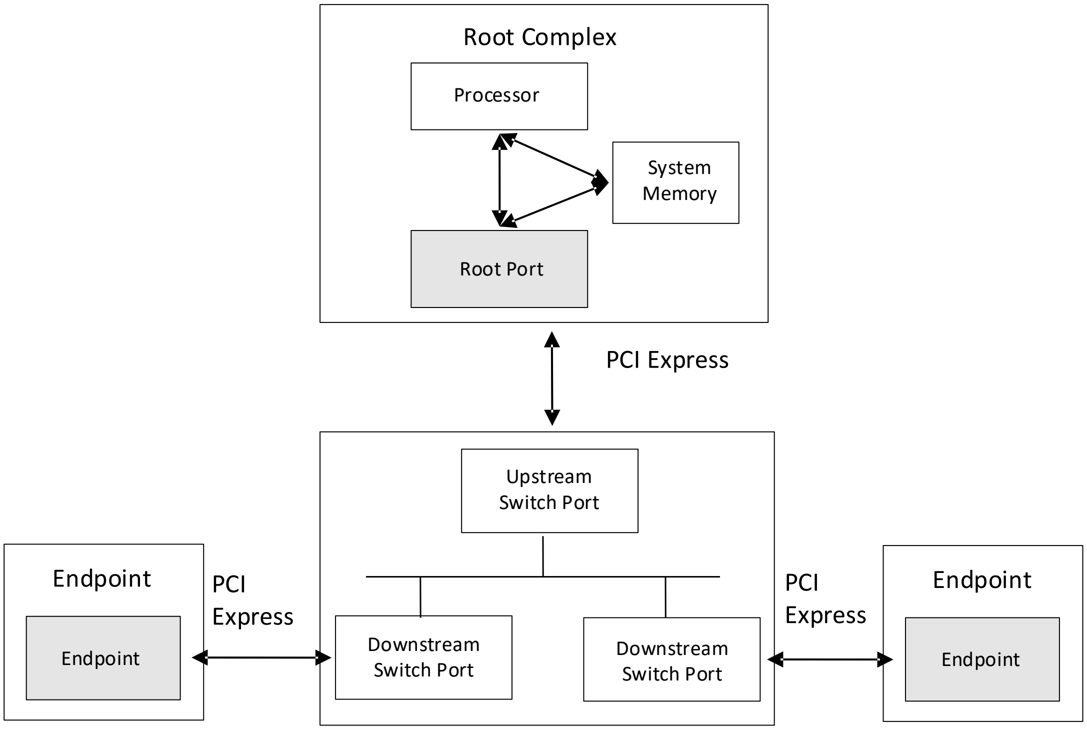


Figure 2.3. Endpoint and Root Port Applications

2.1.3. ECC and Parity Data Path Protection

The Lattice PCIe X1 Core protects the TLP data path with ECC and Parity Protection. This is implemented in the Hard IP block.

ECC is used to protect TLP data in the following data path RAMs:

- Replay Buffer
- Receive Buffer
- Transmit Buffer

The ECC implementation enables correction for 1-bit errors and detection for 2-bit errors. 8-bits of ECC information are included in the RAMs for each 64 (or fraction thereof) data bits.

Even (XOR) Parity (parity[i] = ^(data[((i+1)*8)-1:(i*8)]) is used to protect the data path. Parity provides detection for 1-bit errors (and other odd-bit errors). To enable continuous parity protection coverage, parity is passed through RAMs that are also protected by ECC.

For the receive data path, parity is generated for received TLPs prior to the removal and validation of the Link CRC (LCRC). Parity protection is thus overlapped with LCRC protection. Received TLP parity is passed with the associated received TLP (header and payload) bytes through the Receive Buffer and onto the user Transaction Layer Receive Interface. It is expected that parity is checked, and errors are handled, by the ultimate TLP consumer. Because TLP can have parity errors on any byte (toward the end of a longer TLP for instance), it is generally not possible to avoid processing the error TLP because the earlier portion of the TLP may already have been processed by the time that the error is detected. Applications that don't want to process TLPs with errors need to store and forward the TLP for processing only after inspecting the parity of all data bytes. If the core Transaction Layer detects a parity error while it is consuming a received TLP (Type 0 Configuration Read/Write, Malformed TLP, Message, etc.), the error is reported as an Uncorrectable Error (in AER capability) and the core discards the TLP without processing it.

For the transmit data path, parity is generated by the TLP source. For user TLPs (for example those transmitted on the core's Transaction Layer Transmit Interface), you provded parity along with the associated TLP (header and payload) bytes. The provided parity is kept with the associated data as it traverses the core. The parity is checked and discarded just after the TLP PCIe LCRC is generated. Parity protection is thus overlapped with LCRC protection including the associated PCIe replay mechanism. If the core detects a parity or uncorrectable ECC error during transmission of a TLP, the error is reported and the associated TLP is nullified (discarded) and not retransmitted. This is a serious error that must be handled by software. The TLP is discarded in order to not propagate the error and risk potential worse consequences in other components that would receive a TLP with known bit errors.

The core includes the ability to enable/disable the reporting and handling of ECC/Parity errors. Correctable errors (ECC 1-bit errors) are fixed when correction is enabled. Uncorrectable ECC/Parity errors in the transmit data path result in the associated TLP being discarded/nullified when error handling is enabled. While error handling can be disabled, this is not recommended as passing a known TLP with bad contents can result in a more serious error condition than discarding the TLP.

2.1.3.1. Uncorrectable Error Recovery

PCI Express includes the ability to nullify (cancel) a TLP transmission immediately after it finishes by inverting the LCRC and using End Bad (EDB) end framing instead of the normal TLP end framing. TLP may be desired to be nullified to reduce propagation, and potentially multiplying, the affects of the error. Nullified TLPs are generally not regenerated by the original TLP source (it is difficult for software to construct the missing TLP), so a fatal system error condition generally results whether the error TLP is nullified or not. When TLP are nullified due to errors, the core attempts to keep the transmit stream active so that software can be notified of the error using the standard in-band mechanisms (transmission of ERR_NFAT or ERR_FAT message for example).

TLPs are allocated a Sequence Number during transmission and the PCIe receiver only accepts TLPs in sequence order. When a TLP is nullified due to an uncorrectable error, the missing sequence number must be recovered, before the link can continue to transmit TLPs.

TLPs are allocated Virtual Channel Flow Control Credits when they are transmitted by the Transaction Layer. The PCI Express device receiving the TLP over the PCI Express link, frees the associated credits by sending Flow Control Update DLLPs. TLPs which are nullified due to uncorrectable ECC and Parity errors have been allocated credits by the Transaction Layer, which is never freed since the TLP is nullified and is not received by the Receiver (nullified TLP are discarded by the Receiver without affecting Flow Control Credits or Sequence Number).

Whenever a transmitted TLP is nullified due to an uncorrectable error, this causes the PCI Express link to be unable to process further TLPs. The sequence number and flow control credits that were allocated to the nullified TLP must be reclaimed before the link is repaired. The Lattice PCIe X1 Core contains logic to correct the link when TLPs are nullified due to uncorrectable errors.

Whenever an uncorrectable ECC or Parity error is detected, even when the link has been corrected for further transmission, it is recommended for software to cause the link to be reset. The TLP(s) which are discarded, due to uncorrectable errros, compromise the application and the application is typically unable to recover.

2.1.4. LTSSM State Definitions

2.1.4.1. Main LTSSM

The Lattice PCIe X1 Core follows the PCI Express specification for the Link Training and Status State Machine. However, to help hit higher frequencies the LTSSM is split into one Major State LTSSM state machine and several separate LTSSM sub-state machines with one sub-state state machine for each major state.

The Lattice PCIe X1 Core implements additional LTSSM sub-states that are necessary to meet PCIe Spec. LTSSM operation, but are not given an explicit sub-state in the PCI Express specification. Table 2.1 provides information about each state machine state

Table 2.1. LTSSM State Definitions

LTSSM Major State	LTSSM Sub-state	DESCRIPTION
0 – Detect	0 – DETECT_INACTIVE	The sub-state is "DETECT_INACTIVE" whenever the LTSSM Major State is not Detect.
	1 – DETECT_QUIET	Detect.Quiet
	2 – DETECT_SPD_CHG0	Detect.Quiet – Sub-state to change speed change back to 2.5G if needed – request PHY speed change
	3 – DETECT_SPD_CHG1	Detect.Quiet – Sub-state to change speed change back to 2.5G if needed – wait for speed change to complete
	4 – DETECT_ACTIVE0	Detect.Active – First Rx Detection
	5 – DETECT_ACTIVE1	Detect.Active – Wait 12mS between Rx Detection attempts
	6 – DETECT_ACTIVE2	Detect.Active – Second Rx Detection (if needed)
	7 – DETECT_P1_TO_P0	Detect.Active – Change PHY power state from P1 to P0 (inactive to active) if needed (ie. on Detect->Polling transition).
	8 – DETECT_P0_TO_P1_0	Change PHY power state from P0 to P1 (active to inactive) – Transmit Electrical Idle Ordered Sets to notify the link partner that we are going idle
	9 – DETECT_P0_TO_P1_1	Change PHY power state from P0 to P1 – Wait for TX Electrical Idle Ordered Set transit request made in DETECT_P0_TO_P1_0 to get transmitted at the output of the core
	10 – DETECT_P0_TO_P1_2	Change PHY power state from P0 to P1 – Wait for PHY to reach P1 state before continuing
1 – Polling	0 – POLLING_INACTIVE	The sub-state is "POLLING_INACTIVE" whenever the LTSSM Major State is not Polling.
	1 – POLLING_ACTIVE_ENTRY	Polling.Active – Entry to Polling.Active – State exists because in some cases the LTSSM must exit Polling without Tx of TS OS.
	2 – POLLING_ACTIVE	Polling.Active
	3 – POLLING_CFG	Polling.Configuration
	4 – POLLING_COMP	Polling.Compliance – Transmitting compliance pattern.
	5 – POLLING_COMP_ENTRY	Polling.Compliance entry state – directs a speed change via POLLING_COMP_EIOS, POLLING_COMP_EIOS_ACK, and POLLING_COMP_IDLE when necessary before going to POLLING_COMP.
	6 – POLLING_COMP_EIOS	Polling.Compliance – Transmit Electrical Idle Ordered Sets to notify the link partner that we are going idle
	7 – POLLING_COMP_EIOS_ACK	Polling.Compliance – Wait for the Electrical Idle Ordered Sets transmitted in POLLING_COMP_EIOS to exit the core
	8 – POLLING_COMP_IDLE	Polling.Compliance – Perform speed change now that link is idle.

LTSSM Major State	LTSSM Sub-state	DESCRIPTION
2 – Configuration	0 – CONFIGURATION_INACTIVE	The sub-state is “CONFIGURATION_INACTIVE” whenever the LTSSM Major State is not Configuration.
	1 – CONFIGURATION_US_LW_START	Acting as Upstream Port – Configuration.Linkwidth.Start
	2 – CONFIGURATION_US_LW_ACCEPT	Acting as Upstream Port -Configuration.Linkwidth.Accept
	3 – CONFIGURATION_US_LN_WAIT	Acting as Upstream Port -Configuration.Lanenum.Wait
	4 – CONFIGURATION_US_LN_ACCEPT	Acting as Upstream Port -Configuration.Lanenum.Accept
	5 – CONFIGURATION_DS_LW_START	Acting as Downstream Port -Configuration.Linkwidth.Start
	6 – CONFIGURATION_DS_LW_ACCEPT	Acting as Downstream Port -Configuration.Linkwidth.Accept
	7 – CONFIGURATION_DS_LN_WAIT	Acting as Downstream Port -Configuration.Lanenum.Wait
	8 – CONFIGURATION_DS_LN_ACCEPT	Acting as Downstream Port -Configuration.Lanenum.Accept
	9 – CONFIGURATION_COMPLETE	Configuration.Complete
	10 – CONFIGURATION_IDLE	Configuration.Idle
3 – L0	0 – L0_INACTIVE	The sub-state is “L0_INACTIVE” whenever the LTSSM Major State is not L0.
	1 – L0_L0	L0 – Link is in L0
	2 – L0_TX_EL_IDLE	Tx_L0s.Entry, L1.Entry, or L2.Entry – Transmit Electrical Idle Ordered Sets to notify the link partner that we are going idle – i.e. for preparing to enter low power states such as Tx_L0s, L1, and L2.
	3 – L0_TX_IDLE_MIN	Tx_L0s.Entry, L1.Entry, or L2.Entry – Guarantee the minimum Tx Elec Idle time when entering electrical idle and also require Rx EIOS to have been received when necessary.
4 – Recovery	0 – - RECOVERY_INACTIVE	The sub-state is “RECOVERY_INACTIVE” whenever the LTSSM Major State is not Recovery.
	1 – RECOVERY_RCVR_LOCK	Recovery.RcvrLock
	2 – RECOVERY_RCVR_CFG	Recovery.RcvrCfg
	3 – RECOVERY_IDLE	Recovery.Idle
	4 – RECOVERY_SPEED0	Recovery.Speed – Transmit Electrical Idle Ordered Sets to notify the link partner that we are going idle
	5 – RECOVERY_SPEED1	Recovery.Speed – Determine to which speed to change
	6 – RECOVERY_SPEED2	Recovery.Speed – Wait for remote device to enter electrical idle and remain there for required minimum time
	7 – RECOVERY_SPEED3	Recovery.Speed – Request PHY change speed and wait for PHY to finish changing speed
5 – Disable	0 – DISABLE_INACTIVE	The sub-state is “DISABLE_INACTIVE” whenever the LTSSM Major State is not Disable.
	1 – DISABLE0	Disable – Transmit 16 to 32 TS1 Ordered Sets w/ Disable Link bit asserted
	2 – DISABLE1	Disable – Transition to Electrical Idle
	3 – DISABLE2	Disable – Wait to receive an Electrical Idle Ordered Set and min time of TX_IDLE_MIN afterwards
	4 – DISABLE3	Disable – Wait until a Disable exit condition occurs.

LTSSM Major State	LTSSM Sub-state	DESCRIPTION
6 – Loopback	0 – LOOPBACK_INACTIVE	The sub-state is "LOOPBACK_INACTIVE" whenever the LTSSM Major State is not Loopback.
	1 – LOOPBACK_ENTRY	Loopback.Entry – Loopback entry state – Loopback Master may be required to Tx Loopback TS OS before continuing or speed may need to be changed before beginning loopback.
	2 – LOOPBACK_ENTRY_EXIT	Loopback.Entry – Prepare to enter Loopback.Active
	3 – LOOPBACK_EIOS	Loopback.Entry – Transmit Electrical Idle Ordered Sets to notify the link partner that we are going idle (to change speed)
	4 – LOOPBACK_EIOS_ACK	Loopback.Entry – Wait for the Electrical Idle Ordered Sets transmitted in LOOPBACK_EIOS to exit the core
	5 – LOOPBACK_IDLE	Loopback.Entry – Stay in Electrical Idle for required minimum time.
	6 – LOOPBACK_ACTIVE	Loopback.Active
	7 – LOOPBACK_EXIT0	Loopback.Exit – Tx Electrical Idle
	8 – LOOPBACK_EXIT1	Loopback.Exit – Stay in Electrical Idle for required minimum time.
7 – Hot Reset	0 – HOT_RESET_INACTIVE	The sub-state is "HOT_RESET_INACTIVE" whenever the LTSSM Major State is not Hot Reset.
	1 – HOT_RESET_HOT_RESET	Hot Reset – as Slave
	2 – HOT_RESET_MASTER_UP	Hot Reset – as Master with Link Up
	3 – HOT_RESET_MASTER_DOWN	Hot Reset – as Master with Link Down
8 – TX L0s	0 – TX_L0S_INACTIVE	The sub-state is "TX_L0S_INACTIVE" whenever the LTSSM Major State is not TX L0s.
	1 – TX_L0S_IDLE	Tx_L0s.Idle – Idle
	2 – TX_L0S_TO_L0	Tx_L0s.Idle – Exiting TX L0s; wait for PHY to indicate exit from L0s complete
	3 – TX_L0S_FTS0	Tx_L0s.FTS – Transmit requested NFTS
	4 – TX_L0S_FTS1	Tx_L0s.FTS – Transmit additional FTS required by Cfg Register Extended Sync
9 – L1	0 – L1_INACTIVE	The sub-state is "L1_INACTIVE" whenever the LTSSM Major State is not L1.
	1 – L1_IDLE	L1.Idle
	2 – L1_SUBSTATE	L1.1 or L1.2 depending upon higher level Power Management State Machine control.
	3 – L1_TO_L0	L1.Idle – Exiting L1; wait for PHY to indicate exit from L1 complete
10 – L2	0 – L2_INACTIVE	The sub-state is "L2_INACTIVE" whenever the LTSSM Major State is not L2.
	1 – L2_IDLE	L2.Idle – Idle
	2 – L2_TX_WAKE0	L2.TransmitWake – Transmit a Beacon until remote device exits electrical idle
	3 – L2_TX_WAKE1	L2.TransmitWake – Assert Tx Electrical Idle before changing power state to P1
	4 – L2_EXIT	L2.Idle – L2 exit; wait until PHY finishes power change out of L2
	5 – L2_SPEED	L2.Idle – Change speed if required before going to L2

2.1.4.2. RX L0s State Machine

The Rx_L0s State Machine follows the L0s state of the receiver. The Rx_L0s State Machine operates independently of the main LTSSM which controls the state of the transmitter.

Table 2.2. Single-Ended I/O Standards

LTSSM Sub-state	Description
0 – RX_L0S_L0	The sub-state is "RX_L0S_L0" whenever the receiver is in L0 (ie. not en route to or in Rx L0s).
1 – RX_L0S_ENTRY	Rx_L0s.Entry
2 – RX_L0S_IDLE	Rx_L0s.Idle
3 – RX_L0S_FTS	Rx_L0s.FTS
4 – RX_L0S_REC	Rx_L0s.FTS – Wait until LTSSM Major State == Recovery due to Rx L0s exit error

2.1.5. Multi-Function Support

The Lattice PCIe X1 Core supports 1-4 Functions. Multi-Function Support can only be enabled for Endpoints (functions implementing Type 0 Configuration Space). See the 2.4.1.5.1. function Register 0x8 for the register configuration.

When Multiple Function Support is present, each function is assigned a static Function Number starting at Function Number 0 and incrementing upwards. For ports that communicate function-specific information, port[0] applies to Function[0], port[1] applies to Function[1], etc. If a function is disabled, it does not affect the Function Number of the other enabled functions. Function[0] is always present and cannot be disabled.

2.1.6. Root Port Operation

The Lattice PCIe X1 Core is available in the following application configurations:

- Endpoint
- Root Port

2.1.6.1. Physical Layer Training

When operating as a Downstream Port (Root Port), the Lattice PCIe X1 Core trains as an upstream component (downstream port/lanes). When operating as an Upstream Port (Endpoint), the Lattice PCIe X1 Core trains as a downstream component (upstream port/lanes). Downstream port training and upstream port training are similar, but in some LTSSM states there are important differences in behavior. In a PCI Express link, the device closest to the Root Complex uses the downstream port training protocol and the other device uses the upstream port training protocol.

The direction of traffic which flows from the various Endpoints towards the Root Complex is called upstream traffic. The direction of traffic which flows in the direction from the Root Complex to the various Endpoints is called downstream traffic.

2.1.7. Root Port Configuration

In Root Port mode of operation, Type 1 rather than Type 0 Configuration Registers are implemented in the core.

When the core is operating as a Downstream Port (Root Port), Configuration requests are not permitted to be transmitted upstream and any received PCI Express Configuration Requests is treated as Unsupported Requests.

In Root Port configurations, the I/O, Memory, and Bus windows that are downstream of the Root Port are defined (in accordance with Type 1 Configuration Space) using the I/O Base, I/O Limit, Mem Base, Mem Limit, Prefetchable Mem Base, Prefetchable Mem Limit, Secondary Bus Number, and Subordinate Bus Number Configuration Registers.

When operating as a Downstream Port (Root Port), the core accepts received transactions, which fall outside (and hence target resources which are upstream of the core) these windows and pass them onto the user Local Receive Interface. Transactions falling inside these windows are to devices downstream of the Root Port and considered Unsupprted Requests if received.

A window is disabled (indicating that there are no resources of this type downstream of the port) by programming a Base value that is greater than the Limit value.

2.1.7.1. Root Port PCI Configuration

Configuration of devices downstream of the Root Port should be done in accordance with standard PCI Configuration rules by using the Root Port Transmit Interfaces to transmit Configuration Write and Configuration Read requests and the Root Port Receive Interface to receive the resulting completions.

Configuring a PCI Express hierarchy is a complicated task that is best accomplished with a software routine rather than with dedicated hardware. See the PCI and PCI Express Specifications for details. Leveraging existing PCI configuration software is recommended whenever possible.

2.1.7.2. Root Port vc_rx_cmd_data_o Differences

When operating as a Root Port, the vc_rx_cmd_data_o port has a different meaning. Instead of conveying BAR hit information as in Endpoint Mode, these ports convey Type1 Configuration Space Window hit information:

Individual bits of vc_rx_cmd_data_o[12:0] carry the following meaning in Root Port configurations:

- Bits[12:10] – Traffic Class of the packet (same as for Endpoint)
- Bit[9] – Completion/Request indicator; when set indicates the packet is a Completion packet; when clear the packet is a I/O or Memory Request or Message; the remaining bits in this field are decoded differently for Completion versus Base Address Region hits. (similar as for Endpoint)
- Bits[8:0] : Completion packets (Bit[9] == 1)
  a. Bits[8] – If (1) the completion targets the core's ID; if (0) the completion targets another device behind the core as determined by Secondary Bus Number and Subordinate Bus Number (new bit definition for Root Port mode)
  b. Bits[7:0] – Tag; the Requestor Tag contained in the packet; use to route completions to the associated requestor logic (same as for Endpoint)
- Bits[8:0] : Base Address Region Packet (Bit[9] == 0)
  a. Bit[8] – When (1), the packet is a "write" transaction; when (0), the packet is a "read" transaction (same as for Endpoint)
  b. Bit[7] – When (1), the packet requires one or more Completion transactions as a response; otherwise is (0) (same as for Endpoint)
  c. Bit[6] – (1) if the TLP targets the Expansion ROM Base Address region (same as for Endpoint)
  d. Bit[5] – (1) if the TLP is a Type 1 Configuration Request targeting the Secondary Bus Number
  e. Bit[4] – (1) if the TLP is a Memory Request (or Message routed by address) and the address hits the memory region specified by Prefetchable Mem Base and Prefetchable Mem Limit (BAR4 hit for Endpoint)
  f. Bit[3] – (1) if the TLP is a Memory Request (or Message routed by address) and the address hits the memory region specified by Mem Base and Mem Limit (BAR3 hit for Endpoint)
  g. Bit[2] – (1) if the TLP is an I/O Request and and the address hits the IO region specified by I/O Base and I/O Limit (BAR2 hit for Endpoint)
  h. Bit[1] – (1) if the TLP is a Write or Read Request (or Message routed by address) that targets Base Address Region 1 (same as for Endpoint)
  i. Bit[0] – (1) if the TLP is a Write or Read Request (or Message routed by address) that targets Base Address Region 0 (same as for Endpoint)

2.1.7.3. Locked Transactions

The Lattice PCIe X1 Core does not support locked transactions in Root Port or Endpoint mode. Locked transactions are only required for legacy functions and their use is heavily discouraged by the PCI Express Specification due to the severe performance penalty of implementing locked transactions and the uncertainty as to whether locked transactions are supported by devices in the PCI Express hierarchy.

2.1.7.4. Power Management

The Lattice PCIe X1 Core supports the L1 link state when operating as a Root Port.

The Power Management process for Root Port operation differs from the Endpoint Power Management process described in the Error Handling section wherein the core is acting as the system-side device rather than as the Endpoint.

2.1.7.4.1. Downstream Port L1 Entry and Exit

2.1.7.4.1.1. L1 Entry

Before L1 entry begins, operating system power management software informs the drivers of devices downstream of the core that the link goes to L1. The drivers should complete all outstanding transactions and save any necessary state information before allowing power management software put the link in L1.

Power management software initiates the process of putting the PCI Express link into L1 by writing to the Upstream Port's (Endpoint) Power State Configuration register to a D-state other than D0. A write of a non-D0 Power State value causes the Upstream Port to stop TLP transmissions, wait for all outstanding TLPs to be acknowledged, and to indicate its readiness to Enter L1 by transmitting PM_ENTER_L1 DLLPs.

When the Lattice PCIe X1 Core's Power Management State Machine, operating as a Downstream Port, receives PM_ENTER_L1 DLLPs, the core begins the process of transitioning the link to L1. The core stops TLP transmissions, waits for all outstanding TLPs to be acknowledged, and then transmits PM_REQ_ACK DLLPs in response to Upstream Port's PM_ENTER_L1 DLLPs to inform the Upstream Port that the Downstream Port is also ready for L1 entry. When the Upstream Port receives the core's PM_REQ_ACK DLLPs, the Upstream Port transmits 1 or 2 (depending on Link Speed) Electrical Idle ordered sets and transitions its transmitter to electrical idle. When the Power Management State Machine operating as aDownstream Port, detects the Upstream Port entering electrical idle, the core transmits 1 or 2 (depending on Link Speed) Electrical Idle ordered sets and transitions its transmitter to electrical idle. The link is now in the L1 link state. The PHYs of both devices are put into a low power state to conserve power. Main power, PCI Express reference clock, PHY clock, and core clock are all maintained.

2.1.7.4.1.2. L1 Exit

L1 exit begins when either the core or Upstream Port brings its PHY out of the low power state and exits electrical idle in order to transmit a TLP. The device that wants to transmit a TLP exits electrical idle and transitions to Recovery to retrain the link. The other device senses the remote device's exit from electrical idle, takes its PHY out of the low power state, takes its transmitter out of electrical idle, and also enters Recovery. During Recovery the link is retrained and the PHYs of both devices recover symbol lock. After successful Recovery, both devices transition back to L0 and resume normal operation including transmitting and receiving TLPs. Note that during a normal, successful L1 entry and exit, the core is not reset or powered down and retains all context information. The remote device normally also retains all context information. The Upstream Port's reset behavior in L1 is detected by its advertisement of the No_Soft_Reset Power Management Capability register. If the Upstream Port's No_Soft_Reset register == 1 then the Upstream Port is not reset by a L1 link transition. If No_Soft_Reset == 0, then the link is reset by a L1 link transition and the Upstream Port must be re-enumerated and initialized by the BIOS/OS before being accessed.

2.1.8. Error Handling

The Lattice PCIe X1 Core detects errors and implements the appropriate response, without user intervention, for most error types. In general, you only need to handle error detection and reporting for error types that the core does not have enough information to detect.

Table 2.3, Table 2.4, Table 2.5, and Table 2.6 list the PCI Express defined error types:

- The TYPE column defines the PCI Express defined error severity:
  - COR – Correctable
  - NFAT – Uncorrectable – Non-Fatal
  - FAT – Uncorrectable – Fatal

2.1.8.1. General PCI Express Errors

Table 2.3. General PCI Express Error List

Error	Type
Corrected Internal Error	COR
Uncorrectable Internal Error	FAT
Header Log Overflow	COR

2.1.8.2. Physical Layer Errors

Table 2.4. Physical Layer Error List

Error	Type
Receiver Error	COR

2.1.8.3. Data Link Layer Errors

Table 2.5. Data Link Layer Error List

Error	Type
Bad TLP	COR
Bad DLLP	COR
Replay Timeout	COR
REPLAY_NUM Rollover	COR
Data Link Layer Protocol Error	FAT
Surprise Down	FAT

2.1.8.4. Transaction Layer Errors

Table 2.6. Transaction Layer Error List

Error	Type
Poisoned TLP Received	NFAT
ECRC Check Failed	NFAT
Unsupported Request	NFAT
Completion Timeout	NFAT
Completer Abort	NFAT
Unexpected Completion	NFAT
ACS Violation	NFAT
MC Blocked TLP	NFAT
AtomicOp Egress Blocked	NFAT
Receiver Overflow	FAT
Flow Control Protocol Error	FAT
Malformed TLP	FAT

2.1.9. Power Management

The Lattice PCIe X1 Core supports L0, ASPM L0s,ASPM L1, L1 PM Substates, L1, and L3 link states. L0 (fully operational state) and L3 (off) support is always enabled. The remaining link states may be enabled/disabled via Core Configuration ports. If ASPM L0s, ASPM L1, L1 PM Substates, or L1 support is enabled, then the user design must configure the power management capabilities of the core, and for some link states, take additional action when link states are entered or exited. This section describes the recommended actions user logic should take to control and react to power management states ASPM L0s, ASPM L1, and L1.

The PCI Express Specification defines the following link states:

- L0 – Active
  - Powered
  - Clock & PLLs active; core clock active
  - All PCI Express Transactions and operations are enabled
- ASPM L0s – Low resume latency, energy saving “standby” state
  - Powered
  - Clock & PLLs active; core clock active
  - PHY transmitter in electrical idle
  - Remote PHY receiver must re-establish symbol lock during L0s exit
  - When L0s is enabled by power management software, the core autonomously enters L0s when the transmit side of the link is idle and exit L0s when there is pending information to transmit. Link management DLLPs are required to be transmitted periodically so when a link is otherwise idle, it still enters and exit L0s with regularity to transmit link management DLLPs.
- ASPM L1 – Low resume latency, energy saving “standby” state
  - Powered
  - Clock & PLLs active; core clock active
  - Significant portion of PHY powered down
  - PHY transmitter in electrical idle
  - PHY receiver in electrical idle
  - Deeper power savings but longer resume time than ASPM L0s
  - Remote and local PHY must re-establish symbol lock during L1 exit
  - When ASPM L1 is enabled by power management software, the core autonomously negotiates L1 entry with the link partner after an extended period of link inactivity. The link autonomously returns to L0 when either device in the link has TLPs to transmit.
- L1 – Higher latency, lower power “standby” state
  - Powered
  - Clock & PLLs active; core clock active
  - Significant portion of PHY powered down
  - PHY transmitter in electrical idle
  - PHY receiver in electrical idle
  - Remote and local PHY must re-establish symbol lock during L1 exit
  - The L1 state is entered both under control of power management software
- L3 – Off
  - Main power off; auxiliary power off
  - In this state all power is removed and the core, PHY, and user logic are all non-operational
  - All state information is lost

2.1.9.1. Configuring Core to Support Power Management

The Lattice PCIe X1 Core allows for user logic to implement a wide variety of power management functionality. Primarily, you advertise and control the design’s power management capabilities through the use of core configuration ports.

2.1.9.2. L0s

The Lattice PCIe X1 Core supports Active State Power Management (APSM) L0s.

When L0s support is enabled, ASPM L0s TX Entry Time, the desired amount of time for TLP and DLLP transmissions to be idle before L0s TX is entered, is determined by mgmt_ptl_pm_aspm_l0s_entry_time.

The Number of NFTS sets required by the local PHY to recover symbol lock when exiting L0s is determined by mgmt_tlb_ltssm_nfts_nfts.

NFTS Timeout Extend, mgmt_tlb_ltssm_nfts_to_extend, controls how long the core waits after the expected L0s exit time before directing the link to Recovery to recover from a failed L0s exit. Due to high latencies between a PHY's Rx Electrical Idle output and the associated Rx Data it is normally necessary to choose a relatively high NFTS and NFTS Timeout Extend. See mgmt_tlb_ltssm_nfts_to_extend description for details.

The PCI Express Device Capabilities configuration register has the following L0s fields:

- Bits[8:6] – Endpoint L0s Acceptable Latency – From PCI Express Base Specification, Rev 2.1 section 7.8.3: "Acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance." Note that the amount of buffering does not refer to Lattice PCIe X1 Core buffering, but rather to user application buffering. Users should set this field in accordance with how long a delay is acceptable for their application.
  - 000 – Maximum of 64 ns
  - 001 – Maximum of 128 ns
  - 010 – Maximum of 256 ns
  - 011 – Maximum of 512 ns
  - 100 – Maximum of 1 µs
  - 101 – Maximum of 2 µs
  - 110 – Maximum of 4 µs
  - 111 – No limit
  - Non-Endpoints must hard wire this field to 000.

The PCI Express Link Capabilities configuration register has the following L0s fields:

- Bits[14:12] – L0s Exit Latency – Length of time required to complete transition from L0s to L0:
  - 000 – Less than 64 ns
  - 001 – 64 ns to less than 128 ns
  - 010 – 128 ns to less than 256 ns
  - 011 – 256 ns to less than 512 ns
  - 100 – 512 ns to less than 1 µs
  - 101 – 1 µs to less than 2 µs
  - 110 – 2 µs-4 µs
  - 111 – More than 4 µs
  - Exit latencies may be significantly increased if the PCI Express reference clocks used by the two devices in the link are common or separate.
- Bits[11:10] – Active State Power Management (ASPM) Support should be set to "01" or "11" if L0s support is enabled or "00" otherwise.

2.1.9.3. L1

The Lattice PCIe X1 Core supports both software controller L1 entry (via PowerState Configuration Register) and hardware-autonomous L1 entry (Active State Power Management (APSM) L1).

Software-controlled L1 flow for Upstream Ports (Endpoint) is as follows:

- Software initiates changing a link to L1 by writing the core's Power Management Capability: Power State Configuration Register to a value other than 00 == D0. Note that the component's Device driver participates in this process and must ensure that all traffic is idle before permitting the system to power down to L1.
- When the core detects a change of Power State to a non-D0 value, the core's power management state machine, which is responsible for the higher-level power management protocol, follows the following sequence:
  - Block further TLP transmissions
  - Wait for all in process TLPs to complete transmission
  - Wait for the Replay Buffer to empty (all transmitted TLPs acknowledged)
  - Core transmits PM_ENTER_L1 DLLPs until receiving a PM_REQ_ACK DLLP from remote device
  - Core directs LTSSM state machine to L1
  - When a TLP is pending or the LTTSM state machine indicates L1 state has been exited due to link partner activity, the core returns to L0

Software-controlled L1 flow for Downstream Ports (Root Port) is similar to above, but the Downstream Port core is following the Upstream Port's L1 request:

- Core receives PM_ENTER_L1 DLLP from Upstream Port
- Block further TLP transmissions
- Wait for all in process TLPs to complete transmission
- Wait for the Replay Buffer to empty (all transmitted TLPs acknowledged)
- Transmit PM_REQ_ACK DLLP to remote device until remote device idles its transmitter
- Core directs LTSSM state machine to L1
- When a TLP is pending or the LTTSM state machine indicates L1 state has been exited due to link partner activity, the core returns to L0

The PCI Express Device Capabilities configuration register has the following L1 fields:

- Bits[11:9] – Endpoint L1 Acceptable Latency – From PCI Express Base Specification, Rev 2.1 section 7.8.3: "This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance." Note that the amount of buffering does not refer to Lattice PCIe X1 Core buffering, but rather to user application buffering. Users should set this field in accordance with how long a delay is acceptable for their application.
  - 000 – Maximum of 1 µs
  - 001 – Maximum of 2 µs
  - 010 – Maximum of 4 µs
  - 011 – Maximum of 8 µs
  - 100 – Maximum of 16 µs
  - 101 – Maximum of 32 µs
  - 110 – Maximum of 64 µs
  - 111 – No limit
  - Non-Endpoints must hard wire this field to 000.

PCI Express Link Capabilities configuration register has the following L1 fields:

- Bits[17:15] – L1 Exit Latency – Length of time required to complete transition from L1 to L0:
  - 000 – Less than 1µs
  - 001 – 1 µs to less than 2 µs
  - 010 – 2 µs to less than 4 µs
  - 011 – 4 µs to less than 8 µs
  - 100 – 8 µs to less than 16 µs
  - 101 – 16 µs to less than 32 µs
  - 110 – 32 µs-64 µs
  - 111 – More than 64 µs
  - Exit latencies may be significantly increased if the PCI Express reference clocks used by the two devices in the link are common or separate.
- Bits[11:10] – Active State Power Management (ASPM) Support should be set to "10" or "11" if L1 support is enabled or "00" otherwise.

Hardware-autonomous L1 (ASPM L1) entry is initiated only by Upstream Ports (Endpoint). The core ASPM L1 functionality must be enabled and advertised in PCIe Link Capabilities and software must enable ASPM L1 support in order for the hardware-autonomous L1 to be able to be negotiated. When ASPM L1 support is present and enabled for an Upstream Port, the core requests that the link be directed to L1, using ASPM L1 protocol, whenever the link is idle (no TLP or ACK/NAK DLLPs are transmitted) for the configured idle period.

A Downstream Port (Root Port) receives ASPM L1 requests from the remote device and may choose to accept or rejct the request. The core accepts ASPM L1 requests when there are no TLP or ACK/NAK DLLP pending transmit and otherwise rejects the request.

2.1.10. PCIe X1 Core Buffers

The Lattice PCIe X1 Core contains three large RAM buffers:

- Transmit Buffer for transmitting TLPs
- Receive Buffer for receiving TLPs
- Replay Buffer for holding TLPs that were transmitted until positive acknowledgement of receipt is received

The size of the Transmit Buffer, Receive Buffer, and Replay Buffer and the size of the corresponding buffers in the remote PCI Express Device have a fundamental impact on the throughput performance of the PCI Express link.

To achieve the highest throughputs, the buffers for both devices in the PCI Express link must be large enough that they can still accept more data while the oldest data begins to be freed from the buffer. If a buffer is too small, then the link stalls until the buffer has enough space to continue. The buffers must be large enough to overcome the expected latencies or throughput is affected.

2.1.10.1. PCI Express Credits

Flow Control DLLPs communicate available buffer space in units of Header and Data Credits as defined in the PCI Express Specification. The amount of space required by a Header is 12-20 bytes or (3-5 DWORDs with 1 DWORD == 4 bytes). Each Header Credit represents the capability to store a maximum size packet header which includes all of the transaction control information (address, byte enables, requester ID, etc.) and an optional End to End CRC (ECRC). Each Data Credit represents 16 bytes (4 DWORDs) of data payload. A transaction cannot be transmitted unless there is at least 1 header credit and enough data credits for the packet payload available in the remote device's Receive Buffer.

Credits are further divided into three categories for each of the main types of traffic. There are Posted (memory write requests and messages), Non-Posted (all reads; Configuration and I/O writes), and Completion (responses to Non-Posted Requests) credit categories. Each type of traffic must obey PCI Express transaction ordering rules and is stored in its own buffer area.

The credit categories are annotated as:

- PH – Posted Request Header Credits
- PD – Posted Request Data Payload Credits
- NH – Non-Posted Request Header Credits
- ND – Non-Posted Request Data Payload Credits

- CH – Completion Header Credits
- CD – Completion Data Payload Credits

PCI Express is inherently high-latency due to the serial nature of the protocol (clock rate matching, lane-lane de-skewing, etc.) and due to the latency induced by requiring packets to be fully received and robustly checked for errors before forwarding them for higher-level processing.

To achieve the best throughputs, both the Lattice PCIe X1 Core and the remote PCI Express device must be designed with a suitable number of credits and the capability to overlap transactions to bury the transaction latency.

The Lattice PCIe X1 Core Transmit, Receive, and Replay buffers are delivered with sufficient size to overcome the latencies of typical open system components.

2.1.10.2. Max Payload Size

The maximum payload size of any given packet is limited by the Max Payload Size field of the Device Control Configuration Register. The PCI Express Specification defines 128, 256, 512, 1024, 2048, and 4096-byte payload sizes. The maximum payload size that a device can support is limited by the size of its posted and completion TLP buffers. The Transmit Buffer and Receive Buffer Posted and Completion TLP storage and the Replay Buffer TLP storage needs to be able to hold at least 4 Max Payload Size TLPs in order to be reasonably efficient.

Each device advertises the maximum payload size that it can support and the OS/BIOS configures the devices in a link to use the lowest common maximum payload size. Thus, it is not advantageous to support a greater maximum payload size than the devices with which one is communicating.

The higher the TLP payload size, the lower the TLP header and framing overhead is compared to the data. Above 512-byte Max Payload Size the incremental throughput benefit of higher payload sizes is small and the design area and latency for using these larger payloads is expensive. Thus it is generally recommended to design for <= 512 Max Payload Size.

The Lattice PCIe X1 Core supports up to 512 Bytes Max Payload Size and the internal buffers can hold about 3x of the max payload size. However, given that typical PCIe devices currently available to communicate with support 256-byte maximum payloads, supporting greater than this amount is not likely to result in better performance and just consumes more memory/logic resources.

2.1.11. User Interface Options

The Lattice PCIe X1 Core provides an option to convert the Hard IP native interface into some of the standard interfaces such as AHB-Lite, AXI4-stream, and APB.

The different user interface options are illustrated in the following figures:

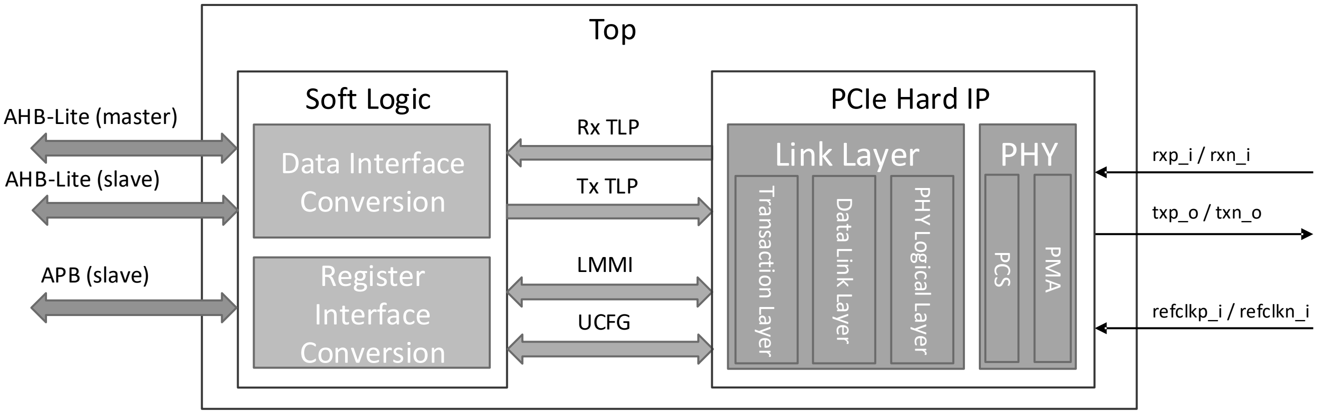


Figure 2.4. AHB-Lite Data Interface, APB Register Interface

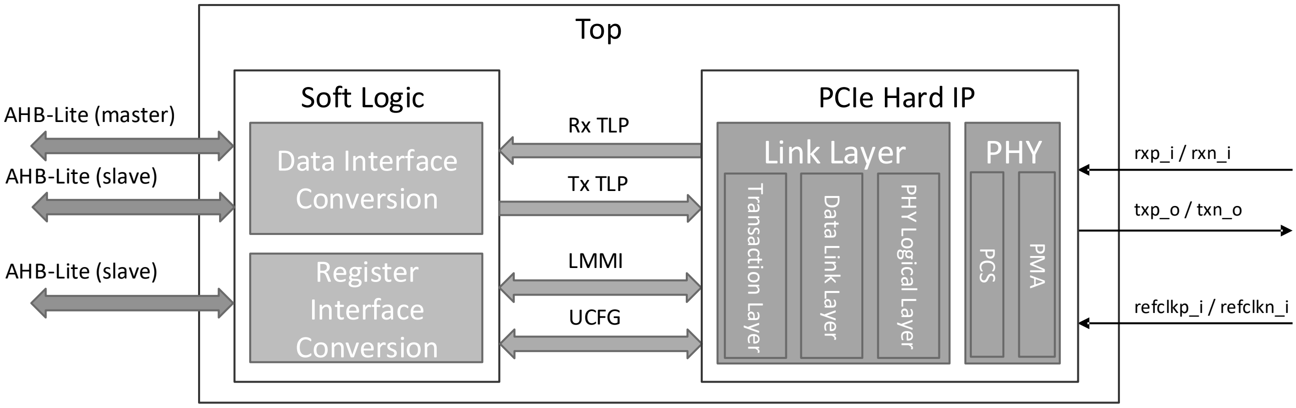


Figure 2.5. AHB-Lite Data Interface, AHB-Lite Register Interface

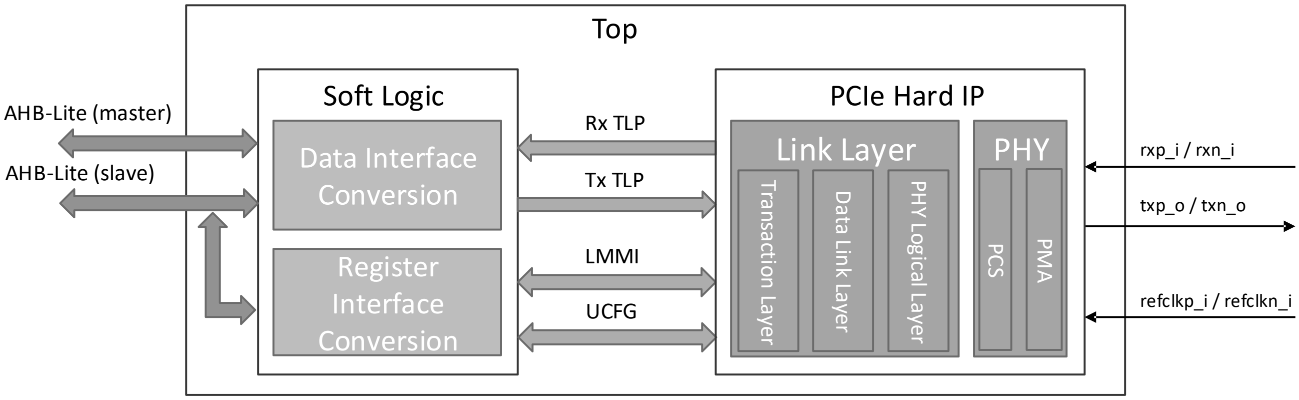


Figure 2.6. AHB-Lite Data Interface shared with Register Interface

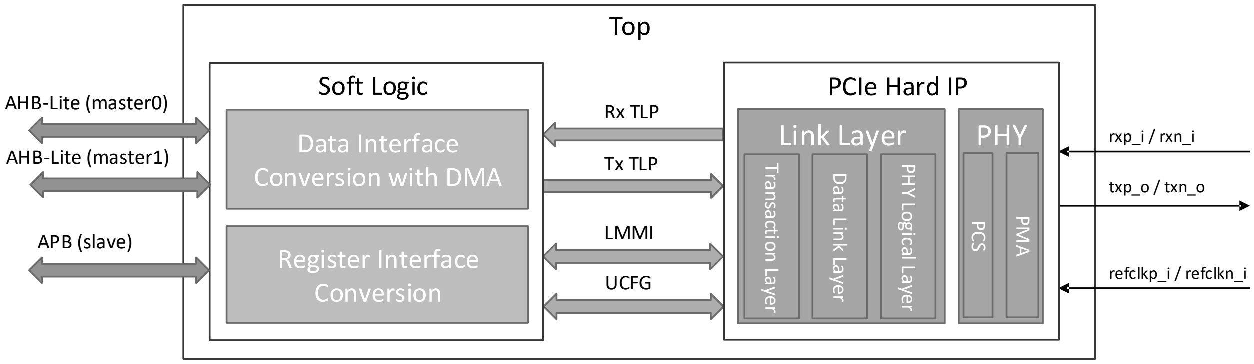


Figure 2.7. AHB-Lite Data Interface with DMA, APB Register Interface

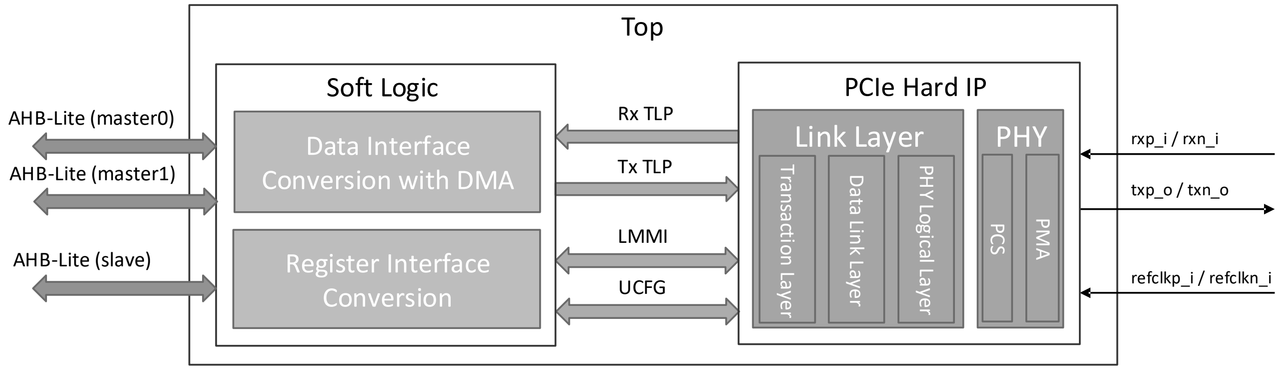


Figure 2.8. AHB-Lite Data Interface with DMA, AHB-Lite Register Interface

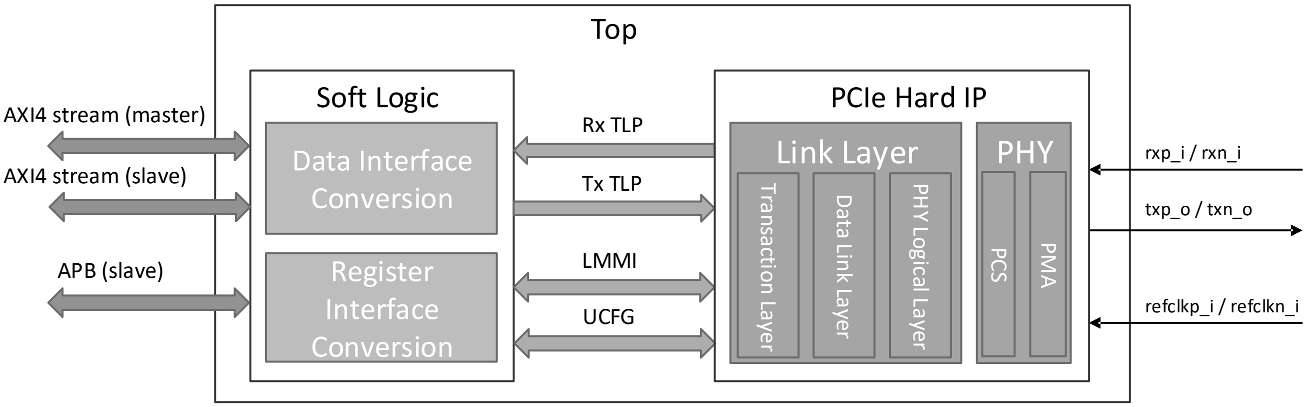


Figure 2.9. AXI4-Stream Data Interface, APB Register Interface

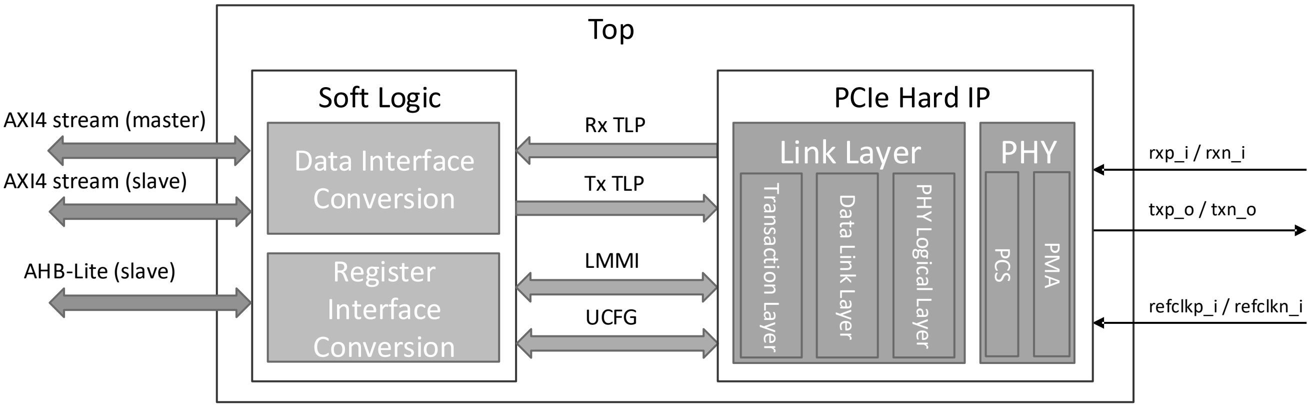


Figure 2.10. AXI4-Stream Data Interface, AHB-Lite Register Interface

2.1.12. DMA Support

Direct Memory Access (DMA) support is an option provided by soft IP to enable a more efficient data transfer when the device acts as initiator or master. This feature can be enabled if the data interface selected is “AHB_LITE”.

The data transfer with DMA support is illustrated in the following figures. Additional registers required by DMA are implemented as well as status registers and interrupt signals which are discussed in the subsections below.

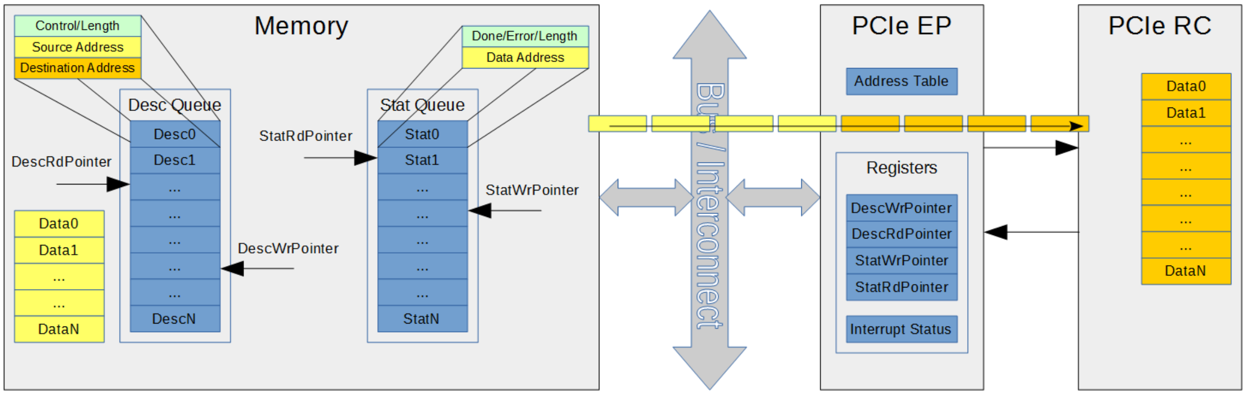


Figure 2.11. AHB-Lite to PCIe Data Transfer

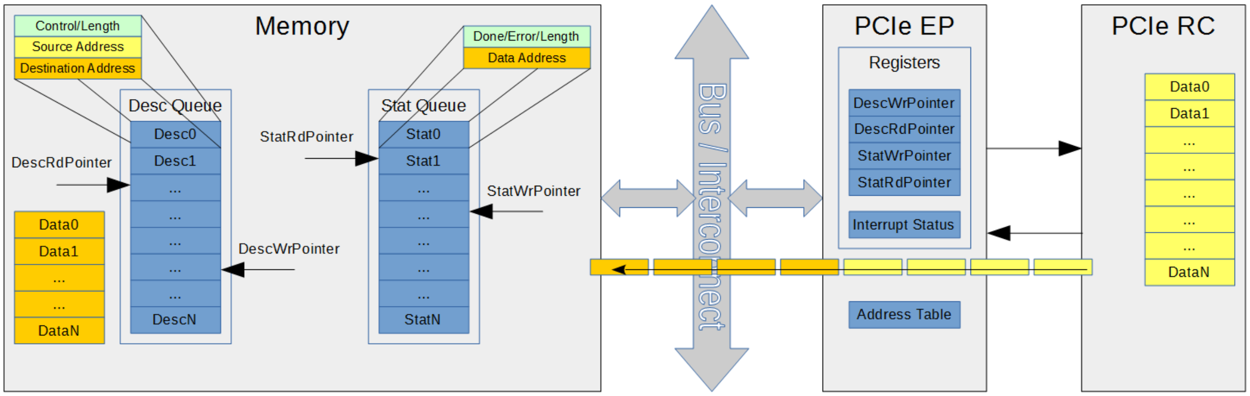


Figure 2.12. PCIe to AHB-Lite Data Transfer

To transfer data from AHB-Lite to PCIe or PCIe to AHB-Lite, the Core requires the source and destination address and transfer controls (e.g. length and direction of transfer). This information or details of transfer request are grouped into a structure called descriptor.

The descriptors are queued and stored by the DMA master (that is, Driver or Application logic) in a local memory and informs the Core by setting the value of DescWrPointer register. It is expected that the DMA master is aware of the maximum queue size and the rollover value of the pointers.

When the difference between the values of DescWrPointer and DescRdPointer is greater than zero it means that the descriptor queue is not empty and it means that the DMA master is requesting a data transfer. The Core fetches the descriptor through the AHB-Lite Master interface and increments the value of DescRdPointer register. An interrupt signal and status (if enabled) is asserted to inform the DMA master that the DescRdPointer is updated. The DMA master can post more entries in the descriptor queue as long as it is less than the full condition ((DescWrPointer+1) != DescRdPointer).

After fetching the descriptors, the Core reads the data from the specified source address and transfer it to the specified destination address. If the direction of transfer is from AHB-Lite to PCIe, the Core reads the data from the memory through AHB-Lite master and one or more MemWr TLPs are generated and sent to PCIe until the transfer is completed. If the direction of transfer is from PCIe to AHB-Lite, the Core sends a MemRd request and post an entry to the internal read request queue and wait for completion data. Incoming completions are matched with the read request entries and then transferred to the specified destination through AHB-Lite Master interface.

Once the transfer is completed or aborted, a status which contains done flag, error flag, length of transfer and data (buffer) address offset is written to the status queue. The Core increments the value of StatWrPointer register and assert the interrupt signal (if enabled) to inform the DMA master. The DMA master processes each status entry and update the StatRdPointer. The Core does not fetch and process new descriptors when the status queue is full. To prevent stalling of descriptor processing, the DMA master should monitor the status queue and update the StatRdPointer regularly.

2.1.12.1. Descriptor and Status Entry Format

Table 2.7. Descriptor Entry Format

DW	DW Name	Field Name	Bit Offset	Size	Description
0	desc_ctrl	length	0	13	Size of data transfer in bytes. (4096 bytes maximum)
		direction	13	1	Direction of transfer. 0 – AHB-Lite to PCIe 1 – PCIe to AHB-Lite
			14	10	Reserved
		desc_id	24	8	Optional descriptor ID. If the parameter EN_DESC_ID == "Enable" the Core adds this information in the Status entry.
1	desc_src	addr_offset	0	32	Source Address/Offset.
2	desc_dst	addr_offset	0	32	Destination Address/Offset.
3	desc_hdr	requester_id	0	16	Requester ID to be used in TLP Header requester_id [7:0] – bus number[7:0] requester_id [10:8] – function number[2:0] requester_id [15:11] – device number[4:0]
		traffic_class	16	3	Traffic Class to be used in TLP Header
		use_requester_id	19	1	When set, indicates that the requester_id field is valid and should be used in TLP header. Otherwise, the Core uses the captured configuration ID of function 0 as the default requester ID.
			20	12	Reserved
1	stat_flag	done	0	1	If this bit is asserted, it indicates that the transfer has been completed
		with_error	1	1	If this bit is asserted, it Indicates an error occurred during transfer.
		aborted	2	1	If this bit is asserted, it indicates the transfer was terminated before it completes the specified length.
		direction	3	1	Direction of transfer. 0 – AHB-Lite to PCIe 1 – PCIe to AHB-Lite
			4	4	Reserved
		desc_id	8	8	Optional descriptor ID. Available if the parameter EN_DESC_ID == "Enable"
		length	16	13	Size of data transfer in bytes. (4096 bytes maximum)
			29	3	Reserved
1	stat_buf	addr	0	32	Data Address. This is the local memory addess where the data is stored (direction==1) or fetched (direction==0).

2.1.12.2. Data Transfer Example

Below is a sample pseudo-code that shows how the DMA support operates.

Transfer 64 KB data from remote PCIe device to local memory.

```
descriptor_queue_size = 8
status_queue_size = 16
Initialization:
DMA master allocates memory space for descriptor queue (descriptor_queue_size *
sizeof(descriptor_struct)).
   descriptor_base_address = (unsigned int *) malloc (8*16)
DMA master allocates memory space for status queue (status_queue_size *
sizeof(status_struct)).
   status_base_address = (unsigned int *) malloc (16*8)
DMA master programs the descriptor base address register (dma_support_reg0)
   apb_write ({PCIE_CSR_BASE, 0x2800C}, descriptor_base_address)
DMA master programs the status base address register (dma_support_reg1)
   apb_write ({PCIE_CSR_BASE, 0x28010}, status_base_address)
DMA master programs the descriptor queue size and status queue size register
(dma_support_reg2)
   apb_write ({PCIE_CSR_BASE, 0x28014}, {7'd0,9'd16,7'd0,9'd8})
Main Program:
DMA master creates descriptor entries. The maximum transfer size of a single
descriptor is 4KB, so we need 16 descriptor entries to transfer the whole 64 KB.
In this example, assume that the data in remote PCIe device is located at
SRC_BUF_BADDR and will be transferred to the local memory at address
DST_PCIE_BADDR.
SRC_BUF_BADDR - source base address (in remote PCIe device) of the data to be
transferred
DST_PCIE_BADDR - destination base address of the local memory
int main () {
  descriptors_created = 0
  desc_wr_pointer = 0
  while (descriptors_created < 16) {
    available_desc_slots = 7 - check_desc_queue_size          // Full
condition is max queue size - 1 = 7
    if (available_desc_slots > 0) {
      for (idx=0; idx < available_desc_slots; idx=idx+1) {
  descriptor_entry = {4{32'd0}}
  descriptor_entry.length = 4096       // 4 KB max transfer per descriptor
  descriptor_entry.direction = 1       // PCIe -to-AHB-Lite
  descriptor_entry.source_addr = SRC_BUF_BADDR + (descriptors_created *4096)
  descriptor_entry.destination_addr = DST_PCIE_BADDR + (descriptors_created
*4096)
  // write descriptor entry to descriptor queue
  ahbl_write ((descriptor_base_address+ desc_wr_pointer*16+0),
descriptor_entry[1*32 : 0*32])
  ahbl_write ((descriptor_base_address+ desc_wr_pointer*16+4),
descriptor_entry[2*32 : 1*32])
  ahbl_write ((descriptor_base_address+ desc_wr_pointer*16+8),
descriptor_entry[3*32 : 2*32])
  ahbl_write ((descriptor_base_address+ desc_wr_pointer*16+12),
descriptor_entry[4*32 : 3*32])
  descriptors_created = descriptors_created + 1
  // Increment descriptor write pointer
```

```
    if (desc_wr_pointer == 7) desc_wr_pointer = 0     // roll over to 0 when
pointer == max queue size-1
    else desc_wr_pointer = desc_wr_pointer + 1
    if(descriptors_created == 16) break
            } // for
            // Update descriptor write pointer register
            apb_write({PCIE_CSR_BASE, 0x28018}, desc_wr_pointer)
        } // if
    } // while
} // main
int check_desc_queue_size() {
// Read the descriptor pointer register
apb_read({PCIE_CSR_BASE, 0x28018}, read_data)
desc_write_pointer = read_data[7:0]
desc_read_pointer = read_data[23:16]
if(desc_write_pointer > desc_read_pointer) {
    return (desc_write_pointer – desc_read_pointer)
} else return (desc_read_pointer  – desc_write_pointer)
}
```

When the descriptor write pointer is updated, the Core would detect that the descriptor queue is not empty and therefore fetch the descriptor entries and perform the DMA request. The direction of transfer would determine what kind of TLP is generated (MWr or MRd). In this case, the direction is from PCIe to AHB-Lite so memory read request (MRd) TLPs are sent to PCIe.The number of read request TLPs to be sent would depend on the maximum read request size specified in the PCIe Device Control register. If for instance the maximum read request size is set to 512 bytes, then the Core sends (4096/512) 8 memory read request (MRd) TLPs for each descriptor entry.

The received data from completion TLPs is written to the specified destination address. A status entry is posted in the status queue once all the requested data of a descriptor entry have been received and transferred to the local memory.

2.2. Signal Description

2.2.1. Clock and Reset

2.2.1.1. Clock and Reset Port Descriptions

Table 2.8. Clock and Reset Port Descriptions

Port	Type	Description
perst_n_i	Input	PCI Express Fundamental Reset. Active-low asynchronous assert, synchronous de-assert (synchronous to clk_usr_i). Resets the Link Layer, PHY, and Soft Logic blocks. On perst_n_i and rst_usr_n_i de-assertion the core starts in the Detect.Quiet Link Training and Status State Machine (LTSSM) state with the Physical Layer down and Data Link Layer down. perst_n_i must remain asserted while the PHY registers are being configured.
rst_usr_n_i	Input	User Clock Domain Link Layer Reset (Link Layer Reset). Active-low asynchronous assert, synchronous de-assert reset to the User clock domain, Link Layer and Soft Logic blocks. On perst_n_i and rst_usr_n_i de-assertion the core starts in the Detect.Quiet Link Training and Status State Machine (LTSSM) state with the Physical Layer down and Data Link Layer down. It is recommended that rst_usr_n_i remain asserted while the Link Layer core registers are being configured.
clk_usr_i	Input	User Clock Domain Input Clock. It is recommended to use the following minimum clock frequency in order to achieve the maximum throughput with respect to link data rate: 5.0G – 125 MHz 2.5G – 62.5 MHz Note: The uclk_period_in_ps register (0xF00C) should be updated with the actual value of the clock period used in clk_usr_i. Users must ensure that clk_usr_i has an active (toggling) clock input when de-asserting the reset ports (perst_n_i / rst_usr_n_i), otherwise the core may get stuck at reset.
clk_usr_o	Output	User Clock Domain Output Clock. This is the pclk output that comes from the PHY. By default, the clk_usr_o uses the divide-by-2 version (125 MHz) of the pclk from PHY. You have the option to use this clock as input to clk_usr_i. Note that clk_usr_o is inactive (stays low) when PHY is on reset (perst_n_i is asserted or the register pipe_rst (0x0F004) is asserted).

Port	Type	Description
flr_o [NUM_FUNCTIONS-1:0]	Output	Per function Function Level Reset (FLR) indicator. flr_o [i] == 1 indcates FLR is active for function[i]. flr_o [i] == 0 indicates FLR is not active for function[i]. FLR is a function-specific soft reset that occurs when software writes the FLR register in a function's configuration space to 1. When FLR is acive, the function's Configuration Space registers are reset to their default values (except Sticky registers as specified by PCIe Spec.). A function's FLR Configuration Space register remains set until flr_ack_i[i] for the associated function[i] is set to 1 for one clock to indicate that you have completed resetting the application logic associated with that function.
flr_ack_i [NUM_FUNCTIONS-1:0]	Input	Per function Function Level Reset (FLR) acknowledge. Set flr_ack_i [i] == 1 for one clock to indicate that you have completed processing an active flr_o[i] for function[i] and is ready to exit FLR for the function. FLR is only enabled for Endpoints. FLR support may be disabled via mgmt_ftl_pcie_dev_cap_disable_flr_capability register except per PCIe Spec..
u_pl_link_up_o	Output	Physical Layer Link Up Status. (1) Up. (0) Down. u_pl_link_up_o is used as an active-low, synchronous reset for the core's Data Link Layer. Users are not expected to use this port except for status since their RTL does not interface directly with the Data Link Layer.
u_dl_link_up_o	Output	Data Link Layer Link Up Status. (1) Up. (0) Down. u_dl_link_up_o is used as an active-low, synchronous reset for the Transaction Layer and also indicates when TLPs can be successfully transmitted across the link. For Endpoint-only applications, users must use u_dl_link_up_o as a synchronous reset for their RTL interfacing to the core's Transaction Layer interfaces.
u_tl_link_up_o	Input	Transaction Layer Link Up Status. (1) Up. (0) Down. u_tl_link_up_o is an active-low, synchronous reset to the core's upper transaction layer. A Downstream Port's (Root Port) Configuration Registers, and the RTL which enables them to be accessed them from the User Transmit and Receive Interfaces, are reset by u_tl_link_up_o so that the PCIe Configuration Registers may be read even when the link is down. Downstream Port applications must use u_tl_link_up_o as a synchronous reset for their RTL interfacing to the core's Transaction Layer interfaces. When u_tl_link_up_o == 1 the core's PCIe Configuration Registers may be read. When u_dl_link_up_o == 1, accesses the PCIe devices behind the DS Port may be attempted. When u_dl_link_up_o == 0 accesses to the PCIe devices behind the DS Port is not possible; if accesses are attempted, they fail with error status.

Port	Type	Description
u_ltssm_disable_i	Input	The LTSSM does not transition from Detect.Quiet to Detect.Active to begin LTSSM training while u_ltssm_disable_i == 1. u_ltssm_disable_i may be thus be used to delay the start of LTSSM training which otherwise begins as soon as perst_n_i and rst_usr_n_i are deasserted. u_ltssm_disable_i must be set to 1 relatively soon (within a few mS) after perst_n_i and rst_usr_n_i are released as the system allocates a finite amount of time for devices to initialize before it begins to scan for devices. If u_ltssm_disable_i is held for too long, software may scan for the device before it becomes operational and assume that no device is present.

***Note:** NUM_FUNCTIONS – range (1,4)

2.2.1.2. Clock and Reset Relationship

The Lattice PCIe X1 Core uses the PHY's PIPE PCLK output as internal clock for the PIPE interface. Before the Link Layer reset (rst_usr_n_i) can be released, the PHY's PIPE PCLK output must be valid, the PHY must be ready for operation, and the Link Layer has been configured. The PHY and Link Layer clock and reset operation is illustrated in Figure 2.13.

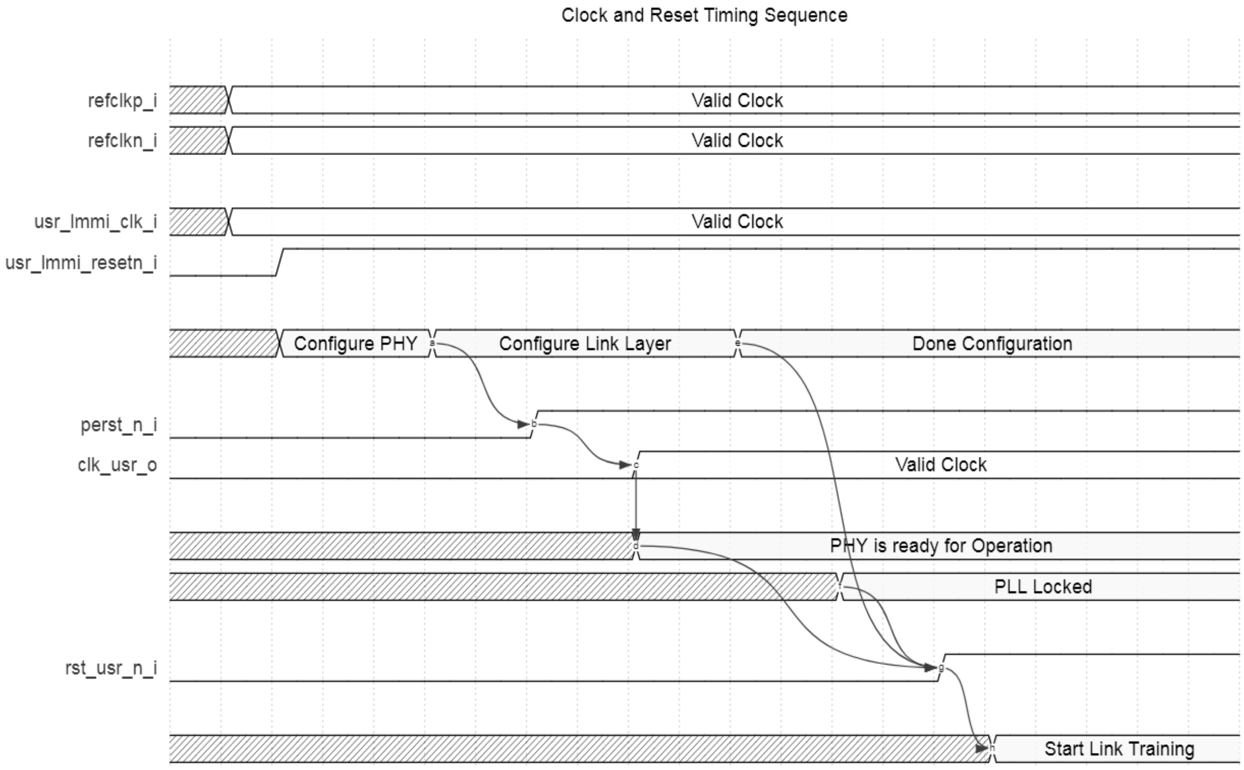


Figure 2.13. Clock and Reset Sequence Diagram

The Lattice PCIe X1 Core configuration register implementation has default values that are appropriate for most applications. Customers typically would only want to change a small number of values such as Vendor/Device ID and BAR configuration. Such changes can be made in the IP generation user interface or by register programming through the Lattice Memory Mapped Interface (LMMI).

To configure the PHY registers through LMMI, then it should be done before the de-assertion of perst_n_i. The PHY would then be released from reset and it is ready for operation once it is able to generate the PIPE clock output (clk_usr_o). The user domain reset (rst_usr_n_i) can be de-asserted if the Link Layer register configuration is done or skipped.

To ensure that the clock is stable before the link training, you may wait for the PLL locked status before de-asserting the user domain reset (rst_usr_n_i). The PLL status (0xF200) can be read through the LMMI.

2.2.2. PHY Interface

The Link Layer is used in conjunction with a third party PCI Express PHY to implement a complete Lattice PCIe X1 Core PCI Express implementaion. The PHY implements the high-speed serial and analog functions required to support PCI Express while the Link Layer implements the majority of the digital logic as well as the higher levels of the PCI Express protocol.

The PIPE PHY Interface that connects the Link Layer and PHY is not shown here since the interface is only internal and is not visible to you.

2.2.2.1. PHY Interface Port Descriptions

Table 2.9. PHY Interface

Port	Type	Description
rxp_i	Input	Differential Receive Serial signal, Rx+
rxn_i	Input	Differential Receive Serial signal, Rx-
txp_o	Output	Differential Transmit Serial signal, Tx+
txn_o	Output	Differential Transmit Serial signal, Tx-
refclkp_i	Input	Differential Reference Clock, CLK+ (100 MHz)
refclkn_i	Input	Differential Reference Clock, CLK- (100 MHz)
aux_clk_i	Input	Low speed Auxilliary Clock (16 MHz minimum). This clock is required when L1 Substate is enabled. During low power mode when the Core enters L1 substate (L1.1 or L1.2), the PHY turns off most of the power consuming blocks including PLLs thus turning off the Link Layer clock. The aux_clk_i serves as an always on clock that is used by the Link Layer to wake up and exit from L1 substate. **Note:** The aux_clk_period_in_ps register (0xF010) should be updated with the actual value of the clock period used in aux_clk_i.
clkreq_n_io	InOut	CLKREQ# bidirectional open-drain pin. The CLKREQ# signal is an open drain, active low signal that is driven low by the add-in card to request that the PCI Express reference clock be available (active clock state) in order to allow the PCI Express interface to send/receive data. Operation of the CLKREQ# signal is determined by the state of the Enable Clock Power Management bit in the Link Control Register (offset 010h). When disabled, the CLKREQ# signal shall be asserted (clkreq_n_io = 1'b0) at all times whenever power is applied to the card, with the exception that it may be de-asserted during L1 PM Substates. When enabled, the CLKREQ# signal may be de-asserted (clkreq_n_io = 1'b1) during an L1 Link state. The CLKREQ# signal is also used by the L1 PM Substates mechanism. In this case, CLKREQ# can be asserted by either the system or add-in card to initiate an L1 exit. See the PCI Express Base Specification for details on the functional requirements for the CLKREQ# signal when implementing L1 PM Substates. Whenever dynamic clock management is enabled and when a card stops driving CLKREQ# low, it indicates that the device is ready for the reference clock to transition from the active clock state to a parked (not available) clock state. Reference clocks are not guaranteed to be parked by the host system when CLKREQ# gets de-asserted and module designs shall be tolerant of an active reference clock even when CLKREQ# is de-asserted by the module. **Note:** This signal must be tied to low if CLKREQ# is not used.

2.2.3. Transaction Layer Interface

The Lattice PCIe X1 Core implements a complete PCI Express implementation incuding Physical, Data Link, and Transaction Layer functionality.

You transmit PCI Express Transaction Layer Packets (TLPs) on the PCI Express link via the Transmit Interface.

You receive PCI Express TLPs from the PCI Express link through the Receive Interface.

2.2.3.1. Transmit Interface Port Descriptions

The Transmit Interface is the mechanism with which you transmit PCI Express TLPs over the PCI Express bus. You formulate TLPs for transmission in the same format as defined in the PCI Express Specification; you are responsible for supplying a complete TLP comprised of packet header, data payload, and optionally a TLP Digest. The core Data Link Layer adds the necessary framing (STP/END/EDB), sequence number, Link CRC (LCRC), and optionally computes and appends the ECRC (TLP Digest) when ECRC is not already present in the TLP.

TLPs are transmitted to master write and read requests, to respond with completions to received read and I/O requests, to transmit messages, etc.

The Lattice PCIe X1 Core automatically implements any necessary replays due to transmission errors, etc. If the remote device does not have sufficient space in its Receive Buffer for transmit TLPs, the Lattice PCIe X1 Core pauses TLP transmission until space becomes available.

For Endpoint applications, PCI Express TLPs are transmitted exactly as received by the core on the Transmit Interface with no validation that the packets are formulated correctly by the user. It is critical that all TLPs transmitted are formed correctly and that vc_tx_eop_i is asserted at the appropriate last vc_tx_data_i word in each packet.

For Root Port applications, transmitted TLPs are validated against the core’s Type 1 Configuration Registers for validity. Valid TLPs are forwarded to PCIe while invalid TLPs are handled as errors – returning error completions as appropriate.

The Transmit Interface includes the option to nullify TLPs (instruct the Receiver to discard the TLP) to support the cancellation ol TLP transmisions when errors are detected after the TLP transmission has started. Nullified TLPs that target internal core resources (Root Port Configuration Registers and Power Management Messages) are discarded without affecting the internal core resources. Nullified TLPs that do not hit internal resources are discarded.

The Transmit Interface ports are listed in Table 2.10.

See the timing diagrams in the Transmit Interface Example Transactions section for additional details.

Table 2.10. Transmit Interface

Port	Type	Description
vc_tx_valid_i	Input	Source valid. 1==Valid, 0==Not valid.
vc_tx_ready_o	Output	Destination ready. 1==Ready, 0==Not ready. A transfer occurs on the Transmit Interface when vc_tx_valid_i==vc_tx_ready_o==1.
vc_tx_sop_i	Input	Start of packet indicator Set == 1 coincident with the first vc_tx_data_i word in each TLP.
vc_tx_eop_i	Input	End of packet indicator Set == 1 coincident with the last vc_tx_data_i word in each TLP.
vc_tx_eop_n_i	Input	Nullify packet indicator Set == 1 coincident with vc_tx_eop_i == 1 to instruct the core to nullify the current TLP (invert LCRC and use EDB framing) instead of transmitting the TLP normally.
vc_tx_data_i[31:0]	Input	TLP data to transfer vc_tx_data_i must be valid from the assertion of vc_tx_sop_i until the TLP is fully consumed with the assertion of vc_tx_eop_i. TLP data must comprise a complete Transaction Layer Packet (TLP) as defined by the PCI Express Specification including the entire 3 or 4 DWORD TLP header, data payload (if present), and optionally a TLP Digest (ECRC). The core adds the necessary STP/END/EDB framing, Sequence Number, LCRC, and add ECRC (if enabled to do so and ECRC is not already present in the transmission) as part of its Data Link Layer functionality. Transmitted TLPs are required to be formulated correctly per the PCIe Specification including filling in the Requester/Completer ID, Attributes, Traffic Class, etc. For Mutli-Function, the core uses the Requestor/Completer ID in transmitted TLPs to determine which function's Configuration Regsiters should be applied to determine the validity of the transmitted TLP.
vc_tx_data_p_i[3:0]	Input	Parity of associated vc_tx_data_i: vc_tx_data_p_i [i] == ^(vc_tx_data_i[((i+1)*8)-1:(i*8)]) vc_tx_data_p_i must be valid for all bytes in vc_tx_data_i that contain a portion of a TLP (header, payload, and TLP digest (if present));

2.2.3.2. Transmit Credit Interface Port Descriptions

The Transmit Credit Interface provides the means for flow control of non-posted transmit transactions between the core Transmit Buffer and user. This is important for allowing Posted and Completion TLPs to continue to make progress when non-posted TLPs are blocked (which can be necessary in some cases to avoid potential deadlock conditions). The amount of non-posted TLP storage in the core Transmit Buffer is communicated on the Transmit Credit Interface. You are expected to use this interface to limit simultaneously outstanding TLP transmission of non-posted TLPs to the amount of non-posted TLPs that the core is able to aborsb into the non-posted Transmit Buffer.

Note that core/link partner transmit TLP flow control is not managed via this interface; the core manages transmit flow control between the core and the PCIe link partner Receive Buffer without user intervention.

Table 2.11. Transmit Credit Interface

Port	Type	Description
vc_tx_credit init_o	Output	When the core Transaction Layer for Link[i] is ready to accept TLP transmissions, the core asserts vc_tx_credit init_o == 1 for one clock cycle and on the same cycle indicates the non-posted TLP Header strorage capacity of the Transmit Buffer on vc_tx_credit_nh_o[11:0]. You are expected to keep and initialize their NH available transmit credit counters on vc_tx_credit init_o==1. When a non-posted TLP is pending for transmission within user logic, user logic should check the currently available NH credit count for the associated link and hold the transmission until enough NH credits are available to transmit the TLP. Once the TLP has been committed for transmit, the amount of NH credits required by that TLP are decremented from the NH credit count. As the core forwards transmitted TLPs from the Transmit Buffer and thus makes room for new TLPs, the core asserts vc_tx_credit_return_o==1 for one clock cycle and places the number of NH credits being returned on vc_tx_credit_nh_o[11:0]. In this manner you can manage sending only enough non-posted TLPs that the core can hold in its Transmt Buffer. This permits you to know when non-posted TLPs would be blocked and thus send posted and/or completion TLPs instead. This is important for avoiding deadlocks and also keeps non-posted TLP blockage from reducing posted and completion throughput. Should the core receive more non-posted TLPs than the core can store in its non-posted TLP transmit storage, the core pauses TLP transmission rather than allow an overflow to occur. Thus users that do not wish to use the Transmit Credit Interface may ignore this interface provided they are willing to permit blocked non-posted TLPs from also blocking following posted and completion TLPs.
vc_tx_credit_return_i	Output	As the core forwards transmitted TLPs from the Transmit Buffer and thus makes room for new TLPs, the core asserts vc_tx_credit_return_o==1 for one clock cycle and places the number of NH credits being returned on vc_tx_credit_nh_o[11:0].
vc_tx_credit_nh_o[11:0]	Output	Number of NH credits to return == vc_tx_credit_nh_o[11:0].

2.2.3.3. Receive Interface Port Descriptions

The Receive Interface is the mechanism with which you receive PCI Express TLPs from the PCI Express link partner. TLPs are received and presented on the Receive Interface in the same format defined in the PCI Express Specification; you receive the complete Transaction Layer Packets (TLPs) comprised of a 3 or 4 DWORD TLP header, data payload (if present), and TLP Digest (ECRC, if present).

Only TLPs which were received without errors, and which were not nullified, are presented on the Receive Interface, so the user logic only needs to handle valid received TLPs.

- The core checks received TLPs for transmission errors (Sequence Number or LCRC error) and negotiates replay of TLPs with the link partner as required.
- The core discards TLPs which were nullified by the link partner during transmission (TLP was received without transmission errors and with EDB instead of END framing).
- The core checks received TLPs which were received without transmission errors and without being nullified for Malformed TLP due to length errors. If the core determines that a received TLP is malformed due to length (TLP length calculated from the received TLP Header Format and Type, Length, and TLP Digest does not match the received TLP length), the core discards the TLP and report the error.
- The core checks received TLPs which were received without transmission errors, without being nullified, and which were not Malformed TLP due to length errors against its Configuration Registers to determine the validity of the TLP. If the TLP fails to hit an enabled resource or is malformed due to its content (invalid Traffic Class, invalid Format and Type, invalid Byte Enables, etc), the core discards the TLP and reports the error.
- If the TLP passes all of the above checks, then it is considered a valid TLP and is forwarded onto the Receive Interface for the user's logic to consume. The core strips the Physical Layer framing (STP/END/EDB) and Data Link Layer Sequence Number and Link CRC (LCRC) before presenting the TLPs to you on the Receive Interface. The core does not strip the received TLP ECRC (if present) because some user designs require forwarding the ECRC either to transmit the TLP out another PCIe port or because the ECRC value is checked at a later point in the user's data path in order to continue the ECRC error detection protection for a larger portion of their receive data path. If an ECRC is present in the TLP, the core checks the validity of the received ECRC and reports detected ECRC errors on the Receive Interface.

The core also decodes received TLPs against its Configuration Registers and provides the transaction decode information on the Receive Interface such that the TLP can be directed to the appropriate destination without the need for you to parse the TLP until its destination. For example, if the received TLP is an I/O or Memory write or read request, the Base Address Register (BAR) resource that was hit is indicated and if the TLP is a completion, the TLP's tag field is provided. The core also provides additional useful transaction attributes.

The Receive Interface ports are listed in Table 2.24.

See the timing diagrams in the Receive Interface Example Transactions section for additional details.

Table 2.12. Receive Interface

Port	Type	Description
vc_rx_valid_o	Output	vc_rx_valid_o == 1 when the other vc_rx_* output ports are valid. 0 otherwise. A data transfer occurs when vc_rx_valid_o == 1 and vc_rx_ready_i == 1.
vc_rx_ready_i	Input	The user sets vc_rx_ready_i == 1 whenever the user logic is ready to accept received TLP data. A data transfers occur when vc_rx_valid_o == 1 and vc_rx_ready_i == 1. For maximum throughput, you must consume TLPs at the rate that they are presented (vc_rx_ready_i == 1 on all clock cycles). However, since a minimum of 64-bits of packet framing, Sequence Number, and Link CRC are stripped from the TLP prior to appearing on this interface, you can allow two unutizlized clocks for each TLP without impacting throughput.
vc_rx_sel_o[1:0]	Output	Receive TLP type indicator: 0 == Posted Request (write request) 1 == Non-Posted Request (request requiring a completion) 2 == Completion (completion to a previous request)

Port	Type	Description
		3 == Reserved vc_rx_sel_o is useful for steering the TLP to the appropriate processing logic. For example, Posted Requests should be directed to receive write logic while Non-Posted Requests should be directed to receive read logic. Completions should be directed back to the original read request source using the TLP Tag information. vc_rx_sel_o is valid for the entire TLP (from vc_rx_sop_o == 1 through vc_rx_eop_o == 1)
vc_rx_cmd_data_o[12:0]	Output	Received TLP Type Indicator vc_rx_cmd_data_o provides information about the received TLP to facilitate user TLP processing. This port has a different meaning in Root Port Modes of operation and Endpoint operation. See Section 2.1.7.2 for details. The Lattice PCIe X1 Core decodes received TLPs to determine their destination. The core passes this information to the Transaction Layer Receive Interface by asserting the appropriate bits in this field. Individual bits of vc_rx_cmd_data_o[12:0] carry the following information: Bits[12:10] – Traffic Class[2:0] of the TLP Bit[9] – Completion/Base Address Region indicator (1) indicates the TLP is a Completion or Message routed by ID (0) indicates the TLP is a read or write request or a Message routed by address that hit an enabled Base Address Region. Bits[8:0] are decoded differently based on the value of Bit[9] Bits[8:0] : Completion or Message routed by ID (Bit[9] == 1) Bits[8] – Reserved Bits[7:0] – Tag[7:0] – The Requestor Tag contained in the TLP. Tag[7:0] is used to associate the completion with its orginal source request. This field is reserved if the TLP is a message rather than a completion. Bits[8:0] : Read or write request or a Message routed by address that hit an enabled Base Address Region (Bit[9] == 0) Bit[8] – When (1), the packet is a "write" transaction; when (0), the packet is a "read" transaction Bit[7] – When (1), the packet requires one or more Completion transactions as a response; (0) otherwise Bit[6] – (1) the TLP hit the Expansion ROM else (0) Bit[5] – (1) the TLP hit Base Address Region 5 else (0) Bit[4] – (1) the TLP hit Base Address Region 4 else (0) Bit[3] – (1) the TLP hit Base Address Region 3 else (0) Bit[2] – (1) the TLP hit Base Address Region 2 else (0) Bit[1] – (1) the TLP hit Base Address Region 1 else (0) Bit[0] – (1) the TLP hit Base Address Region 0 else (0) vc_rx_cmd_data_o is valid for the entire packet (from vc_rx_sop_o == 1 through vc_rx_eop_o == 1).
vc_rx_f_o[1:0]	Output	Function Hit by the Received TLP vc_rx_f_o indicates which PCIe Function received the TLP. vc_rx_f_o == 0 indicates Function #0. vc_rx_f_o == 1 indicates Function #1, ..., etc.

Port	Type	Description
vc_rx_sop_o	Output	Start of TLP indicator. vc_rx_sop_o == 1 coincident with the first vc_rx_data_o word in each TLP. 0 otherwise.
vc_rx_eop_o	Output	End of TLP indicator. vc_rx_eop_o == 1 coincident with the last vc_rx_data_o word in each TLP. 0 otherwise.
vc_rx_err_ecrc_o	Output	Received TLP ECRC error indicator vc_rx_err_ecrc_o == 1 from vc_rx_sop_o to vc_rx_eop_o inclusive for received TLPs which contain a detected ECRC error. 0 otherwise. vc_rx_err_ecrc_o only reports ECRC errors when ECRC checking is enabled. ECRC checking is enabled by software through the AER Capability. TLPs with ECRC errors are presented on the Receive Interface in the same format that they are received including the TLP Digest (ECRC). ECRC errors are serious, uncorrectable errors. The user design must decide how to handle/recover from the error including whether to use the TLP with the error. ECRC errors need for higher level software to correct/handle the error. PCIe does not have a standard mechanism for retransmitting TLPs end to end as it does for a given PCIe link (via the LCRC/SequenceNumber and Replay mechanisms).
vc_rx_data_o[31:0]	Output	Received TLP Data Received TLP data comprises a complete Transaction Layer Packet (TLP) as defined by the PCI Express Specification including the entire TLP header, data payload (if present), and TLP Digest (ECRC, if present). The core strips the packet's STP/END/EDB framing, Sequence Number, and Link CRC (LCRC) prior to the TLP appearing on this interface. The core checks TLP ECRC, when present and when checking is enabled, and can be optionally enabled to remove the ECRC from the TLP.
vc_rx_data_p_o[3:0]	Output	Received TLP Data Parity Even parity of associated vc_rx_data_o: vc_rx_data_p_o[i] == ^(vc_rx_data_o[((i+1)*8)-1:(i*8)]) vc_rx_data_p_o is valid for all bytes in vc_rx_data_o that contain a portion of a TLP (header, payload (if present), and TLP digest (ECRC, if present). vc_rx_data_p_o is not valid for any trailing, unused bytes in the final vc_rx_data_o word in a TLP.

2.2.3.4. Receive Credit Interface Port Descriptions

The Receive Credit Interface provides the means for flow control of non-posted receive transactions between the core Receive Buffer and user receive TLP logic. This is important for allowing Posted and Completion TLPs to continue to make progress when non-posted TLPs are blocked (which can be necessary in some cases to avoid potential deadlock conditions). The amount of non-posted TLP storage in the user's design is communicated on the Receive Credit Interface. The core uses this interface to limit the simultaneously outstanding receive non-posted TLPs to the amount of non-posted TLPs that the user design advertises that it is able to aborsb.

Note that link partner/core receive TLP flow control is not managed via this interface; the core manages Receive Buffer flow control between itself and the PCIe link partner transmit gating function without user intervention.

Table 2.13. Receive Credit Interface

Port	Type	Description
vc_rx_credit init_i	Input	When the user Transaction Layer logic is ready to accept non-posted TLP reception, you assert vc_rx_credit init_i == 1 for one clock cycle and on the same cycle indicates the non-posted TLP Header storage capacity of the user design on vc_rx_credit_nh_i[11:0]. You must initialize vc_rx_credit_init_i shortly (within 10s of clocks) after u_tl_link_up for Root Port and shortly after u_dl_link_up for Endpoint. Holding off credit initializion for an extended period can cause received non-posted TLP transactions to timeout in the source component which may be serious errors. The core limits simultaneous outstanding non-posted receive TLPs on the Receive Interface to ensure no more than the initialized NH credits are simultaneously outstanding to user receive TLP logic. Once you process/forward received non-posted TLPs such that more room is available to receive new non-posted TLPs, the user asserts vc_rx_credit_return_i==1 for one clock cycle and places the number of NH credits being returned on vc_rx_credit_nh_i[11:0]. In this manner, you can limit the outstanding core receive TLPs to the user design. This permits the core to know when non-posted TLPs would be blocked and thus send posted and/or completion TLPs to the user design instead. This is important for avoiding deadlocks and also keeps non-posted TLP blockage from reducing posted and completion throughput. If you do not wish to implement flow control of NH credits via this interface, then when vc_rx_credit_init==1, vc_rx_credit_nh_inf_i is set to 1 to advertise infinite NH credits. NH cedit flow control is not implemented for links that advertised infinite NH credits.
vc_rx_credit_return_i	Input	Once you processe/forward received non-posted TLPs such that more room is available to receive new non-posted TLPs, you assert vc_rx_credit_return_i==1 for one clock cycle and places the number of NH credits being returned on vc_rx_credit_nh_i[11:0].
vc_rx_credit_nh_i[11:0]	Input	Number of NH credits to return == vc_rx_credit_nh_i[11:0].
vc_rx_credit_nh_inf_i	Input	Infinite NH Credits vc_rx_credit_nh_inf_i: 1==Do not limit TLP reception due to NH credits 0==Limit simultaneously outstanding NH credits to the value of vc_rx_credit_nh_i[11:0] when vc_rx_credit_init was 1.

2.2.3.5. Data Byte Order

The core transmits TLP data in the following byte order:

vc_tx_data_i[7:0], vc_tx_data_i[15:8], vc_tx_data_i[23:16], …

The core receives TLP data in the following byte order:

vc_rx_data_o[7:0], vc_rx_data_o[15:8], vc_rx_data_o[23:16], …

For example, you would transmit, or the core would receive, a 32-bit Memory Read Transaction Layer Packet in the following byte order:

Table 2.14. Receive Credit Interface

vc_tx_data_i/ vc_rx_data_o	First Data Word	Second Data Word	Third Data Word
[7:0]	{R, Fmt[1:0], Type[4:0]}	RequesterID[15:8]	Addr[31:24]
[15:8]	{R, TC[2:0], R[3:0]}	RequesterID[7:0]	Addr[23:16]
[23:16]	{TD, EP, Attr[1:0], R[1:0], Length[9:8]}	Tag[7:0]	Addr[15:8]
[31:24]	Length[7:0]	{LastDWBE[3:0],1stDWBE[3:0]}	{Addr[7:2], R[1:0]}

2.2.3.6. Transaction Layer Interface Control Flow and Error Detection and Correction

The Transaction Layer Interface is used to move all Transaction Layer Packets (TLPs) between the user’s application and the remote PCI Express device. All TLPs on the Transaction Layer Receive and Transmit Interfaces are received, and must be transmitted, in the TLP format described in the PCI Express Base Specification: “Transaction Layer Protocol – Packet” section.

The vc_rx_cmd_data_o port provides useful information about the TLP that is pending on the Receive Interface to enable you to determine the destination of the packet (BAR0, Tag, etc.), its traffic class, and whether it is a write or a read without having to read and parse the TLP. This allows you to optimize their code to reduce latency and relieves you of the necessity to decode the TLP header to determine the packet’s destination.

PCI Express has built in error detection and correction mechanisms for both Transaction Layer Packets (TLPs) which are transferred to/from PCI Express and the Transaction Layer Interface and Data Link Layer Packets (DLLPs) which are used by the core internally for link management.

The Lattice PCIe X1 Core adds the required Physical Layer framing (STP/END/EDB) and Data Link Layer error detection and correction information (Sequence Number/Link CRC) to the TLP packets transmitted on the Transmit Interface. Likewise, when TLP packets are received from PCI Express, the core validates that the packet was received correctly by checking the Physical Layer framing (STP/END/EDB) and Data Link Layer error detection and correction information (Sequence Number/Link CRC). Packets that are received that target user resources are forwarded to you on the Local Receive Interface after stripping the Physical Layer framing (STP/END/EDB) and Data Link Layer error detection and correction information (Sequence Number/Link CRC).

If transmission errors are detected in packet transmission or reception, the core coordinates with the remote PCI Express device (per PCI Express protocol) to retry the transaction and recover from the error. This process occurs without any user intervention. The Lattice PCIe X1 Core logs both corrected and uncorrected errors. This error status information is made available through the status registers and is accessed by system software through the Configuration Registers. The core generates and transmits error Message TLPs (per PCI Express Specification) to the remote PCI Express device in response to many types of errors when operating as an Upstream Port (Endpoint). When operating as a Downstream Port (Root Port), errors status information is redirected onto ports and is also captured in the core’s Configuration regsiters. See Section 2.1.6. for details.

ECRC (TLP Digest) generation and checking is a core option. When ECRC generation support is enabled by software (AER Capability : ECRC Generation Enable == 1), the core generates and adds ECRC to all transmitted TLPs (except those that already contain an ECRC). When ECRC checking support is enabled by software (AER Capability : ECRC Check Enable == 1), the ECRC fields present in received TLPs are checked for validity and any errors are noted on the Receive Interface and are reported in the AER Capability. The core does not modify the ECRC or TD (TLP Digest == ECRC present indicator) fields on received TLPs and passes these fields onto the Receive Interface as received.

The Lattice PCIe X1 Core handles TLPs that are Type 0 Configuration Transaction Requests (to its own Configuration Registers), Messages Requests for link management, TLPs that don't hit an enabled resource, and any requests that the core determines are malformed. If the core handles a TLP, then that TLP is consumed by the core (and not forwarded) and the core transmits any required Completion packet(s), generates any required error messages, and logs any required errors.

The core has been designed to perform as much of the commonly needed PCI Express functionality for you as possible. In general, you must only consume and generate TLPs directly related to transferring data and control information between your application and remote PCI Express devices.

2.2.3.7. Transmit Interface Example Transactions

2.2.3.7.1. Minimum Timing of vc_tx_ready_o

Figure 2.14 illustrates the minimum timing of vc_tx_ready_o.

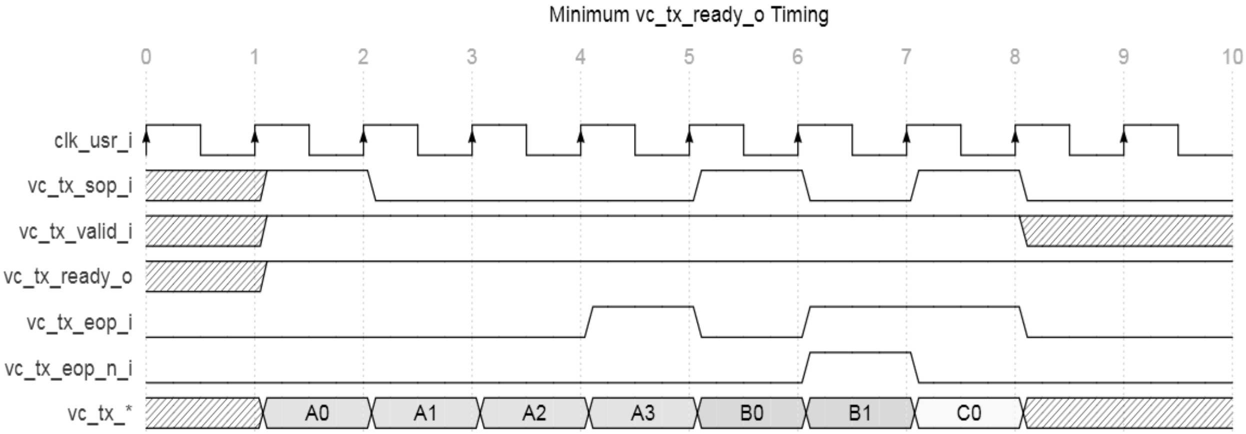


Figure 2.14. Transmit Interface Example Transactions – Minimum vc_tx_ready_o Timing

Transaction A begins on cycle 2 with the assertion of vc_tx_sop_i and ends on cycle 5 with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers with minimum timing with vc_tx_valid_i==vc_tx_ready_o==1 on cycles 2-5.

Transaction B begins immediately after Transaction A on cycle 6 with the assertion of vc_tx_sop_i and ends on cycle 7 with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers with minimum timing with vc_tx_valid_i==vc_tx_ready_o==1 on cycles 6-7. Transaction B is nullified (dropped during forwarding) because vc_tx_eop_i_n==1 when vc_tx_eop_i==1 on cycle 7.

Transaction C begins immediately after Transaction B on cycle 8 with the assertion of vc_tx_sop_i and ends on the same cycle with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers with minimum timing with vc_tx_valid_i==vc_tx_ready_o==1 on cycle 8.

2.2.3.7.2. Wait State Timing of vc_tx_ready_o

Figure 2.15 illustrates some of the possible wait state timing of vc_tx_ready_o.

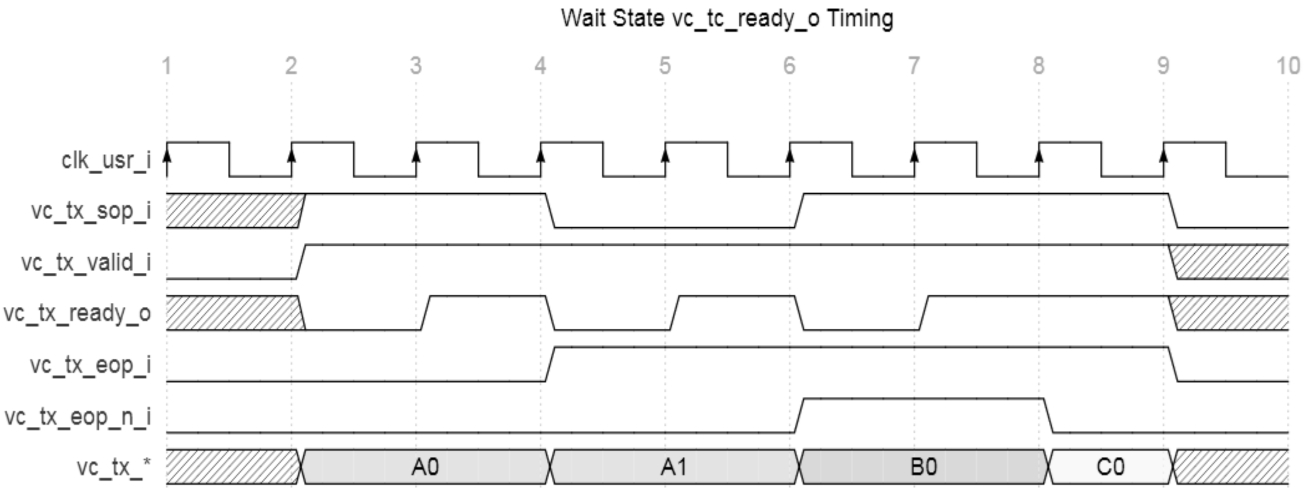


Figure 2.15. Transmit Interface Example Transactions – Example Wait State vc_tx_ready_o Timing

Transaction A begins on cycle 2 with the assertion of vc_tx_sop_i and ends on cycle 5 with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers only on cycles 3 and 5 when vc_tx_valid_i==vc_tx_ready_o==1.

Transaction B begins immediately after Transaction A on cycle 6 with the assertion of vc_tx_sop_i and ends on cycle 7 with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers only on cyle 7 when vc_tx_valid_i==vc_tx_ready_o==1. Transaction B is nullified (dropped during forwarding) because vc_tx_eop_i_n==1 when vc_tx_eop_i==1 on cycle 7.

Transaction C begins immediately after Transaction B on cycle 8 with the assertion of vc_tx_sop_i and ends on the same cycle with the assertion of vc_tx_eop_i==vc_tx_valid_i==vc_tx_ready_o==1. The packet transfers with minimum timing (no wait states) with vc_tx_valid_i==vc_tx_ready_o==1 on cycle 8.

2.2.3.8. Transmit Interface Considerations

The following considerations are provided to simplify logic using the Transmit Interface and to address common problems which should be avoided:

- For each TLP that you transmit, the core adds a minimum of 2-bytes of STP/END/EDB framing, a 2-byte Sequence Number, and a 4 byte Link CRC for a total of 8 bytes (64-bits). These additional 8 bytes, which the core transmits but which do not appear on vc_tx_data_i, provide you with the flexibility of not using every clock cycle on the Transmit Interface. This flexibility can be useful to simplify user logic and improve design timing closure.
- Completions which are transmitted in response to a previously received non-posted request must reflect the Traffic Class, Requestor ID, Tag, and Attributes of the original request. While most of these are obvious, it may not be obvious to reflect the Attributes, and this is known to cause problems on some systems.
- When the link trains at less than full width or speed, vc_tx_ready_o is gapped in relation to the number of lanes being used and the number of lanes available in the core. User's should remember to include a simulation case which forces the link into lower than full-width and/or speed to test that their logic properly handles the gapping of vc_tx_ready_o and the corresponding lower data transfer rate in this case.
- At the same time that TLPs are transmitted over PCI Express, they are placed into a Replay Buffer in case they need to be replayed due to transmission errors. The core negotiates the Replay process in conjunction with the remote PCI Express Device and does not require any user intervention. You can monitor the frequency of replays, if desired, by monitoring the appropriate error status registers.
- The Lattice PCIe X1 Core interface has been designed to support high throughput applications. Small interruptions in transmissions occur, however, as the core periodically needs to transmit link management DLLPs and SKP

Ordered Sets and may also need to transmit error messages, configuration write/read completions, interrupt TLPs, etc.

- The Lattice PCIe X1 Core handles all Data Link Layer functionality as well as most of the error cases for you. To accomplish these functions, the core occasionally delays a user's Local Transmit Interface requests while it completes its own TLP transmissions for these purposes. All of the core's TLP transmissions are short so should delay a user's request for only a few clock cycles. The core transmits DLLPs used for link maintenance, TLP messages to communicate errors, interrupt TLPs, and completions to notify the system of malformed or un-routable TLP requests.
- If user TLP transmit requests are delayed for extended periods of time, this may be due to insufficient link partner Receive Buffer space or local Replay Buffer space or due to the link having to wake from a lower power state or recover from an error before transmission can occur.

2.2.3.9. Receive Interface Example Transactions

2.2.3.9.1. Minimum Timing of vc_rx_ready_i

Figure 2.16 illustrates the minimum timing of vc_rx_ready_i.

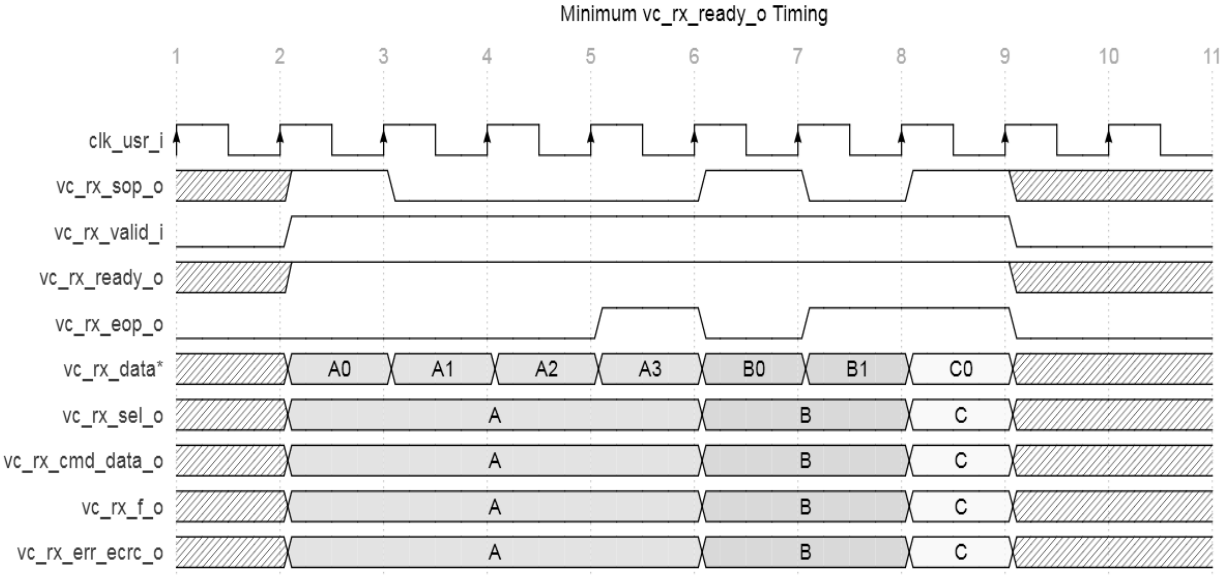


Figure 2.16. Receive Interface – Minimum vc_rx_ready_i Timing

Transaction A begins on cycle 2 with the assertion of vc_rx_sop_o==vc_rx_valid_o==1 and ends on cycle 5 with the assertion of vc_rx_eop_o==vc_rx_valid_o==vc_rx_ready_i==1. The packet transfers with minimum timing since vc_rx_valid_o == vc_rx_ready_i == 1 on cycles 2-5. Data transfers on cycles 2-5.

Transaction B begins immediately after Transaction A on cycle 6 with the assertion of vc_rx_sop_o==vc_rx_valid_o==1 and ends on cycle 7 with the assertion of vc_rx_eop_o==vc_rx_valid_o==vc_rx_ready_i==1. Data transfers on cycles 6-7.

Transaction C begins immediately after Transaction B on cycle 8 with the assertion of vc_rx_sop_o==vc_rx_valid_o==1 and ends on the same cycle since vc_rx_eop_o==vc_rx_ready_i==1 are also asserted. Data transfers on cycle 8.

2.2.3.9.2. Wait State Timing of vc_rx_ready_i

Table 2.17 illustrates some of the possible wait state timing of vc_rx_ready_i.

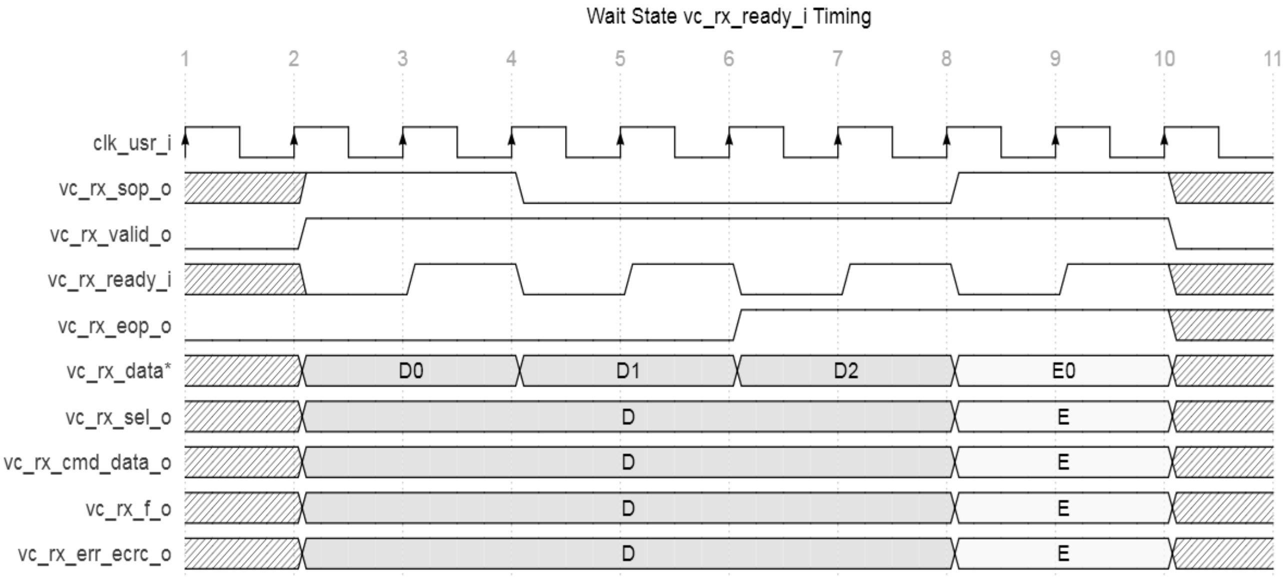


Figure 2.17. Receive Interface – Wait State vc_rx_ready_i Timing

Transaction D begins on cycle 2 with the assertion of vc_rx_sop_o==vc_rx_valid_o==1 and ends on cycle 7 with the assertion of vc_rx_eop_o==vc_rx_valid_o==vc_rx_ready_i==1. The packet transfer wait states due to vc_rx_ready_i==0 on cycles 2, 4, and 6. Data transfers on cycles 3, 5, and 7.

Transaction E begins on cycle 8 with the assertion of vc_rx_sop_o==vc_rx_valid_o==1 but is wait stated due to vc_rx_ready_i==1 on cycle 8. On cycle 9 the transaction completes with vc_rx_sop_o==vc_rx_eop_o==vc_rx_valid_o==vc_rx_ready_i==1.

2.2.3.10. Receive Interface Considerations

The following considerations are provided to simplify logic using the Receive Interface and to address common problems which should be avoided:

- For each TLP that you receive, the core strips a minimum 2-bytes of STP/END/EDB framing, a 2-byte Sequence Number, and a 4 byte Link CRC for a total of 8 bytes (64-bits). These additional 8 bytes, which the core receives but which do not appear on vc_rx_data_o, provides you with the flexibility of not using every clock cycle on the Receive Interface. This flexibility can be useful to simplify user logic and improve design timing closure.
- TLPs that appear on the Receive Interface have passed the Physical Layer and Link Layer error detection and correction logic and can be assumed to be free of transmission errors. When the core receives a TLP with a STP/END/EDB framing, Sequence Number, or Link CRC error, the core coordinates retransmission of the TLP with the remote PCI Express device and only forwards packets that pass transmission error checks onto to the Receive Interface.
- TLPs that are received from PCI Express are decoded for validity against the core's Configuration Registers and are only forwarded to the Receive Interface if they hit an enabled resource, so you only need to handle valid TLPs which target user resources. TLPs which do not hit user resources are terminated by the core and the appropriate error message and response is handled by the core on your behalf.
- The Lattice PCIe X1 Core handles all Data Link Layer functionality as well as most of the Transaction Layer error cases for you. The core consumes Configuration Transactions, Messages, and TLPs which do not map to user resources and transmits the appropriate response. TLPs which are handled by the core do not appear on the Receive Interface.
- User logic which is mastering read requests (for DMA, etc) assigns a Tag to each read request that is transmitted. The core provides the Tag of each received completion on vc_rx_cmd_data_o to allow user logic to route completions from different sources to their destination without having to parse the TLP for Tag information. The

core does not track the outstanding tags that are in use by the user. If a completion is received with a Tag that does not correspond to an outstanding user read request, then you must report the error.

2.2.4. Lattice Memory Mapped Interface (LMMI)

The Lattice PCIe X1 Core implements a bus for configuring core options and obtaining core status. The Core Configuration and Status Registers (CSR) are made accessible to the user design through the Lattice Memory Mapped interface (LMMI).

Table 2.15. Lattice Memory Mapped interface (LMMI)

Port	Type	Description
usr_lmmi_clk_i	Input	Positive edge clock
usr_lmmi_resetn_i	Input	Active low, asynchronous assert, synchronous de-assert reset
usr_lmmi_offset_i [16:2]	Input	Register DWORD offset. usr_lmmi_offset_i [16] == 0 → Link Layer CSR Access usr_lmmi_offset_i [16] == 1 → PHY CSR Access Link Layer registers: usr_lmmi_offset_i [15:2] → word aligned offset PHY registers: usr_lmmi_offset_i [7:2] → word aligned offset Example: (1) Link Layer Register access: mgmt_ftl_bar0 register mgmt_ftl_BASE = 0x4000, bar0 offset = 0x60 reg_byte_offset [15:0] = 0x4060 usr_lmmi_offset_i [16:2] = {1'b0, reg_byte_offset [15:2]} (2) PHY Register access: reg09 (Tx Impedance ratio) reg_byte_offset [7:0] = 0x09 usr_lmmi_offset_i [16:2] = {1'b1,8'd0, reg_byte_offset [7:2]}
usr_lmmi_request_i	Input	Start Transaction (1==Active; 0==Otherwise) A transaction is started when usr_lmmi_request_i==usr_lmmi_ready_o==1. Consecutive request should be done with at least 1 clock period wait cycle. (i.e. usr_lmmi_request_i should de-assert first after a successful transaction before making another request) When usr_lmmi_request_i==usr_lmmi_ready_o==1, usr_lmmi_wr_rdn_i, and usr_lmmi_offset_i must be valid and describe the transaction to execute; if the transaction is a write as indicated by usr_lmmi_wr_rdn_i==1, usr_lmmi_wdata_i must also be valid.
usr_lmmi_wr_rdn_i	Input	Direction (1==Write, 0==Read)
usr_lmmi_wdata_i [31:0]	Input	Write data
usr_lmmi_rdata_o [31:0]	Ouptut	Read data
usr_lmmi_ready_o	Output	Slave is ready to start a new transaction (1==Ready; 0==Not ready)
usr_lmmi_rdata_valid_o	Output	Read transaction is complete and usr_lmmi_rdata_o contains valid data (1==Valid; 0==Othwerwise)

2.2.5. Legacy Interrupt Interface

The Legacy Interrupt Interface enables you to generate interrupts.

2.2.5.1. Legacy Interrupt Interface Port Descriptions

The Legacy Interrupt Interface ports are described in Table 2.16.

Table 2.16. Legacy Interrupt Interface

Port	Type	Function
legacy_interrupt_i [NUM_FUNCTIONS-1:0]	Input	legacy_interrupt_i is used to generate Legacy interrupts on the PCI Express link. legacy_interrupt_i has one input for each Base (Physical) Function. For each function, system software configures the function to use MSI-X, MSI, or Legacy Interrupt mode as part of the PCI enumeration process. User interrupt logic must behave differently depending upon whether the function is enabled for MSI-X, MSI, or Legacy Interrupts. When Legacy Interrupt Mode is enabled, legacy_interrupt_i implements one level-sensitive interrupt (INTA, INTB, INTC, or INTD) for each Base Function. Each functions' interrupt sources must be logically ORed together and input as legacy_interrupt_i [i] for a given function i. Each interrupt source must continue to drive a 1 until it has been serviced and cleared by software at which time it must switch to driving 0. The core ORs together INTA/B/C/D from all functions to create an aggregated INTA/INTB/INTC/INTD. The core monitors high and low transitions on the aggregated INTA/B/C/D and sends an Interrupt Assert message on each 0 to 1 transition and an Interrupt De-Assert Message on each 1 to 0 transition of the aggregated INTA/B/C/D. Transitions which occur too close together to be independently transmitted are merged. When a function has MSI-X or MSI Interrupt Mode enabled, legacy_interrupt_i is not used for that function. MSI-X/MSI interrupts are signaled using MSI-X/MSI Message TLPs which you can be generate and transmit on the Transmit Interface.
legacy_interrupt_o	Output	The core-implemented PCI Express Capability and Advanced Error Reporting Capability contain mechanisms to interrupt system software when events occur. The core asserts mgmt_interrupt_o[i] for Link[i] when an event occurs that per the PCI Express Capability or Advance Error Reporting Enhanced Capability must generate an interrupt. You must merge mgmt_interrupt_o with user interrupt sources into mgmt_interrupt_leg, mgmt_interrupt_msix_req, mgmt_interrupt_msix_ack, and mgmt_interrupt_msix_vector just like it would for any user interrupt. The core outputs mgmt_interrupt_o as an active high level-based interrupt when level-based interrupts are in use (MSI-X_Enable == 0 & MSI_Enable == 0) and as an active high single clock pulse when edge based interrupts are in use ~(MSI-X_Enable == 0 & MSI_Enable == 0). The core implements only one interrupt output. In MSI and MSI-X interrupt modes of operation, core interrupts must use MSI/MSI-X interrupt vector mgmt_interrupt_message_num[4:0]. This value is advertised in the configuration registers for software association of interrupts, so it is important that the user route core generated interrupts on mgmt_interrupt_o to the vector advertised.

2.2.6. Power Management Interface

The Lattice PCIe X1 Core supports optional capabilities such as Dynamic Power Allocation, Latency Tolerance Reporting and Power Budgeting.

2.2.6.1. Power Management Interface Port Descriptions

The Lattice PCIe X1 Core Power Management Interface ports are described in Table 2.17.

Table 2.17. Power Management Interface

Port	Type	Function
pm_dpa_control_en_o	Output	Dynamic Power Allocation Enable. If set to 1, the pm_dpa_control_o should be monitored for change requests in the D0 power substate. If set to 0, pm_dpa_control_o should be ignored.
pm_dpa_control_o[4:0]	Output	Dynamic Power Allocation Control. This output specified the desired D0 power substate. If pm_dpa_control_en_o is set to 1, any change on this output should be used to indicate that the D0 power substate should be changed, and the process should begin to change the power substate. On completion of the substate transition, this output should be compared again with the current power substate. If they don't match, and pm_dpa_control_en_o is set to 1, then a new power transition should begin.
pm_dpa_status_i[4:0]	Input	Dynamic Power Allocation Status. This input should be changed to match the pm_dpa_control_o D0 power substate. If the change on pm_dpa_control_o was to a higher power substate, the pm_dpa_status_i should be updated as soon as the change on pm_dpa_control_o was detected. If the change on pm_dpa_control_o was to a lower power state, the power state change should complete first, and then pm_dpa_status_i should be updated to the lower power state. These rules assure that the device never operates at a power level exceeding the power level reported on pm_dpa_status_i.
pm_ltr_msg_send_i	Input	When operating as an upstream port, set to 1 for one clock to cause an LTR message to be transmitted and 0 otherwise. pm_ltr_snoop_i[12:0], pm_ltr_nosnoop_i[12:0], pm_ltr_snoop_req_i, and pm_ltr_nosnoop_req_i specify the contents of the message. Unused for downstream ports.
pm_ltr_snoop_i[12:0]	Input	See pm_ltr_msg_send_i.
pm_ltr_nosnoop_i[12:0]	Input	See pm_ltr_msg_send_i.
pm_ltr_snoop_req_i	Input	See pm_ltr_msg_send_i.
pm_ltr_nosnoop_req_i	Input	See pm_ltr_msg_send_i.
pm_pb_data_sel_o[7:0]	Output	Power Budgeting Data Select. Specifies an index into a table of Power Budgeting Status by power rail and operating conditions.
pm_pb_data_reg_rd_i [31:0]	Input	Power Budgeting Data Register. This input should be updated with the Power Budgeting Data looked up by the index value on pm_pb_data_sel_o [7:0]. A minimum of 2 Status Conditions are needed for each power rail for which the device requires power. The last entry in the table should be identified by having all 32 bits being set to 0. The bit encoding is: [31:21] reserved (0) [20:18] pb_rail [17:15[pb_op_type [14:13] pb_pm_state [12:10] pb_pm_substate [9:8] pb_data_scale [7:0] pb_base_power See the Power Budgeting Capability section for further details.
user_aux_power_detected_i	Input	Set to 1 if your design implements Aux Power and Aux Power is detected as present else set to 0. The value of this port is reflected in the PCIe

Port	Type	Function
		Configuration Register: PCIe Status : AUX Power Detected.
user_transactions_pending_i [NUM_FUNCTIONS-1:0]	Input	Set to 1 when you have outstanding (not yet completed) non-posted requests else set to 0. The values of this port is reflected in the PCIe Configuration Register: PCIe Status : Transactions Pending.

2.2.7. Configuration Space Register Interface (UCFG)

The UCFG Interface is provided for users to read the current values of the Lattice PCIe X1 Core's PCIe Configuration Registers and also to obtain status of the Lattice PCIe X1 Core that may be needed to implement the user's design.

2.2.7.1. UCFG Interface Port Descriptions

The UCFG Interface is a simple SRAM-like interface that accepts write/read transactions. The UCFG Interface supports multiple outstanding transaction requests to enable higher throughput on the interface. Writes and reads are executed out in the same order that they are accepted on the interface.

Table 2.18. UCFG Interface Port Description

Port	Type	Function
ucfg_valid_i	Input	Transaction Request Valid A transaction is started when ucfg_valid_i==ucfg_ready_o==1. Multiple transactions can be outstanding simultaneously and is executed in the order received. When ucfg_valid_i==ucfg_ready_o==1, ucfg_f_i, ucfg_wr_rd_n_i, and ucfg_addr_i must be valid and describe the transaction to execute; if the transaction is a write as indicated by ucfg_wr_rd_n_i==1, ucfg_wr_be_i and ucfg_wr_data_i must also be valid.
ucfg_ready_o	Output	Transaction Request Ready
ucfg_f_i [2:0]	Input	Transaction Request Function Number Selects which function in a multi-function core is to be accessed. This port is only present for cores that are delivered supporting multiple functions.
ucfg_wr_rd_n_i	Input	Transaction Request Type Selects the type of transaction: 1 = Write 0 = Read
ucfg_addr_i [11:2]	Input	Transaction Request Address Selects the DWORD (32-bit) address of the register accessed by the transaction.
ucfg_wr_be_i[3:0]	Input	Transaction Request Write Byte Enables Selects which bytes to write during a write transaction. For each ucfg_wr_be_i[i]: 1 = Write byte 0 = Do not write byte ucfg_wr_be_i[i] is associated with ucfg_wr_data_i[(i*8)+7:(i*8)].
ucfg_wr_data_i[31:0]	Input	Transaction Request Write Data Selects data to write during a write transaction. ucfg_wr_data_i [7:0] is the least significant byte (byte address offset 2'b00) and ucfg_wr_data_i [31:0] is the most significant byte (byte address offset 2'b11).

Port	Type	Function
ucfg_rd_done_o	Output	Read Transaction Done Indicates that a prior read transaction request has been completed and the resulting data on ucfg_rd_data is valid. 1 = Read done 0 = Otherwise Read transactions complete in the same order that the transaction requests were accepted.
ucfg_rd_data_o [31:0]	Output	Read Transaction Data. Provides the read data for a UCFG read transaction. ucfg_rd_data_o [7:0] is the least significant byte (byte address offset 2'b00) and ucfg_rd_data_o [31:0] is the most significant byte (byte address offset 2'b11).

2.2.7.2. UCFG Address Space

The UCFG Interface may be used to access all of the Lattice PCIe X1 Core's PCIe Configuration Registers. In addition, the UCFG Interface implements a small number of registers to provide useful status that is not available in the PCIe Configuration Registers. The PCIe Configuration Registers are accessed by ucfg_addr_i [11:2] which is a DWORD (32-bit) aligned address.

Table 2.19 highlights the PCIe Configuration Registers that are most likely to be needed for your design.

Table 2.19. UCFG Address Space

Capability/Data	ucfg_addr_i	Description
Configuration Header	0x00-0x0F	PCI Configuration Header The following fields are likely to be needed: Address 0x04 Bit 0 – I/O Space Enable Bit 1 – Memory Space Enable Bit 2 – Bus Master Enable Bit 10 – Interrupt Disable
PCI Express Capability	0x10-0x1E	PCI Express Capability Structure The following fields are likely to be needed: Address 0x10 Bits 29:25 – Interrupt Message Number Address 0x12 Bit 4 – Enable Relaxed Ordering Bits 7:5 – Maximum Payload Size Bit 8 – Extended Tag Enable Bit 11 – Enable No Snoop Bits 14:12 – Maximum Read Request Size Address 0x14 Bits 3:0 – Link Speed Bits 9:4 – Negotiated Link Width Address 0x17 Bit 4 – CRS Software Visibility Enable Address 0x1A Bits 3:0 – Completion Timeout Value Bit 4 – Completion Timeout Disable Bit 6 – AtomicOp Requester Enable Bit 8 – IDO Request Enable Bit 9 – IDO Completion Enable Bit 10 – LTR Mechanism Enable

Capability/Data	ucfg_addr_i	Description
Power Management Capability	0x20-0x21	Power Management Capabilty Structure The following fields are likely to be needed: Address 0x21 Bits 1:0 – PM Power State
MSIX Capability	0x24-0x27	MSIX Capability Structure The following fields are likely to be needed: Address 0x24 Bit 30 – MSIX Function Mask Bit 31 – MSIX Enable Address 0x25 Bits 31:3 – MSIX Table Offset Address 0x26 Bits 31:3 – MSIX PBA Offset
MSI Capability	0x28-0x2D	MSI Capability Structure The following fields are likely to be needed: Address 0x28 Bit 16 – MSI Enable Bits 22:20 – MSI Multiple Message Enable Bit 24 – MSI Per-Vector Mask Capable Address 0x29 Bits 31:0 – MSI Address [31:0] Address 0x2A Bits 31:0 – MSI Address [63:32] Address 0x2B Bits 15:0 – MSI Data Address 0x2C Bits 31:0 – MSI Mask Bits Address 0x2D Bits 31:0 – MSI Pending Bits (writable)
AER Capability	0x40-0x51	AER Capability Structure The following fields are likely to be needed: Address 0x46 Bit 6 – ECRC Generation Enable Bit 8 – ECRC Check Enable

Capability/Data	ucfg_addr_i	Description
Vendor Specific Capability	0x54-0x5F	Vendor Specfic Capability Structure These addresses contain important PCIe X1 Core status that is available via the UCFG Interface. Address 0x5D Bits 15:0 – Number of CH Credits implemented by the Receive Buffer. Bits 31:16 – Number of CD Credits implemented by the Receive Buffer. Address 0x5E Bits 3:0 – Current LTSSM Major State Bits 7:4 – Current LTSSM Minor State Bits 10:8 – Current RX L0s State Bits 15:12 – Current Lane Reverse Status Bits 20-16 – Current PM State Bits 31:24 – Current Function Enable Status Address 0x5F Bits 15:0 – Current Configuration ID Bit 16 – Current Port Type 1 = Downstream Port 0 = Upstream Port Bit 17 – Current PCIe Cfg Register Type 1 = Type 1 (Root Port) 0 = Tyep 0 (Endpoint)
ATS Capability	0x80-0x81	ATS Capability Structure The following fields are likely to be needed: Address 0x81 Bits 20:16 – ATS Smallest Transaction Unit (STU) Bit 31 – ATS Enable
LTR Capability	0xF8-0xF9	LTR Capability Structure The following fields are likely to be needed: Address 0xF9 Bits 12:0 – LTR Max Snoop Latency Bits 28:16 – LTR Max No-Snoop Latency
Error Report Header	0x3F0-0x3F3	Address Range used to Report TLP Error Headers See Section 5.2 for details. Address 0x3F0 Bits 31:0 – TLP Header [31:0] Address 0x3F1 Bits 31:0 – TLP Header [63:32] Address 0x3F2 Bits 31:0 – TLP Header [95:64] Address 0x3F3 Bits 31:0 – TLP Header [127:96]

Capability/Data	ucfg_addr_i	Description
Error Report	0x3F8	This address is used to report errors to the AER capability. Please see the 2.2.7.3.1. User Error Reporting section for details. Bits 7:0 – Error function Number Bits 13:8 – Error flags Bit 8 – Poisoned TLP received Bit 9 – Completion Timeout Bit 10 – Completer Abort Bit 11 – Unexpected Completion Bit 12 – Unsupported Request Bit 13 – Uncorrectable Internal Error Bits 21:16 – Advisory Flags Bit 16 – Poisoned TLP received Bit 17 – Completion Timeout Bit 18 – Completer Abort Bit 19 – Unexpected Completion Bit 20 – Unsupported Request Bit 21 – Uncorrectable Internal Error To report an error, first write the Error Report Header registers with the associated TLP Header of the TLP with the error; write 0s if the error is not associated with a specific TLP. Then write Error Report with the error type. Indicate the function number that should log the error on bits [7:0]. If the error is not associated with a specific function, then write 0 to assign the error to Function [0]. Indicate the error type by setting only one bit of bits [13:8] If the error is of "advisory" type as defined by the PCIe Spec., then set the corresponding bit in bits [21:16] of the error bit set in bits [13:8]. Advisory errors are downgraded to correctable error status so the host generally continues operation unimpeded after an advisory error is reported. If a non-advisory error is reported, the host OS typically faults (blue screen for windows) as these are serious errors that the host OS must either handle via software or halt operation of the OS. The error is only logged when the Error Report register is written with one of bits [13:8] non-zero. The TLP Header associated with the error must already have been written and is taken from the Error Report Header registers.

2.2.7.3. Using the UCFG Interface

The UCFG Interface is primarily intended to obtain the current values of the Lattice PCIe X1 4.0 Core's PCIe Configruation Registers in order to obtain the current values of registers that are needed by the user's design. Although the UCFG interface supports both reading and writing, PCIe Configuration Registers should not be written by the user (with exception for the MSI Pending and Error Reporting functions described below). The PCIe Configuration Registers are written by the Host OS/BIOS during PCIe enumeration and writing them via this interface risks creating incompatibilities with the OS/BIOS that may cause serious errors.

The UCFG Interface provides the current value of the PCIe Configuration Registers. PCIe Configuration Registers may change at any time as the Host OS/BIOS updates them with writes. Also some registers contain current PCIe X1 Core status which changes in response to link events such as speed changes, recovery cycles, width changes, power state changes etc. The UCFG Interface thus should be used to poll the values needed for a user's design as frequently as possible in order to maintain up-to-date values.

There are a small number of registers accessible with the UCFG Interface that are designed to be written:

Address 0x2D	MSI Pending
Address 0x3f0-3F3	Error Report Header
Address 0x3F8	Error Report

Writes to these registers are not part of the core's PCIe Configuration Registers but instead enable you to implement the MSI Pending register and report errors detected in the user design to the core's AER Capability.

2.2.7.3.1. User Error Reporting

The Lattice PCIe X1 Core detects and implements the appropriate response to most error conditions without user intervention. You are generally responsible only for detecting and reporting errors that the core does not have enough information to detect.

It may be desirable for a user hardware design to detect and report the following errors (see relevant error in Table 2.3, Table 2.4, Table 2.5, and Table 2.6 for details):

- Uncorrectable Internal Error
  - Signal if AER Version 0x2 is enabled in the core and user hardware has detected, and was unable to correct, an application specific error that is not reported through another error mechanism.
  - If AER is supported by the core, then the header of the first TLP associated with the error may optionally be logged.
- Poisoned TLP Received w/ Advisory Non Fatal Severity
  - Signal if the core's default poison handling has been disabled (ignore_poison == 1) and you receive a poisoned TLP and the poison is considered to be of Advisory Non-Fatal (less serious) severity. If the data payload of a poisoned packet is used or the poison can be recovered from via software or other mechanism, then the poison should generally be treated as Advisory Non-Fatal since a Non-Fatal error often is fatal to system operation.
  - If AER is supported by the core and the core is operating in Endpoint mode, then an ERR_COR message is requested and transmitted if enabled.
  - If AER is supported by the core, then the header of the poisoned packet must be logged.
- Poisoned TLP w/ Non-Fatal Severity
  - Signal if the core's default poison handling has been disabled (ignore_poison == 1) and you receive a poisoned TLP and the poison is considered to be of Non-Fatal (serious) severity. Handling poison as Non-Fatal severity should be avoided when possible as this is often fatal to system operation.
  - If AER is supported by the core, then the header of the poisoned packet must be logged.
- Unsupported Request
  - A Type0 Vendor Defined Message that is received that is not supported by user logic is an Unsupported Request. This is uncommon since only devices designed to receive Type0 Vendor Defined Messages should receive them. However, compliance tests may require this error to be handled, so it is recommended to implement this check. Receipt of a message with Message Code == 0x7E should cause "Unsupported Request" to be reported unless the user design has been designed to receive these messages.
  - Completions that are received with a reserved Completion Status should be handled as if the Completion Status was Unsupported Request.

- Completion Timeout
  - If a user masters non-posted request (all reads, I/O Write, and Configuration Write) then you are required to implement a completion timeout timer that fires if completions to a non-posted request are not received in the allotted time period (typically ~10 mS). This error check needs to be implemented by user designs that master non-posted requests.
- Completer Abort
  - Signal if permanently unable to process a Request due to a device-specific error condition. Generally, this error would only be signaled if you chose to implement a restricted programming model (require software to always do DWORD size transactions; not support burst transactions, etc.). This is not recommended unless it is known with certainty that the only software that accesses the user design is your own software, which is designed to obey the restricted programming model.
  - If AER is supported by the core, then the header of aborted request must be logged
- Unexpected Completion
  - User should signal if a completion is received but the tag does not match any outstanding requests
  - If the core is enabled for “Target_Only” mode indicating that the user design does not master non-posted requests, then the core considers all completions as Unexpected Completions, discard them, and generate the appropriate response. In this case, you do not have to handle this error.
  - If AER is supported by the core, then the header of the completion must be logged

It is recommended to report the following errors at a minimum:

- Completion Timeout if mastering non-posted requests (for example DMA read requests)
- Unsupported Request for the cases described above
- Unexpected Completion for the case described above
- Poison when the core’s default poison handling has been disabled (ignore_poison == 1)

The UCFG Interface enables you to log errors that they detect into the core’s AER Capability for reporting to software. The process for reporting an error is as follows:

1. Write the Header of the TLP with the eror into the four addresses for Error Report Header. If the error was not generated by a specific TLP, write zeros into Error Report Header.
2. Write the function number and error flags to Error Report to trigger the Core to record the error in the appropriate registers in Configuration Space. One write is need for each function that should receive the error report. In a single function core, the function number is always be 0.

2.2.7.3.2. MSI Pending

The MSI Pending register is used to report MSI Interrupts that are pended in the user design. MSI Pending is a PCIe Configuration Register in the MSI Capability Structure that software uses to obtain status on pending MSI Interrupt vectors.

The MSI Pending register must be written whenever a MSI Interrupt Vector’s pending status changes. A 1 must be written to the associated interrupt vector bit when an interrupt becomes pending and a 0 must be written to indicate that the interrupt is no longer pending.

The MSI Pending register must be updated whenever the status of your pending MSI interrupts changes. If MSI interrupts are not used, writing to the MSI Pending Register is not needed.

2.2.8. AHB-Lite Data Interface

This interface is available if the data interface type selected in the IP generation user interface is "AHB_LITE".

A slave interface is provided for transmit path and a master interface on the receive path for simple interface conversion. You must provide the base addresses (4KB aligned) to be used as destination address when receiving packets. This can be configured through the IP generation user interface or by setting the following parameters:

Example:

PCIE_LOC_ADDR_POSTED = 32'hFFFF_0000

PCIE_LOC_ADDR_NONPOSTED = 32'hFFFF_1000

PCIE_LOC_ADDR_COMPLETION = 32'hFFFF_2000

The received packets is sent to the user's application logic through AHB-Lite master interface with an address starting from the base address that corresponds to the type of TLP and increments as each dword is received.

To transmit a packet, you must forward it through AHB-Lite slave interface using the CSR base address plus an offset of 0x30000 as starting address.

Example:

PCIE_CSR_BASEADR = 32’hC2500000

AHB-Lite Slave start address = 32’hC2500000 + 32’h00030000 = 32’hC2530000

When DMA support is enabled, there are two AHB-Lite master interfaces instead of 1 master and 1 slave interface. The details of DMA operation are discussed in the User Interface Options section.

AHB-Lite interface follows little endian convention, where the least significant byte is stored in the lowest address.

2.2.8.1. AHB-Lite Master 0 Interface Port Descriptions

Table 2.20. AHB-Lite Master 0 Interface Port Descriptions

Port	Type	Function
m_w_haddr_o [31:0]	Output	Bus Address.
m_w_hburst_o [2:0]	Output	Burst type – indicate if the transfer is a single transfer or forms part of a burst. 0 – Singe 1 – Incremental
m_w_hmastlock_o	Output	This is not supported (tied to 0).
m_w_hprot_o [3:0]	Output	This is not supported (tied to 4’b0011).
m_w_hsize_o [2:0]	Output	Indicates the size of the transfer. 0 – Byte 1 – Halfword 2 – Word
m_w_htrans_o [1:0]	Output	Indicates the transfer type of the current transfer. 0 – IDLE 1 – BUSY 2 – NONSEQUENTIAL 3 – SEQUENTIAL
m_w_hwdata_o [31:0]	Output	Write Data.
m_w_hwrite_o	Output	Indicates write or read access. 0 – Read 1 – Write
m_w_hrdata_i [31:0]	Input	Read Data.
m_w_hready_i	Input	When HIGH, indicates that the previous transfer is complete

Port	Type	Function
m_w_hresp_i	Input	Transfer response. 0 – OKAY 1 – ERROR

2.2.8.2. AHB-Lite Master 1 Interface Port Descriptions

This interface is only available when DMA support is enabled.

Table 2.21. AHB-Lite Master 1 Interface Port Descriptions

Port	Type	Function
m_r_haddr_o [31:0]	Output	Bus Address.
m_r_hburst_o [2:0]	Output	Burst type – indicate if the transfer is a single transfer or forms part of a burst. 0 – Singe 1 – Incremental
m_r_hmastlock_o	Output	This is not supported (tied to 0).
m_r_hprot_o [3:0]	Output	This is not supported (tied to 4'b0011).
m_r_hsize_o [2:0]	Output	Indicates the size of the transfer. 0 – Byte 1 – Halfword 2 – Word
m_r_htrans_o [1:0]	Output	Indicates the transfer type of the current transfer. 0 – IDLE 1 – BUSY 2 – NONSEQUENTIAL 3 – SEQUENTIAL
m_r_hwdata_o [31:0]	Output	Write Data.
m_r_hwrite_o	Output	Indicates write or read access. 0 – Read 1 – Write
m_r_hrdata_i [31:0]	Input	Read Data.
m_r_hready_i	Input	When HIGH, indicates that the previous transfer is complete
m_r_hresp_i	Input	Transfer response. 0 – OKAY 1 – ERROR

2.2.8.3. AHB-Lite Slave Interface Port Descriptions

This interface is not available when DMA support is enabled. You have the option to reduce resource utilization by removing the dedicated register interface and mapping the CSRs and PCIe Configuration registers to this interface.

Table 2.22. AHB-Lite Slave Interface Port Descriptions

Port	Type	Function
s_hsel_i	Input	Slave select.
s_haddr_i [31:0]	Input	Bus Address. Bits[1:0] must be tied to 2'b00. Bits[31:19] = CSR Base Address (Bits[18:16] == 3'b011) : Bits[15:12] = 4'h0 Bits[11:2] = dummy address for transmitting TLP, always start from 0 when sending the header. Applicable when mapping of register to slave interface is enabled: (Bits[18:16] == 3'b000) : Bits[15:2] = Link Layer Register offset (Bits[18:16] == 3'b001) : Bits[15:8] = 8'h00 Bits[7:2] = PHY Register offset (Bits[18:16] == 3'b010) : Bits[15:14] = 2'b00 Bits[13:12] = Function number Bits[11:2] = PCIe Configuration Register offset (See the Configuration Space Register Interface (UCFG) and Register Description sections for details.)
s_hburst_i [2:0]	Input	Burst type – indicate if the transfer is a single transfer or forms part of a burst. 0 – Singe 1 – Incremental
s_hmastlock_i	Input	This is not supported (tie to 0).
s_hprot_i [3:0]	Input	This is not supported (tie to 4'b0011).
s_hsize_i [2:0]	Input	Indicates the size of the transfer. 0 – Byte 1 – Halfword 2 – Word
s_htrans_i [1:0]	Input	Indicates the transfer type of the current transfer. 0 – IDLE 1 – BUSY 2 – NONSEQUENTIAL 3 – SEQUENTIAL
s_hwdata_i [31:0]	Input	Write Data.
s_hwrite_i	Input	Indicates write or read access. 0 – Read

Port	Type	Function
		1 – Write
s_hreadyin_i	Input	Ready signal from bus multiplexor. When HIGH, indicates that the previous transfer is complete.
s_hrdata_o [31:0]	Output	Read Data.
s_hready_o	Output	When HIGH, indicates that the previous transfer is complete.
s_hresp_o	Output	Transfer response. 0 – OKAY 1 – ERROR

2.2.9. AHB-Lite Configuration Interface

This interface is available if the register interface type selected in the IP generation user interface is “AHB_LITE”. You must provide a 512 KB aligned base address that is used when accessing the Core CSRs and PCIe Configuration Space registers.

AHB-Lite interface follows little endian convention, where the least significant byte will be stored in the lowest address.

2.2.9.1. AHB-Lite Configuration Interface Port Descriptions

Table 2.23. AHB-Lite Configuration Interface Port Descriptions

Port	Type	Function
c_hsel_i	Input	Slave select.
c_haddr_i [31:0]	Input	Bus Address. Bits[1:0] must be tied to 2’b00. Bits[31:19] = CSR Base Address (Bits[18:16] == 3’b000) : Bits[15:2] = Link Layer Register offset (Bits[18:16] == 3’b001) : Bits[15:8] = 8’h00 Bits[7:2] = PHY Register offset (Bits[18:16] == 3’b010) : Bits[15:14] = 2’b00 Bits[13:12] = Function number Bits[11:2] = PCIe Configuration Register offset (See the Configuration Space Register Interface (UCFG) and Register Description sections for details.)
c_hburst_i [2:0]	Input	Burst type – indicate if the transfer is a single transfer or forms part of a burst. 0 – Singe 1 – Incremental
c_hmastlock_i	Input	This is not supported (tie to 0).
c_hprot_i [3:0]	Input	This is not supported (tie to 4’b0011).

Port	Type	Function
c_hsize_i [2:0]	Input	Indicates the size of the transfer. 0 – Byte 1 – Halfword 2 – Word This should be tied to 3’b010.
c_htrans_i [1:0]	Input	Indicates the transfer type of the current transfer. 0 – IDLE 1 – BUSY 2 – NONSEQUENTIAL 3 – SEQUENTIAL
c_hwdata_i [31:0]	Input	Write Data.
c_hwrite_i	Input	Indicates write or read access. 0 – Read 1 – Write
c_hreadyin_i	Input	Ready signal from bus multiplexor. When HIGH, indicates that the previous transfer is complete.
c_hrdata_o [31:0]	Output	Read Data.
c_hready_o	Output	When HIGH, indicates that the previous transfer is complete.
c_hresp_o	Output	Transfer response. 0 – OKAY 1 – ERROR

2.2.10. APB Configuration Interface

This interface is available if the register interface type selected in the IP generation user interface is "APB". You must provide a 512 KB aligned base address that is used when accessing the Core CSRs and PCIe Configuration Space registers.

2.2.10.1. APB Configuration Interface Port Descriptions

Table 2.24. APB Configuration Interface Port Descriptions

Port	Type	Function
c_apb_pclk_i	Input	Clock.
c_apb_preset_n_i	Input	Active-low asynchronous assert, synchronous de-assert reset.
c_apb_paddr_i [31:0]	Input	Bus Address. Bits[1:0] must be tied to 2'b00. Bits[31:19] = CSR Base Address (Bits[18:16] == 3'b000) : Bits[15:2] = Link Layer Register offset (Bits[18:16] == 3'b001) : Bits[15:8] = 8'h00 Bits[7:2] = PHY Register offset (Bits[18:16] == 3'b010) : Bits[15:14] = 2'b00 Bits[13:12] = Function number Bits[11:2] = PCIe Configuration Register offset (See the Configuration Space Register Interface (UCFG) and Register Description sections for details.)
c_apb_psel_i	Input	Slave select.
c_apb_penable_i	Input	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
c_apb_pwrite_i	Input	Indicates write or read access. 0 – Read 1 – Write
c_apb_pwdata_i [31:0]	Input	Write Data.
c_apb_prdata_o [31:0]	Output	Read Data.
c_apb_pready_o	Output	Ready. The slave uses this signal to extend an APB transfer.
c_apb_pslverr_o	Output	Slave error. 0 – Otherwise 1 – Error

2.2.11. AXI4-Stream Data Interface

This interface is available if the data interface type selected in the IP generation user interface is "AXI4_STREAM".

2.2.11.1. AXI4-Stream Master Interface Port Descriptions

Table 2.25. AXI4-Stream Master Interface Port Descriptions

Port	Type	Function
m_tready_i	Input	Destination ready. 1==Ready, 0==Not ready. A transfer occurs when m_tvalid_o==m_tready_i==1.
m_tvalid_o	Output	Source valid. 1==Valid, 0==Not valid.
m_tdata_o [31:0]	Output	Received TLP Data Received TLP data comprises a complete Transaction Layer Packet (TLP) as defined by the PCI Express Specification including the entire TLP header, data payload (if present), and TLP Digest (ECRC, if present). The core strips the packet's STP/END/EDB framing, Sequence Number, and Link CRC (LCRC) prior to the TLP appearing on this interface. The core checks TLP ECRC, when present and when checking is enabled, and can be optionally enabled to remove the ECRC from the TLP.
m_tstrb_o [3:0]	Output	Byte qualifier that indicates whether the content of the associated byte of m_tdata_o is processed as a data byte or a position byte. This is always 4'hF.
m_tkeep_o [3:0]	Output	Byte qualifier that indicates whether the content of the associated byte of m_tdata_o is processed as part of the data stream. Associated bytes that have the m_tkeep_o byte qualifier deasserted are null bytes and can be removed from the data stream. This is always 4'hF.
m_tlast_o	Output	End of TLP indicator. m_tlast_o == 1 coincident with the last m_tdata_o word in each TLP. 0 otherwise.
m_tid_o [7:0]	Output	Data stream identifier that indicates different streams of data.
m_tdest_o [3:0]	Output	m_tdest_o provides routing information for the data stream. Bits[3:2] – Function Hit by the Received TLP Bits[1:0] – Receive TLP type indicator: 0 == Posted Request (write request) 1 == Non-Posted Request (request requiring a completion) 2 == Completion (completion to a previous request)

2.2.11.2. AXI4-Stream Slave Interface Port Descriptions

Table 2.26. AXI4-Stream Slave Interface Port Descriptions

Port	Type	Function
s_tvalid_i	Input	Source valid. 1==Valid, 0==Not valid.
s_tdata_i [31:0]	Input	TLP data to transfer.
s_tstrb_i [3:0]	Input	Unused. Set to 4'hF.
s_tkeep_i [3:0]	Input	Unused. Set to 4'hF.
s_tlast_i	Input	End of packet indicator. Set == 1 coincident with the last vc_tx_data_i word in each TLP.
s_tid_i [7:0]	Input	Unused. Set to 8'h00.
s_tdest_i [3:0]	Input	Unused. Set to 4'h0.
s_tready_o	Output	Destination ready. 1==Ready, 0==Not ready. A transfer occurs when s_tvalid_i==s_tready_o==1.

2.2.12. DMA Interrupt Interface

This interface is only available if DMA support is enabled.

2.2.12.1. DMA Interrupt Interface Port Descriptions

Table 2.27. DMA Interrupt Interface Port Descriptions

Port	Type	Function
int_normal_o	Output	Normal Interrupt. When asserted, it means that one or more interrupt status bits classified as "normal" are asserted. Refer to interrupt register descriptions from the interrupt_type Register 0x20 section to the interrupt_status Register 0x2C section for details.
int_critical_o	Output	Critical Interrupt. When asserted, it means that one or more interrupt status bits classified as "critical" are asserted. Refer to interrupt register descriptions from the interrupt_type Register 0x20 section to the interrupt_status Register 0x2C section for details.

2.3. Attribute Summary

The PCIe X1 Core attributes are configurable through the IP Catalog's Module/IP wizard of the Lattice Radiant Software. Refer to Table 2.28 for the description of each attribute.

Table 2.28. User Interface Attribute Descriptions

Attribute	Values	Description
General		
PCIe Device Type	PCIe X1, Root Port	Configures the IP as Endpoint or Root Port. For Endpoint: Port type is set to upstream, Configuration Register Type is set to Type 0, PCIe Capability Device Port Type is set to (0) – PCI express endpoint, PCIe Capability Slot implemented is set to (0) For Root Port: Port type is set to downstream, Configuration Register Type is set to Type 1, PCIe Capability Device Port Type is set to (4) – Root Port, PCIe Capability Slot implemented is set to (1) See the EP/RP Configuration Settings section for details. Parameter: DEVICE_TYPE = {“PCIe X1”, “Root Port”}
PCIe Link Width	X1	Display only.
Target Link Speed	2.5G, 5.0G	Initial value of Target Link Speed Configuration Register. Determines the maximum initial link speed which can be reached during initial training. Must be set to the lesser of the maximum speed supported by the core and the maximum speed at which the user desires the core to operate. Parameter: MGMT_FTL_INITIAL_TARGET_LINK_SPEEED = {0,1}
Number of Physical Functions	1- 4	Set the number of enabled functions. Parameter: NUM_FUNCTIONS = {1,2,3,4}
Use TLP Interface	Checked Unchecked	Set to use the default hard IP interface, otherwise select interface from the available standard interface below. The default interface uses the hard IP's TLP data interface (Rx and Tx) and Lattice Memory Mapped Interface (LMMI) for register access. Parameter: USE_DEFAULT_IF = {0,1}
Data Interface Type	AHB_LITE, AXI4_STREAM, Default (Rx/Tx TLP IF)	Available if default interface is not selected. The selected interface replaces the native TLP data interface of the hard IP by adding a soft logic bridge. Parameter: USR_MST_IF_TYPE = {"TLP", "AHB_LITE", "AXI4_STREAM"} USR_SLV_IF_TYPE = {"TLP", "AHB_LITE", "AXI4_STREAM", "NONE"}

Attribute	Values	Description
Enable DMA Support	Checked Unchecked	Available if selected Data Interface Type is AHB-Lite. Add DMA support soft logic for a more efficient data transfer. See section User Interface Options for details. Parameter: EN_DMA_SUPPORT = {"Disable", "Enable"}
Master (Data) Interface Type	AHB_LITE, AXI4_STREAM, TLP	Display only.
Slave (Data) Interface Type	AHB_LITE, AXI4_STREAM, TLP, NONE	Display only.
Enable Mapping of Register to Slave (Data) Interface	Checked Unchecked	Available if selected Data Interface Type is AHB-Lite. Set to reduce the interface and be able to access the registers through the slave data interface. Parameter: EN_MAP_CSR2SLVIF = {"Disable", "Enable"}
Register Interface Type	APB, AHB_LITE, Default (LMMI), NONE	Available if default interface is not selected and mapping of register to slave data interface is not selected. The selected interface replaces the native Lattice Memory Mapped Interface (LMMI) of the hard IP by adding a soft logic bridge. Parameter: USR_CFG_IF_TYPE = {"LMMI", "APB", "AHB_LITE", "NONE"}
PCIe CSR Base Address	(Hex, 512 KiB aligned) 0000_0000 – FFF8_0000	Available if default interface is not selected. This is a 512 KiB aligned base address used to access both Hard IP and Soft IP registers as well as PCIe Configuration Space Registers. Parameter: PCIE_CSR_BASEADR = {13’h0000 – 13’h1FFF}
Hard IP Core CSR Reset Mode	Soft Reset Only (via register write),	Display only.
DMA Support		
Descriptor Queue Base Address	(Hex, 4 KiB aligned) 0000_0000 – FFFF_F000	Available if DMA support is enabled. This is a 4 KiB aligned base address of the descriptor queue. Parameter: DESC_QUEUE_BASEADR = {20’h00000 – 20’hFFFFF}
Descriptor Queue Depth	4 – 256	Available if DMA support is enabled. Set the descriptor queue size. Parameter: DESC_QUEUE_SIZE = {4,8,16,32,64,128,256}
Status Queue Base Address	(Hex, 4 KiB aligned) 0000_0000 – FFFF_F000	Available if DMA support is enabled. This is a 4 KiB aligned base address of the status queue. Parameter: STAT_QUEUE_BASEADR = {20’h00000 – 20’hFFFFF}

Attribute	Values	Description
Status Queue Depth	4 – 256	Available if DMA support is enabled. Set the status queue size. Parameter: STAT_QUEUE_SIZE = {4,8,16,32,64,128,256}
Rx TLP Destination Base Address		
Posted TLP Base Address	(Hex, 4 KiB aligned) 0000_0000 – FFFF_F000	Available if data interface selected is AHB-Lite and DMA support is not selected. This is a 4 KiB aligned base address used when sending the received posted TLP to AHB-Lite bus. Parameter: PCIE_LOC_ADDR_POSTED = {20'h00000 – 20'hFFFFF}
Non-Posted TLP Base Address	(Hex, 4 KiB aligned) 0000_0000 – FFFF_F000	Available if data interface selected is AHB-Lite and DMA support is not selected. This is a 4 KiB aligned base address used when sending the received non-posted TLP to AHB-Lite bus. Parameter: PCIE_LOC_ADDR_NONPOSTED = {20'h00000 – 20'hFFFFF}
Completion TLP Base Address	(Hex, 4 KiB aligned) 0000_0000 – FFFF_F000	Available if data interface selected is AHB-Lite and DMA support is not selected. This is a 4 KiB aligned base address used when sending the received completion TLP to AHB-Lite bus. Parameter: PCIE_LOC_ADDR_COMPLETION = {20'h00000 – 20'hFFFFF}
Optional Ports		
Enable CLKREQ# Port	Checked Unchecked	Set to add the clkreq_n_io port. Parameter: USE_CLKREQ_SIGNAL = {0,1}
Enable LTSSM disable Port	Checked Unchecked	Set to add the u_ltssm_disable_i port.
Enable PM LTR Ports	Checked Unchecked	Available if LTR capability is enabled. Set to add the Latency Tolerance Reporting ports.
Enable PM DPA Ports	Checked Unchecked	Available if DPA capability is enabled. Set to add the Dynamic Power Allocation ports.
Enable PM PB Ports	Checked Unchecked	Available if PB capability is enabled. Set to add the Power Budgeting Ports.
Enable Legacy Interrupt Ports	Checked Unchecked	Available if legacy interrupt is enabled. Set to add the Legacy interrupt ports. Parameter: PCIE_LL_MAIN_CTRL_4_EN_PORT_MGMT_INTERRUPT_LEG = {0,1}

Attribute	Values	Description
Flow Control		
Flow Control Update		
Disable FC Update Timer	Checked Unchecked	Set to disable FC Update Timer (i.e schedule a FC Update on Every Consumed RX TLP, otherwise schedule FC Updates in accordance with PCIe. Spec. recommended values) Parameter: MGMT_PTL_RX_CTRL_FC_UPDATE_TIMER_DISABLE = {0,1}
FC Update Timer Divider	Use PCIe Spec recommended values, Divide by 2, Divide by 4, Divide by 8	Select the FC Update frequency of the Receive Buffer when FC update timer is enabled. Parameter: MGMT_PTL_RX_CTRL_FC_UPDATE_TIMER_DIV = {0,1,2,3}
Completion Credit Advertisement	Advertise Infinite for Endpoint and Actual for Root Port, Advertise Actual, Advertise Infinite	Select the completion credit advertisement behavior. Parameter: MGMT_PTL_RX_CTRL_ADV_CH_CD_SEL = {0,1,2}
Receive Buffer Allocation		
Posted Header Credits	(20 bytes per credit) 1 – 16	Set the amount of buffer credits for Posted TLP header. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space (2KB). Parameter: MGMT_PTL_RX_ALLOC_P_H = {1 – 16}
Posted Data Credits	(16 bytes per credit) 16 – 108	Set the amount of buffer credits for Posted TLP data. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space (2KB). Parameter: MGMT_PTL_RX_ALLOC_P_D = {16 – 108}
Non-Posted Header Credits	(20 bytes per credit) 1 – 8	Set the amount of buffer credits for Non-Posted TLP header. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space (256 Bytes). Parameter: MGMT_PTL_RX_ALLOC_N_H = {1 – 8}
Non-Posted Data Credits	(16 bytes per credit) 2 – 6	Set the amount of buffer credits for Non-Posted TLP data. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space (256 Bytes). Parameter: MGMT_PTL_RX_ALLOC_N_D = {2 – 6}
Completion Header Credits	(16 bytes per credit) 1 – 32	Set the amount of buffer credits for Completion TLP header. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space (2KB). Parameter: MGMT_PTL_RX_ALLOC_C_H = {1 – 32}

Attribute	Values	Description
Completion Data Credits	(16 bytes per credit) 16 – 96	Set the amount of buffer credits for Completion TLP data. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space (2KB). Parameter: MGMT_PTL_RX_ALLOC_C_D = {16 – 96}
Transmit Buffer Allocation		
Posted Header Credits	(20 bytes per credit) 1 – 16	Set the amount of buffer credits for Posted TLP header. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space (2KB). Parameter: MGMT_PTL_TX_ALLOC_P_H = {1 – 16}
Posted Data Credits	(16 bytes per credit) 16 – 108	Set the amount of buffer credits for Posted TLP data. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space (2KB). Parameter: MGMT_PTL_TX_ALLOC_P_D = {16 – 108}
Non-Posted Header Credits	(20 bytes per credit) 1 – 8	Set the amount of buffer credits for Non-Posted TLP header. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space (256 Bytes). Parameter: MGMT_PTL_TX_ALLOC_N_H = {1 – 8}
Non-Posted Data Credits	(16 bytes per credit) 2 – 6	Set the amount of buffer credits for Non-Posted TLP data. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space (256 Bytes). Parameter: MGMT_PTL_TX_ALLOC_N_D = {2 – 6}
Completion Header Credits	(16 bytes per credit) 1 – 32	Set the amount of buffer credits for Completion TLP header. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space (2KB). Parameter: MGMT_PTL_TX_ALLOC_C_H = {1 – 32}
Completion Data Credits	(16 bytes per credit) 16 – 96	Set the amount of buffer credits for Completion TLP data. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space (2KB). Parameter: MGMT_PTL_TX_ALLOC_C_D = {16 – 96}
Function 0		
Configuration		
Disable Function	Unchecked	Cannot disable function 0. Display only.
Device ID	(Hex) 0000 – FFFF	Value returned when the Device ID Configuration Register is read. Parameter: MGMT_FTL_ID1_DEVICE_ID = {16’h0000 – 16’hFFFF}

Attribute	Values	Description
Vendor ID	(Hex) 0000 – FFFF	Value returned when the Vendor ID Configuration Register is read. Parameter: MGMT_FTL_ID1_VENDOR_ID = {16'h0000 – 16'hFFFF}
Subsystem ID	(Hex) 0000 – FFFF	Value returned when the Subsystem ID Configuration Register is read. Parameter: MGMT_FTL_ID2_SUBSYSTEM_ID = {16'h0000 – 16'hFFFF}
Subsystem Vendor ID	(Hex) 0000 – FFFF	Value returned when the Subsystem Vendor ID Configuration Register is read. Parameter: MGMT_FTL_ID2_SUBSYSTEM_VENDOR_ID = {16'h0000 – 16'hFFFF}
Class Code	(Hex) 00000 – FFFFF	Value returned when the Class Code Configuration Register is read. Parameter: MGMT_FTL_ID3_CLASS_CODE = {16'h0000 – 16'hFFFF}
Revision ID	(Hex) 00 – FF	Value returned when the Revision ID Configuration Register is read. Parameter: MGMT_FTL_ID3_REVISION_ID = {8'h00 – 8'hFF}
Resizable BAR Capability		
Enable Resizable BAR Capability	Checked, Unchecked	Set to enable the Resizable BAR Capability Parameter: LINK0_FTL_RBAR_CAP_ENABLE = {0,1}
Base Address Register n (n == 0 – 5)		
BAR n : Enable	Checked, Unchecked	Set to enable the BAR.
BAR n : Resizable	Checked, Unchecked	Set to make this BAR resizable.
BAR n : Address Type	Memory, I/O	Select if the BAR is for Memory or I/O space.
BAR n : 64 bit Address	Checked, Unchecked	Applicable for memory space only. Set to use 64-bit address. Note that BAR n and BAR n+1 are used for the 64 bit address.
BAR n : Prefetchable	Checked, Unchecked	Applicable for memory space only. Set to identify the memory address as prefetchable.
BAR n : Resizable BAR Supported Sizes [23:4]	(Hex) 00000 – FFFFF	Each bit indicates a supported size, which is 2(i+16) bytes, where i is the index from [23:4]. For example, if bit[4] == 1, then 2(4+16) Bytes = 1 MB.
BAR n : Default Size (unit)	Bytes, KiB (2^{10}), MiB (2^{20}), GiB (2^{30}), TiB (2^{40}), PiB (2^{50}), EiB (2^{60}),	Select the size of Memory space.

Attribute	Values	Description
BAR n : Default Size (value)	(power of 2) 32 bits Memory Space: 16 bytes – 2 GiB 64 bits Memory Space: 64 bits: 4 GiB – 8 EiB 32 bits IO Space: 2 Bytes – 256 Bytes	Select the size of Memory or IO space.
BAR n	32 bits: FFFF_FFF0 – 1000_0000 64 bits: FFFF_FFFF_0000_0000 – 1000_0000_0000_0000	Display only. Parameter: Function 0: MGMT_FTL_BAR0_CFG … … … MGMT_FTL_BAR5_CFG Function m: MGMT_FTL_MF1_BAR0_CFG … … … MGMT_FTL_MF[m]_BAR[n]_CFG
Local Memory Base Address n	(Hex, Aligned to BAR size) FFFF_FFF0 – 0000_0000	Applicable for memory space only. This is the base address of the local system memory that maps to the configured PCIe BAR. It must be aligned to the specified BAR size. Received Memory requests that hits the BAR are forwarded to this address. Parameter: Function 0: F0BAR0_TO_LOCADR … … … F0BAR5_TO_LOCADR Function m: F1BAR0_TO_LOCADR … … … F[m]BAR[n]_TO_LOCADR
Legacy Interrupt		
Disable Legacy Interrupt	Checked, Unchecked	Set to disable the legacy interrupt support. Parameter: MGMT_FTL_INTERRUPT_DISABLE = {0,1}

Attribute	Values	Description
Interrupt Pin	INT A, INT B, INT C, INT D	Select which legacy interrupt pin is used. Parameter: MGMT_FTL_INTERRUPT_PIN = {0,1,2,3}
MSI Capability		
Disable MSI Capability	Checked, Unchecked	Set to disable the MSI Capability. Parameter: MGMT_FTL_MSI_CAP_DISABLE = {0,1}
Number of MSI vectors	1 – 32	Set the number of requested MSI vectors. Parameters: MGMT_FTL_MSI_CAP_MULT_MESSAGE_CAPABLE = {0,1,2,3,4,5}
Enable Vector Masking	Checked, Unchecked	Set to enable vector masking capability. Parameter: MGMT_FTL_MSI_CAP_VEC_MASK_CAPABLE = {0,1}
MSI-X Capability		
Disable MSI-X Capability	Checked, Unchecked	Set to disable the MSI-X Capability. Parameter: MGMT_FTL_MSIX_CAP_DISABLE = {0,1}
MSI-X Table Size	1 – 2048	Set the number of requesterd MSI-X vectors. Parameter: MGMT_FTL_MSIX_CAP_TABLE_SIZE = {0 – 2047}
MSI-X Table BAR indicator	BAR 0, BAR 1, BAR 2, BAR 3, BAR 4, BAR 5	Select which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X Table into Memory Space. Parameter: MGMT_FTL_MSIX_TABLE_BIR = {0,1,2,3,4,5}
MSI-X Table Address Offset	(Hex, 8 bytes aligned) 0000_0000 – FFFF_FFF8	Set the byte address offset (8 bytes aligned), within the BAR selected by MSI-X Table BAR indicator, at which the MSI-X Table begins. Parameter: MGMT_FTL_MSIX_TABLE_OFFSET = {29'h00000000 – 29'h1FFFFFFF}
MSI-X PBA BAR indicator	BAR 0, BAR 1, BAR 2, BAR 3, BAR 4, BAR 5	Select which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X PBA into Memory Space. Parameter: MGMT_FTL_MSIX_PBA_BIR = {0,1,2,3,4,5}
MSI-X PBA Address Offset	(Hex, 8 bytes aligned) 0000_0000 – FFFF_FFF8	Set the byte address offset (8 bytes aligned), within the BAR selected by MSI-X PBA BAR indicator, at which the MSI-X PBA begins. Parameter: MGMT_FTL_MSIX_PBA_OFFSET = {29'h00000000 – 29'h1FFFFFFF}

Attribute	Values	Description
Device Serial Number Capability		
Enable DSN Capability	Checked, Unchecked	Set to enable the Device Serial Number capability. Parameter: MGMT_FTL_DSN_CAP_ENABLE = {0,1}
Serial Number	(Hex) 0000_0000_0000_0000 – FFFF_FFFF_FFFF_FFFF	Set the device serial number. Parameter: MGMT_FTL_DSN_SERIAL_NUMBER
PCIe Device Capability		
Maximum Payload Size Supported	128 Bytes, 256 Bytes, 512 Bytes	Select the maximum payload size supported. Parameter: MGMT_FTL_PCIE_DEV_CAP_MAX_PAYLOAD_SIZE_SUPPORTED = {0,1,2}
Disable Function Level Reset(FLR)	Checked, Unchecked	Set to disable Function Level Reset capability. Parameter: MGMT_FTL_PCIE_DEV_CAP_DISABLE_FLR_CAPABILITY = {0,1)
Enable Extended Tag Field	Checked, Unchecked	Set to enable Extended Tag Field (8-bit tag field). Parameter: MGMT_FTL_PCIE_DEV_CAP_EXTENDED_TAG_FIELD_SUPPORTED = {0,1}
Advance Error Reporting Capability		
Enable ECRC Generation and Checking	Checked, Unchecked	Set to enable ECRC generation and checking. Parameter: MGMT_FTL_AER_CAP_ECRC_GEN_CHK_CAPABLE = {0,1}
Enable Reporting : Correctable Internal Error	Checked, Unchecked	Set to enable reporting of correctable internal error. Parameter: MGMT_FTL_AER_CAP_EN_CORR_INTERNAL_ERROR = {0,1}
Enable Reporting : Surprise Down Error	Checked, Unchecked	Set to enable reporting of surprise down error. Parameter: MGMT_FTL_AER_CAP_EN_SURPRISE_DOWN_ERROR = {0,1}
Enable Reporting : Completion Timeout Error	Checked, Unchecked	Set to enable reporting of completion timeout error. Parameter: MGMT_FTL_AER_CAP_EN_COMPLETION_TIMEOUT = {0,1}
Enable Reporting : Completer Abort Error	Checked, Unchecked	Set to enable reporting of completer abort error. Parameter: MGMT_FTL_AER_CAP_EN_COMPLETER_ABORT = {0,1}
Enable Reporting : Uncorrectable Internal Error	Checked, Unchecked	Set to enable reporting of uncorrectable internal error. Parameter: MGMT_FTL_AER_CAP_EN_UCORR_INTERNAL_ERROR = {0,1}

Attribute	Values	Description
ATS Capability		
Enable ATS Capability	Checked, Unchecked	Set to enable the ATS Capability.
Atomic OP Capability		
Enable Atomic Op Capability	Checked, Unchecked	Set to enable Atomic Operations Capability.
Enable Root as Atomic Op Completer	Checked, Unchecked	Set to enable Root as Atomic OP Completer.
Enable Atomic Op Completer 128b Operand	Checked, Unchecked	Set to support Atomic Op 128b operand.
Enable Atomic Op Completer 64b Operand	Checked, Unchecked	Set to support Atomic Op 64b operand.
Enable Atomic Op Completer 32b Operand	Checked, Unchecked	Set to support Atomic Op 32b operand.
Enable Atomic Op Completer Routing	Checked, Unchecked	Set to support Atomic Op routing.
Latency Tolerance Reporting Capability		
Enable LTR Capability	Checked, Unchecked	Set to enable the Latency Tolerance Reporting capability. Parameter: MGMT_FTL_LTR_CAP_ENABLE = {0,1}
Dynamic Power Allocation Capability		
Enable DPA Capability	Checked, Unchecked	Set to enable the Dynamic Power Allocation capability. Parameter: MGMT_FTL_DPA_CAP_ENABLE = {0,1}
Max Substate Number	0–31	Specifies the maximum substate number. Substates from [substate_max:0] are supported. For example, substate_max==0 indicates support for 1 substate.
Transition Latency Unit	1 ms, 10 ms, 100 ms	Specifies Transition Latency Unit.
Power Allocation Scale	10.0x, 1.0x, 0.1x, 0.01x	Specifies Power Allocation Scale.
Transition Latency Value 0	0–255	Specifies Transition Latency Value 0.
Transition Latency Value 1	0–255	Specifies Transition Latency Value 1.
Transition Latency Indicator 32x1b	(Hex) 00000000 – FFFFFFFF	Specifies which Transition Latency Value applies to each substate. Each bit corresponds to a substate.
Power Allocation Array 32x8b	(Hex) {32{00}} – {32{FF}}	Substate Power Allocation Array. Each entry is 8b value.
Power Budgeting Capability		
Enable PB Capability	Checked, Unchecked	Set to enable the Power Budgeting capability. Parameter: MGMT_FTL_PWR_BUDGET_CAP_ENABLE = {0,1}
Function n (n == 1 – 3)		
Configuration		

Disable Function	Checked, Unchecked	Available if the number of physical functions enabled is set to greater than 1.
		Set to disable the function.
		
		Parameter:
		MGMT_FTL_MF1_FUNCTION_DISABLE = {0,1}
		MGMT_FTL_MF2_FUNCTION_DISABLE = {0,1}
		MGMT_FTL_MF3_FUNCTION_DISABLE = {0,1}

Attribute	Values	Description
Device ID	See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab)	
Vendor ID		
Subsystem ID		
Subsystem Vendor ID		
Class Code		
Revision ID		
Base Address Register (See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab))		
Legacy Interrupt (See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab))		
MSI Capability (See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab))		
MSI-X Capability (See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab))		
Device Serial Number Capability (See Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab))		

2.4. Register Description

2.4.1. Hard IP Core Configuration and Status Registers

The Lattice PCIe X1 Core configuration registers have default values that are appropriate for most applications. Customers typically would only want to change a small number of values such as Vendor/Device ID and BAR configuration. Such changes can be made via LMMI writes prior to core reset release or through the IP generation user interface.

2.4.1.1. EP/RP Configuration Settings

The Lattice PCIe X1 Core supports Endpoint (EP) and Root Port (RP) operation. The current mode of operation is determined by the core's CSR. The EP/RP mode of operation is primarily determined by mgmt_tlb_ltssm_port_type_ds_us_n and mgmt_ftl_cfg_type1_type0_n, however, several additional registers typically also need to be changed when switching between modes.

Table 2.29 illustrates the CSR values that are recommended for EP and RP applications:

Table 2.29. CSR Values V for EP and RP Applications

Register Field	Offset	EndPoint	Root Port
mgmt_tlb_*			
mgmt_tlb_ltssm_port_type_ds_us_n	0x2040	1'b0	1'b1
mgmt_ftl_*			
mgmt_ftl_cfg_type1_type0_n	0x4030	1'b0	1'b1
mgmt_ftl_decode_ignore_poison	0x4010	1'b0	1'b1
mgmt_ftl_decode_t1_rx_bypass_msg_dec	0x4014	1'b0	1'b1
mgmt_ftl_pcie_cap_slot_implemented	0x4080	1'b0	1'b1
mgmt_ftl_pcie_cap_device_port_type	0x4080	4'h0	4'h4
mgmt_ftl_id3_class_code	0x4048	User Application Specific	24'h060000
mgmt_ftl_ari_cap_disable	0x40E0	1'b0	1'b1
mgmt_ftl_msi_cap_disable	0x40E8	1'b0	1'b1
mgmt_ftl_msi_cap_mult_message_capable	0x40E8	User Application Specific	4'h0
mgmt_ftl_msix_cap_table_size	0x40F0	User Application Specific	11'h0
mgmt_ftl_msix_cap_disable	0x40F0	1'b0 (Enabled)	1'b1 (Disabled)
mgmt_ftl_aer_cap_en_surprise_down_error	0x4100	1'b0	1'b1

2.4.1.2. mgmt_tlb (0x02000)

mgmt_tlb_BASE 0x2000

2.4.1.2.1. ltssm_simulation Register 0x0

LTSSM simulation speed reduction.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1]	reduce_ts1	read-write	1	0x0
[0]	reduce_timeouts	read-write	1	0x0

reduce_ts1

Reduce the minimum number of TS1 transmitted in Polling.Active from 1024 to 16 to shorten simulation time.

- 0 – Disable
- 1 – Enable

reduce_timeouts

Reduce LTSSM timeouts to shorten simulation time. When enabled, 1ms->20us, 2ms->40us, 12ms->60us, 24ms->80us, 32ms->100us, and 48ms->160us.

- 0 – Disable
- 1 – Enable

2.4.1.2.2. ltssm_cfg_lw_start Register 0x34

LTSSM CFG.LWSTART configuration.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1:0]	min_time	read-write	2	0x0

min_time

Minimum time spent in Cfg.LW.Start before exit is permitted.

- 0 – 4 us
- 1 – 16 us
- 2 – 64u us
- 3 – 256 us

2.4.1.2.3. ltssm_latch_rx Register 0x38

LTSSM latch RX configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	link_lane	read-write	1	0x1

link_lane

Enable latching each lane's received link and lane numbers and state exit condition during LTSSM Configuration link width negotiation.

- 0 – Disable. A lane is included in the link if it is receiving the state exit criteria on the clock cycle that the link width and state exit transition is occuring. A received Physical Layer error occuring close to the clock cycle that the link width is being determined results in a reduction of link width even if the lane had previously recorded valid state exit criteria.
- 1 – Enable. A lane is included in the link if it met the state exit criteria at any time during the state. This is the recommended setting since received Physical Layer errors are less likely to result in reduced link width.

2.4.1.2.4. ltssm_cfg Register 0x3c

LTSSM Configuration configuration.

Field	Name	Access	Width	Reset
[31:28]	lw_start_updn_end_delay	read-write	4	0x9
[27:24]	lw_start_updn_start_delay	read-write	4	0x8
[23:12]	lw_start_updn_count	read-write	12	0xfa
[11:8]	lw_start_updn_rate_en	read-write	4	0xf
[7:6]	*reserved*	read-only	2	0x0
[5]	lw_start_updn_eie_en	read-write	1	0x0
[4]	lw_start_updn_en_dir_ds	read-write	1	0x0
[3:2]	*reserved*	read-only	2	0x0
[1]	lw_start_updn_timer_en	read-write	1	0x0
[0]	lw_start_updn_ack_en	read-write	1	0x0

lw_start_updn_end_delay

LTSSM CFG_[US/DS]_LW_START normal CFG_[US/DS]_LW_START TS1 transmissions and parsing of received TS OS begins (lw_start_updn_end_delay * 64) symbols after the bp_ltssm_cfg_lw_start_updn 1 to 0 transition occurs at the end of PHY adaptation. This delay is intended to flush any corrupted PHY rx data due to the PHY adaptation through the Link Layer Core before the Core begins paying attention to received data again.

lw_start_updn_start_delay

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn==1 assertion is delayed by (lw_start_updn_start_delay * 64) symbols from CFG_[US/DS]_LW_START state entry. The start delay is intended to avoid the PHY beginning adaptation, and thus corrupting the input data, before the link partner data stream has ended. When the Core reaches CFG_[US/DS]_LW_START before the link partner, the link partner may still be in Recovery.Idle with an active data stream. The start delay must be long enough to delay PHY adaptation until the receive data stream has ended or else SKP Data Parity Errors and Receiver Errors can be detected and recorded by the Core due to the PHY corrupting the receive data stream due to adaptation.

lw_start_updn_count

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn==1 duration is set to (lw_start_updn_count * 1024) ns. 0==Disabled.

lw_start_updn_rate_en

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn==1 rate enable/disable. Controls for which speeds the bp_ltssm_cfg_lw_start_updn feature is supported. One bit is provided to enable/disable each speed supported {5G, 2.5G}. Bit positions for speeds that are not supported by a given core delivery must be set to 0.

- 0 – Disable feature when at the associated link speed.
- 1 – Enable feature when at the associated link speed.

lw_start_updn_eie_en

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn==1 EIE Tx OS enable.

- 0 – Disabled.
- 1 – Enabled.

lw_start_updn_en_dir_ds

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn==1 directed down-configure enable.

- 0 – Do not assert bp_ltssm_cfg_lw_start_updn==1 when the CFG_[US/DS]_LW_START entry was due to locally directed downconfigure.
- 1 – Assert bp_ltssm_cfg_lw_start_updn==1 when the CFG_[US/DS]_LW_START entry was due to locally directed downconfigure.

lw_start_updn_timer_en

LTSSM CFG_[US/DS]_LW_START bp_ltssm_cfg_lw_start_updn Timer Enable. Register lw_start_updn_timer_en can be set to stay in adaptation for a fixed time period instead of relying on the PHY to have a port bp_ltssm_cfg_lw_start_updn_ack that is asserted at the end of adaptation. Only one of lw_start_updn_timer_en and lw_start_updn_ack_en can be set to 1.

- 0 – Disabled.
- 1 – Deassert bp_ltssm_cfg_lw_start_updn after (lw_start_updn_count * 1024) ns has expired.

lw_start_updn_ack_en

LTSSM Configuration Link Width Start bp_ltssm_cfg_lw_start_updn Ack Enable.

Some PHY require special processing to support up/downconfiguration. For such PHY, in order to trigger the PHY to perform the necessary processing, output port bp_ltssm_cfg_lw_start_updn is set to 1 and held at 1 when the LTSSM enters the CFG_[DS/US]_LW_START state from Recovery – which is the LTSSM transition for which up/downconfiguration can occur. After the PHY has performed the necessary processing, the PHY must set input port bp_ltssm_cfg_lw_start_updn_ack == 1 for one clock to indicate that its processing has been completed. bp_ltssm_cfg_lw_start_updn_ack must be 0 at all other times. bp_ltssm_cfg_lw_start_updn is set back to 0 when bp_ltssm_cfg_lw_start_updn_ack == 1 or if the CFG_[DS/US]_LW_START state is exited. The CFG_[DS/US]_LW_START state can be exited before the PHY asserts bp_ltssm_cfg_lw_start_updn_ack due to the 24 mS state timeout, due to a transition to the Disable or Loopback states, or due to a Crosslink state transition between CFG_DS_LW_START and CFG_US_LW_START. While bp_ltssm_cfg_lw_start_updn == 1, the LTSSM does not make the state transition from CFG_[DS/US]_LW_START to CFG_[DS/US]_LW_ACCEPT. Once bp_ltssm_cfg_lw_start_updn is deasserted (due to receiving bp_ltssm_cfg_lw_start_updn_ack == 1), an additional minimum 4 uS wait time is enforced to allow enough time for all lanes to sync to their data streams and output valid TS set data, before the transition from CFG_[DS/US]_LW_START to CFG_[DS/US]_LW_ACCEPT is allowed.

Use of ports bp_ltssm_cfg_lw_start_updn and bp_ltssm_cfg_lw_start_updn_ack is optional.

mgmt_tlb_ltssm_cfg_lw_start_updn_ack_en and mgmt_tlb_ltssm_cfg_lw_start_updn_rate_en control whether the bp_ltssm_cfg_lw_start_updn/_ack ports are implemented by the LTSSM.

A 4 uS minimum state exit time is enforced whenever CFG_[DS/US]_LW_START is entered from Recovery independent of the value of mgmt_tlb_ltssm_cfg_lw_start_updn_ack_en and mgmt_tlb_ltssm_cfg_lw_start_updn_rate_en. The 4 uS minimum wait is enforced to allow enough time for all lanes to sync to their data streams and output valid TS set data, before the transition from CFG_[DS/US]_LW_START to CFG_[DS/US]_LW_ACCEPT is allowed.

mgmt_tlb_ltssm_cfg_lw_start_updn_rate_en determines for which speeds output/input ports bp_ltssm_cfg_lw_start_updn/_ack are supported. Output/inputs ports bp_ltssm_cfg_lw_start_updn/_ack are enabled at a given speed when both mgmt_tlb_ltssm_cfg_lw_start_updn_ack_en == 1 and the associated mgmt_tlb_ltssm_cfg_lw_start_updn_rate_en[i] == 1.

mgmt_tlb_ltssm_cfg_lw_start_updn_ack_en does not impact the other LTSSM transitions from CFG_[DS/US]_LW_START which include transitions to Detect (state timeout), Disable, Loopback, and Crosslink transitions between CFG_DS_LW_START and CFG_US_LW_START.

- 0 – Disabled. Output port bp_ltssm_cfg_lw_start_updn is held == 0 and input port bp_ltssm_cfg_lw_start_updn_ack is ignored. When CFG_[DS/US]_LW_START is entered from Recovery, the transition from CFG_[DS/US]_LW_START to CFG_[DS/US]_LW_ACCEPT occurs after a minimum of 4 uS.
- 1 – Enabled. If also enabled, via mgmt_tlb_ltssm_cfg_lw_start_updn_rate_en, at the current link speed, output port bp_ltssm_cfg_lw_start_updn is set upon CFG_[DS/US]_LW_START entry from Recovery and input port bp_ltssm_cfg_lw_start_updn_ack is used. The transition from CFG_[DS/US]_LW_START to CFG_[DS/US]_LW_ACCEPT occurs only after the PHY has asserted bp_ltssm_cfg_lw_start_updn_ack == 1 and additionally a minimum of 4 uS has elapsed. bp_ltssm_cfg_lw_start_updn_ack must not be withheld so long that the state timeout of 24 mS expires or the link exits to Detect and the link goes down – which is a serious error.

2.4.1.2.5. ltssm_port_type Register 0x40

LTSSM port type.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	ds_us_n	read-write	1	0x0

ds_us_n

Determines the PCI Express port type which affects many aspects of LTSSM training.

- 0 – Upstream Port
- 1 – Downstream Port

2.4.1.2.6. ltssm_ds_link Register 0x44

LTSSM downstream link configuration.

Field	Name	Access	Width	Reset
[31:5]	*reserved*	read-only	27	0x0
[4:0]	number	read-write	5	0x0

number

For downstream ports only, unique Link Number assigned to the link and used in TS sets during LTSSM Configuration.

2.4.1.2.7. ltssm_detect_quiet Register 0x48

LTSSM Detect.Quiet configuration.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1:0]	min_time	read-write	2	0x0

min_time

Minimum time spent in Detect.Quiet before an exit is permitted.

- 0 – 0 mS
- 1 – 1 mS
- 2 – 2 mS
- 3 – 12 mS

2.4.1.2.8. ltssm_rx_det Register 0x4c

LTSSM receiver detection configuration.

Field	Name	Access	Width	Reset
[31]	override	read-write	1	0x0
[30:16]	*reserved*	read-only	15	0x0
[15:0]	mask	read-write	16	0x0

override

Lane receiver detection mask enable.

- 0 – Disable
- 1 – Enable

mask

Lane receiver detection mask. When override==1, mask determines which lanes attempt receiver detection. For each lane[i]:

- 0 – Skip receiver detection and exclude the lane from the link.
- 1 – Perform receiver detection and use result to determine whether to include/exclude the lane from the link.

2.4.1.2.9. ltssm_nfts Register 0x50

LTSSM NFTS configuration.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15:8]	to_extend	read-write	8	0x7f
[7:0]	nfts	read-write	8	0xff

to_extend

Number of FTS set transfer times to wait in addition to the time required to transmit the requested NFTS sets before timing out to Recovery on Rx_L0s exit.

nfts

Number of FTS sets to request link partner transmit when exiting L0s. NFTS value transmitted in TS1 and TS2 Ordered Sets during training.

2.4.1.2.10. ltssm_ds_initial_auto Register 0x54

LTSSM initial link speed configuration.

Field	Name	Access	Width	Reset
[31]	rate_enable	read-write	1	0x0
[30:2]	*reserved*	read-only	29	0x0
[1:0]	rate	read-write	2	0x0

rate_enable

Determines whether link speed up is requested by the core after the first entry to L0 following state Detect. If neither port directs the link to a higher speed then the link remains at 2.5G unless software initiates a speed change. It is recommended to set rate_enable=1 and rate==maximum supported speed.

- 0 – Let the link partner or software initiate initial speed changes.
- 1 – Make 1 attempt to direct the link to the maximum speed specified by rate. The speed achieved is the maximum speed, less than or equal to rate, that both the core and link partner support.

rate

When rate_enable==1, indicates the maximum rate that is attempted to negotiate on the initial link training from Detect. Only speeds supported by the core can be indicated.

- 0 – 2.5G
- 1 – 5G

2.4.1.2.11. ltssm_select_deemphasis Register 0x58

LTSSM 2.5/5G deemphasis configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	6db_3_5db_n	read-write	1	0x1

6db_3_5db_n

For 5G capable cores only: For upstream ports only, sets the default deemphasis for 5G operation during LTSSM State Detect.

- 0 – -3.5dB
- 1 – -6dB

2.4.1.2.12. ltssm_beacon Register 0x5c

LTSSM Beacon configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	l2_d3hot_enable	read-write	1	0x0

l2_d3hot_enable

L2 wake Beacon transmission control.

- 0 – Disabled. The customer design must wake the link via WAKE# pin assertion. Set to 0 when using PHY which do not support Beacon transmission. Set to 0 if the core is not clocked (some PHY remove the core's clock in L2 while others supply a keep alive clock) or powered (some applications remove core power in L2 to maximize power savings) in L2, as the core is unable to initiate Beacon generation in these cases.
- 1 – Transmit beacon when directed to wake the link from L2.

2.4.1.2.13. ltssm_mod_cpl Register 0x60

LTSSM Modified Compliance.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1]	one_eieos	read-write	1	0x1
[0]	exit_direct_to_detect	read-write	1	0x0

one_eieos

When entering Modified Compliance Pattern determines the number of EIEOS blocks to send.

- 0 – Send 8 EIEOS blocks to ensure receiver lock
- 1 – Send 1 EIEOS (as per Spec)

exit_direct_to_detect

When transmitting Modified Compliance Pattern and cfg_enter_compliance == 0, determines which of the two PCIe Spec. optional behaviors is selected.

- 0 – Don't exit to Detect for this reason.
- 1 – Exit to Detect.

2.4.1.2.14. ltssm_rx_elec_idle Register 0x64

LTSSM Rx Electrical Idle configuration.

Field	Name	Access	Width	Reset
[31]	rec_spd_infer_rcvr_lock	read-write	1	0x0
[30]	rec_spd_infer_rcvr_cfg	read-write	1	0x0
[28:4]	*reserved*	read-only	25	0x0
[3:0]	filter	read-write	4	0x1

rec_spd_infer_rcvr_lock

Recovery Speed successful and unsuccesful inference expand to Recovery.RcvrLock enable.

- 0 – Do not include time spent in Recovery.RcvrLock when calculating successful and unsuccessful speed change electrical idle inference in Recovery.Speed.
- 1 – Include time spent in Recovery.RcvrLock when calculating successful and unsuccessful speed change electrical idle inference in Recovery.Speed.

rec_spd_infer_rcvr_cfg

Recovery Speed successful and unsuccesful inference expand to Recovery.RcvrCfg enable.

- 0 – Do not include time spent in Recovery.RcvrCfg when calculating successful and unsuccessful speed change electrical idle inference in Recovery.Speed.
- 1 – Include time spent in Recovery.RcvrCfg when calculating successful and unsuccessful speed change electrical idle inference in Recovery.Speed.

filter

After entering a LTSSM state that monitors, pipe_rx_elec_idle for exit, ignore pipe_rx_elec_idle for 128 * filter) nanoseconds to enable tolerance for pipe_rx_elec_idle not latency matched with the associaetd pipe_rx_data.

2.4.1.2.15. ltssm_compliance_toggle Register 0x68

LTSSM Compliance Toggle.

Field	Name	Access	Width	Reset
[31:4]	*reserved*	read-only	28	0x0
[3:2]	max_speed	read-write	2	0x3
[1:0]	min_speed	read-write	2	0x0

max_speed

Maximum speed of compliance patterns that should be generated

- 0 – 2.5G
- 1 – 5G

min_speed

Minimum speed of compliance patterns that should be generated

- 0 – 2.5G
- 1 – 5G

2.4.1.2.16. ltssm_prevent_rx_ts_entry_to Register 0x6c

LTSSM State Rx TS Transition Prevention. The fields in this register are provided to disable LTSSM state transitions from occuring due to Rx of TS OS. Certain states such as Disable, Hot Reset, Loopback, and Polling.Compliance are entered after receiving a small number (typically 2) of consecutive TS OS with the appropriate Control Symbol bit set. There is a very low, but non-zero probability that transmission errors could cause two consective TS OS to falsely assert the critical control bits and cause a false state transition. Some of these states, such as Polling.Compliance and Loopback cannot be exited if falsely entered in this manner. The fields in this register are set to turn off LTSSM state transitions due to Rx TS OS to prevent the false state transitions from being possible due to bit errors. Directed (requested by software) state transitions are not affected by this register; only state transitions due to Rx TS OS are affected. State transitions must not be disabled via this mechanism if those state transtions are necessary for the customer's application. However it should be noted that the states for which disables are provided are used for test purposes or link disabling and would not typically be entered during normal link operation.

Field	Name	Access	Width	Reset
[31:4]	*reserved*	read-only	28	0x0
[3]	compliance	read-write	1	0x0
[2]	loopback	read-write	1	0x0
[1]	hot_reset	read-write	1	0x0
[0]	disable	read-write	1	0x0

compliance

LTSSM to Polling.Compliance Rx TS state transition disable.

- 0 – Enabled
- 1 – Disabled

loopback

LTSSM to Loopback Slave Rx TS state transition disable.

- 0 – Enabled
- 1 – Disabled

hot_reset

LTSSM to Hot Reset Rx TS state transition disable.

- 0 – Enabled
- 1 – Disabled

disable

LTSSM to Disable Rx TS state transition disable.

- 0 – Enabled
- 1 – Disabled

2.4.1.2.17. ltssm_link Register 0x80

Current Link Status

Field	Name	Access	Width	Reset
[31]	dl_link_up	read-only	1	0x0
[30]	pl_link_up	read-only	1	0x0
[29:16]	*reserved*	read-only	10	0x0
[15]	idle_infer_rec_rcvr_cfg	read-write, wr:oneToClear	1	0x0
[14]	idle_infer_loopback_slave	read-write, wr:oneToClear	1	0x0
[13]	idle_infer_rec_speed2_success	read-write, wr:oneToClear	1	0x0
[12]	idle_infer_rec_speed2_unsuccess	read-write, wr:oneToClear	1	0x0
[11]	idle_infer_l0_to_rec_rcvr_lock	read-write, wr:oneToClear	1	0x0
[10:9]	*reserved*	read-only	2	0x0
[8]	speed_change_fail	read-write, wr:oneToClear	1	0x0
[7:2]	*reserved*	read-only	6	0x0
[1:0]	speed	read-only	2	0x0

dl_link_up

Data Link Layer link up status.

- 0 – Down
- 1 – Up

pl_link_up

Physical Layer link up status.

- 0 – Down
- 1 – Up

idle_infer_rec_rcvr_cfg

Electrical Idle inference status in Recovery.RcvrCfg.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

idle_infer_loopback_slave

Electrical Idle inference status in Loopback.Active as a Loopback Slave.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

idle_infer_rec_speed2_success

Electrical Idle inference status in Recovery.Speed on a successful speed negotiation.

- Otherwise.
- Event occurred. Write 1 to clear.

idle_infer_rec_speed2_unsuccess

Electrical Idle inference status in Recovery.Speed on an unsuccessful speed negotiation.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

idle_infer_l0_to_rec_rcvr_lock

Electrical Idle inference status in L0 – event causes entry into Recovery.RcvrLock.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

speed_change_fail

Speed Change Failure error indicator.

- 0 – Otherwise.
- 1 – Speed change failure ocurred. Write 1 to clear.

speed

Current LTSSM Link Speed. Only link speeds supported by the core are indicated.

- 0 – 2.5G
- 1 – 5G

2.4.1.2.18. ltssm_ltssm Register 0x84

LTSSM State Machine State.

Field	Name	Access	Width	Reset
[31:20]	*reserved*	read-only	12	0x0
[19:16]	sub_state	read-only	4	0x1
[15:4]	*reserved*	read-only	12	0x0
[3:0]	state	read-only	4	0x0

[3:0] state read-only 4 0x0

sub_state

Current LTSSM Minor State. Encoding varies depending upon the Current LTSSM Major State.

0 DETECT_INACTIVE

1 DETECT_QUIET

2 DETECT_SPD_CHG0

3 DETECT_SPD_CHG1

4 DETECT_ACTIVE0

5 DETECT_ACTIVE1

6 DETECT_ACTIVE2

7 DETECT_P1_TO_P0

8 DETECT_P0_TO_P1_0

9 DETECT_P0_TO_P1_1

10 DETECT_P0_TO_P1_2

0 POLLING_INACTIVE

1 POLLING_ACTIVE_ENTRY

2 POLLING_ACTIVE

3 POLLING_CFG

4 POLLING_COMP

5 POLLING_COMP_ENTRY

6 POLLING_COMP_EIOS

7 POLLING_COMP_EIOS_ACK

8 POLLING_COMP_IDLE

0 CONFIGURATION_INACTIVE

1 CONFIGURATION_US_LW_START

2 CONFIGURATION_US_LW_ACCEPT

3 CONFIGURATION_US_LN_WAIT

4 CONFIGURATION_US_LN_ACCEPT

5 CONFIGURATION_DS_LW_START

6 CONFIGURATION_DS_LW_ACCEPT

7 CONFIGURATION_DS_LN_WAIT

8 CONFIGURATION_DS_LN_ACCEPT

9 CONFIGURATION_COMPLETE

10 CONFIGURATION_IDLE

0 L0_INACTIVE

1 L0_L0

2 L0_TX_EL_IDLE

3 L0_TX_IDLE_MIN

0 RECOVERY_INACTIVE

1 RECOVERY_RCVR_LOCK

2 RECOVERY_RCVR_CFG

3 RECOVERY_IDLE

4 RECOVERY_SPEED0

5 RECOVERY_SPEED1

6 RECOVERY_SPEED2

7 RECOVERY_SPEED3

0 DISABLED_INACTIVE

1 DISABLED_0

2 DISABLED_1

3 DISABLED_2

4 DISABLED_3

0 LOOPBACK_INACTIVE

1 LOOPBACK_ENTRY

2 LOOPBACK_ENTRY_EXIT

3 LOOPBACK_EIOS

4 LOOPBACK_EIOS_ACK

5 LOOPBACK_IDLE

6 LOOPBACK_ACTIVE

7 LOOPBACK_EXIT0

8 LOOPBACK_EXIT1

0 HOT_RESET_INACTIVE

1 HOT_RESET_HOT_RESET

2 HOT_RESET_MASTER_UP

3 HOT_RESET_MASTER_DOWN

0 TX_L0S_INACTIVE

1 TX_L0S_IDLE

2 TX_L0S_TO_L0

3 TX_L0S_FTS0

4 TX_L0S_FTS1

0 L1_INACTIVE

1 L1_IDLE

2 L1_SUBSTATE

3 L1_TO_L0

0 L2_INACTIVE

1 L2_IDLE

2 L2_TX_WAKE0

3 L2_TX_WAKE1

4 L2_EXIT

5 L2_SPEED

state

Current LTSSM Major State.

- 0 – DETECT
- 1 – POLLING
- 2 – CONFIGURATION
- 3 – L0
- 4 – RECOVERY
- 5 – DISABLED
- 6 – LOOPBACK
- 7 – HOT_RESET
- 8 – TX_L0S
- 9 – L1
- 10 – L2

2.4.1.2.19. ltssm_rx_l0s Register 0x88

Rx L0s State Machine State.

Field	Name	Access	Width	Reset
[31:3]	*reserved*	read-only	29	0x0
[2:0]	state	read-only	3	0x0

state

Current LTSSM RX L0s State.

- 0 – RX_L0S_L0
- 1 – RX_L0S_ENTRY
- 2 – RX_L0S_IDLE
- 3 – RX_L0S_FTS
- 4 – RX_L0S_REC

2.4.1.2.20. l0_to_rec Register 0x8c

Reasons that Recovery is entered from L0.

Field	Name	Access	Width	Reset
[31:15]	*reserved*	read-only	17	0x0
[14]	direct_to_detect_fast	read-write, wr:oneToClear	1	0x0
[13]	direct_to_recovery_ch_bond	read-write, wr:oneToClear	1	0x0
[12]	direct_to_loopback_entry	read-write, wr:oneToClear	1	0x0
[11]	directed_speed_change	read-write, wr:oneToClear	1	0x0
[10]	l0_to_rec_rcvr_lock_rx_ts12	read-write, wr:oneToClear	1	0x0
[9]	l0_to_rec_rcvr_lock_rx_8g_eie	read-write, wr:oneToClear	1	0x0
[8]	l0_to_rec_rcvr_lock_rx_infer	read-write, wr:oneToClear	1	0x0
[7]	direct_to_recovery_phy	read-write, wr:oneToClear	1	0x0
[6]	*reserved*	read-only	1	0x0
[5]	direct_to_recovery_replay	read-write, wr:oneToClear	1	0x0
[4]	direct_to_hot_reset	read-write, wr:oneToClear	1	0x0
[3]	direct_to_disable	read-write, wr:oneToClear	1	0x0
[2]	rx_l0s_direct_to_recovery	read-write, wr:oneToClear	1	0x0
[1]	autonomous_width_change	read-write, wr:oneToClear	1	0x0
[0]	directed_retrain_link	read-write, wr:oneToClear	1	0x0

direct_to_detect_fast

Recovery was entered from L0 due to assertion of mgmt_tlb_ltssm_direct_to_detect_fast.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_recovery_ch_bond

Recovery was entered from L0 due to more lane skew than the Channel Bond circuit can tolerate or was due to channel bond failing to occur within the expected timeout period.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_loopback_entry

Recovery was entered from L0 due to being directed into Master Loopback.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

directed_speed_change

Recovery was entered from L0 due to being directed to make a speed change. This includes the initial hardware-initiated speed change(s) which are made when first exiting Detect.Quiet to L0.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

l0_to_rec_rcvr_lock_rx_ts12

Recovery was entered from L0 due to receiving TS1 or TS2 ordered sets. The link partner is directing Recovery entry.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

l0_to_rec_rcvr_lock_rx_8g_eie

Recovery was entered from L0 due to receiving EIE ordered sets at >= 8G. The link partner is directing Recovery entry.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

l0_to_rec_rcvr_lock_rx_infer

Recovery was entered from L0 due to inferring Electrical Idle due to no SKP ordered set received in 128 uS.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_recovery_phy

Recovery was entered from L0 due to receiving a burst of ~1024 clock cycles of data containing PHY errors at 2.5G or 5G. This normally only occurs when the PHY has lost lock on one or more lanes.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_recovery_replay

Recovery was entered from L0 due to the original and three replay TLP transmissions failing to receive ACK DLLP acknowledgment.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_hot_reset

Recovery was entered from L0 due to being directed into Hot Reset (Secondary Bus Reset Register).

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

direct_to_disable

Recovery was entered from L0 due to being directed into Disable (Link Disable Register).

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

rx_l0s_direct_to_recovery

Recovery was entered from L0 due to failing to receive the complete Rx_L0S FTS exit sequence within the PCIe Spec. allowed timeout period.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

autonomous_width_change

Recovery was entered from L0 due to directed autonomous width change.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

directed_retrain_link

Recovery was entered from L0 due to directed retrain link (Retrain Link Register).

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

2.4.1.2.21. ltssm_rx_detect Register 0x90

Receiver detection status.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15:0]	lanes	read-only	16	0x0

lanes

Per lane receiver detection status. For each lane:

- 0 – Unconnected
- 1 – Present

2.4.1.2.22. ltssm_configured Register 0x94

Configured link status.

Field	Name	Access	Width	Reset
[31:25]	*reserved*	read-only	7	0x0
[24:16]	link_num	read-only	9	0x1ff
[15:0]	lanes	read-only	16	0x0

link_num

Link Number configured during LTSSM Training. link_num == 0x1FF on fundamental reset, changes to 0x1F7 (KPAD) when entering CFG_US_LW_START or CFG_DS_LW_START (the start of LTSSM Configuration), and then changes to the negotiated Link Number determined during LTSSM Configuration when the LTSSM changes from CFG_COMPLETE to CFG_IDLE. This field is provided for diagnostics.

lanes

Per lane configured link status. Each lane status resets to 0. After Receiver Detection results are available, each lane status is updated to show which lanes detected receivers. After a link has been formed, each lane status is updated to show which lanes are part of the configured link. For each lane:

- 0 – Lane did not configure into the link.
- 1 – Lane configured into the link.

2.4.1.2.23. ltssm_direct_to_detect Register 0x98

Rec Rcvr Lock to Detect controls

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15]	fast	read-write	1	0x0
[14:8]	*reserved*	read-only	7	0x0
[7:0]	timer	read-write	8	0x0

fast

A rising edge on this signal instructs the state machine to proceed from L0 or Recovery to Detect as quickly as possible.

timer

This value determines the timeout delay for the state machine to proceed from Recovery Rcvr Lock to Detect when no TS sets are received. A value of 0 disables this timeout.

2.4.1.2.24. ltssm_equalization Register 0x9c

For >= 8G capable cores only: LTSSM equalization status.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	fail	read-only	1	0x0

fail

Equalization Failure error indicator.

- 0 – Otherwise.
- 1 – Equalization failure.

2.4.1.2.25. ltssm_crosslink Register 0xa0

LTSSM crosslink status.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1]	ds_us_n	read-only	1	0x0
[0]	active	read-only	1	0x0

ds_us_n

Crosslink port type. When active==1, indicates which personality the port assumed during crosslink negotiation.

- 0 – Upstream
- 1 – Downstream

active

Crosslink active indicator.

- 0 – Otherwise.
- 1 – Link is operating in a crosslink configuration.

2.4.1.2.26. Physical Layer Status

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	err_tx_pipe_underflow	read-write, wr:oneToClear	1	0x0

err_tx_pipe_underflow

Physical Layer Tx Underflow Error Status.

- 0 – O therwise.
- 1 – Physical Layer Tx data needed to be forwarded to the lanes for transmission and some, but not all lanes, were ready to accept data causing some lanes to underlow. This bit stays asserted once set. Write 1 to clear.

2.4.1.2.27. pl_rx0 Register 0xa8

Lane Rx Status 0.

Field	Name	Access	Width	Reset
[31]	ts2_detect15	read-write, wr:oneToClear	1	0x0
[30]	ts2_detect14	read-write, wr:oneToClear	1	0x0
[29]	ts2_detect13	read-write, wr:oneToClear	1	0x0
[28]	ts2_detect12	read-write, wr:oneToClear	1	0x0
[27]	ts2_detect11	read-write, wr:oneToClear	1	0x0
[26]	ts2_detect10	read-write, wr:oneToClear	1	0x0
[25]	ts2_detect9	read-write, wr:oneToClear	1	0x0
[24]	ts2_detect8	read-write, wr:oneToClear	1	0x0
[23]	ts2_detect7	read-write, wr:oneToClear	1	0x0
[22]	ts2_detect6	read-write, wr:oneToClear	1	0x0
[21]	ts2_detect5	read-write, wr:oneToClear	1	0x0
[20]	ts2_detect4	read-write, wr:oneToClear	1	0x0
[19]	ts2_detect3	read-write, wr:oneToClear	1	0x0
[18]	ts2_detect2	read-write, wr:oneToClear	1	0x0
[17]	ts2_detect1	read-write, wr:oneToClear	1	0x0
[16]	ts2_detect0	read-write, wr:oneToClear	1	0x0
[15]	ts1_detect15	read-write, wr:oneToClear	1	0x0
[14]	ts1_detect14	read-write, wr:oneToClear	1	0x0
[13]	ts1_detect13	read-write, wr:oneToClear	1	0x0
[12]	ts1_detect12	read-write, wr:oneToClear	1	0x0
[11]	ts1_detect11	read-write, wr:oneToClear	1	0x0
[10]	ts1_detect10	read-write, wr:oneToClear	1	0x0
[9]	ts1_detect9	read-write, wr:oneToClear	1	0x0
[8]	ts1_detect8	read-write, wr:oneToClear	1	0x0
[7]	ts1_detect7	read-write, wr:oneToClear	1	0x0
[6]	ts1_detect6	read-write, wr:oneToClear	1	0x0
[5]	ts1_detect5	read-write, wr:oneToClear	1	0x0
[4]	ts1_detect4	read-write, wr:oneToClear	1	0x0
[3]	ts1_detect3	read-write, wr:oneToClear	1	0x0
[2]	ts1_detect2	read-write, wr:oneToClear	1	0x0
[1]	ts1_detect1	read-write, wr:oneToClear	1	0x0
[0]	ts1_detect0	read-write, wr:oneToClear	1	0x0

ts2_detect15

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect14

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect13

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect12

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect11

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect10

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect9

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect8

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect7

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect6

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect5

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect4

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect3

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect2

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect1

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2_detect0

ts2_detect[i] is set to 1 when a TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect15

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect14

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect13

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect12

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect11

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect10

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect9

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect8

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect7

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect6

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect5

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect4

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect3

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect2

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect1

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1_detect0

ts1_detect[i] is set to 1 when a TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

2.4.1.2.28. pl_rx1 Register 0xac

Lane Rx Status 1.

Field	Name	Access	Width	Reset
[31]	ts2i_detect15	read-write, wr:oneToClear	1	0x0
[30]	ts2i_detect14	read-write, wr:oneToClear	1	0x0
[29]	ts2i_detect13	read-write, wr:oneToClear	1	0x0
[28]	ts2i_detect12	read-write, wr:oneToClear	1	0x0
[27]	ts2i_detect11	read-write, wr:oneToClear	1	0x0
[26]	ts2i_detect10	read-write, wr:oneToClear	1	0x0
[25]	ts2i_detect9	read-write, wr:oneToClear	1	0x0
[24]	ts2i_detect8	read-write, wr:oneToClear	1	0x0
[23]	ts2i_detect7	read-write, wr:oneToClear	1	0x0
[22]	ts2i_detect6	read-write, wr:oneToClear	1	0x0
[21]	ts2i_detect5	read-write, wr:oneToClear	1	0x0
[20]	ts2i_detect4	read-write, wr:oneToClear	1	0x0
[19]	ts2i_detect3	read-write, wr:oneToClear	1	0x0
[18]	ts2i_detect2	read-write, wr:oneToClear	1	0x0
[17]	ts2i_detect1	read-write, wr:oneToClear	1	0x0
[16]	ts2i_detect0	read-write, wr:oneToClear	1	0x0
[15]	ts1i_detect15	read-write, wr:oneToClear	1	0x0
[14]	ts1i_detect14	read-write, wr:oneToClear	1	0x0
[13]	ts1i_detect13	read-write, wr:oneToClear	1	0x0
[12]	ts1i_detect12	read-write, wr:oneToClear	1	0x0
[11]	ts1i_detect11	read-write, wr:oneToClear	1	0x0
[10]	ts1i_detect10	read-write, wr:oneToClear	1	0x0
[9]	ts1i_detect9	read-write, wr:oneToClear	1	0x0
[8]	ts1i_detect8	read-write, wr:oneToClear	1	0x0
[7]	ts1i_detect7	read-write, wr:oneToClear	1	0x0
[6]	ts1i_detect6	read-write, wr:oneToClear	1	0x0
[5]	ts1i_detect5	read-write, wr:oneToClear	1	0x0
[4]	ts1i_detect4	read-write, wr:oneToClear	1	0x0
[3]	ts1i_detect3	read-write, wr:oneToClear	1	0x0
[2]	ts1i_detect2	read-write, wr:oneToClear	1	0x0
[1]	ts1i_detect1	read-write, wr:oneToClear	1	0x0
[0]	ts1i_detect0	read-write, wr:oneToClear	1	0x0

ts2i_detect15

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect14

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect13

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect12

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect11

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect10

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect9

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect8

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect7

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect6

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect5

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect4

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect3

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect2

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect1

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts2i_detect0

ts2i_detect[i] is set to 1 when an inverted TS2 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect15

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect14

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect13

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect12

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect11

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect10

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect9

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect8

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect7

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect6

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect5

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect4

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect3

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect2

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect1

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

ts1i_detect0

ts1i_detect[i] is set to 1 when an inverted TS1 ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

2.4.1.2.29. pl_rx2 Register 0xb0

Lane Rx Status 2

Field	Name	Access	Width	Reset
[31]	fts_detect15	read-write, wr:oneToClear	1	0x0
[30]	fts_detect14	read-write, wr:oneToClear	1	0x0
[29]	fts_detect13	read-write, wr:oneToClear	1	0x0
[28]	fts_detect12	read-write, wr:oneToClear	1	0x0
[27]	fts_detect11	read-write, wr:oneToClear	1	0x0
[26]	fts_detect10	read-write, wr:oneToClear	1	0x0
[25]	fts_detect9	read-write, wr:oneToClear	1	0x0
[24]	fts_detect8	read-write, wr:oneToClear	1	0x0
[23]	fts_detect7	read-write, wr:oneToClear	1	0x0
[22]	fts_detect6	read-write, wr:oneToClear	1	0x0
[21]	fts_detect5	read-write, wr:oneToClear	1	0x0
[20]	fts_detect4	read-write, wr:oneToClear	1	0x0
[19]	fts_detect3	read-write, wr:oneToClear	1	0x0
[18]	fts_detect2	read-write, wr:oneToClear	1	0x0
[17]	fts_detect1	read-write, wr:oneToClear	1	0x0
[16]	fts_detect0	read-write, wr:oneToClear	1	0x0
[15]	skp_detect15	read-write, wr:oneToClear	1	0x0
[14]	skp_detect14	read-write, wr:oneToClear	1	0x0
[13]	skp_detect13	read-write, wr:oneToClear	1	0x0

Field	Name	Access	Width	Reset
[12]	skp_detect12	read-write, wr:oneToClear	1	0x0
[11]	skp_detect11	read-write, wr:oneToClear	1	0x0
[10]	skp_detect10	read-write, wr:oneToClear	1	0x0
[9]	skp_detect9	read-write, wr:oneToClear	1	0x0
[8]	skp_detect8	read-write, wr:oneToClear	1	0x0
[7]	skp_detect7	read-write, wr:oneToClear	1	0x0
[6]	skp_detect6	read-write, wr:oneToClear	1	0x0
[5]	skp_detect5	read-write, wr:oneToClear	1	0x0
[4]	skp_detect4	read-write, wr:oneToClear	1	0x0
[3]	skp_detect3	read-write, wr:oneToClear	1	0x0
[2]	skp_detect2	read-write, wr:oneToClear	1	0x0
[1]	skp_detect1	read-write, wr:oneToClear	1	0x0
[0]	skp_detect0	read-write, wr:oneToClear	1	0x0

fts_detect15

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect14

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect13

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect12

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect11

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect10

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect9

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect8

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect7

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect6

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect5

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect4

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect3

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect2

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect1

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

fts_detect0

fts_detect[i] is set to 1 when a FTS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect15

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect14

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect13

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect12

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect11

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect10

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect9

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect8

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect7

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect6

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect5

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect4

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect3

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect2

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect1

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

skp_detect0

skp_detect[i] is set to 1 when a SKP ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

2.4.1.2.30. pl_rx3 Register 0xb4

Lane Rx Status 3

Field	Name	Access	Width	Reset
[31]	eie_detect15	read-write, wr:oneToClear	1	0x0
[30]	eie_detect14	read-write, wr:oneToClear	1	0x0
[29]	eie_detect13	read-write, wr:oneToClear	1	0x0
[28]	eie_detect12	read-write, wr:oneToClear	1	0x0
[27]	eie_detect11	read-write, wr:oneToClear	1	0x0
[26]	eie_detect10	read-write, wr:oneToClear	1	0x0
[25]	eie_detect9	read-write, wr:oneToClear	1	0x0
[24]	eie_detect8	read-write, wr:oneToClear	1	0x0
[23]	eie_detect7	read-write, wr:oneToClear	1	0x0
[22]	eie_detect6	read-write, wr:oneToClear	1	0x0
[21]	eie_detect5	read-write, wr:oneToClear	1	0x0
[20]	eie_detect4	read-write, wr:oneToClear	1	0x0
[19]	eie_detect3	read-write, wr:oneToClear	1	0x0
[18]	eie_detect2	read-write, wr:oneToClear	1	0x0
[17]	eie_detect1	read-write, wr:oneToClear	1	0x0
[16]	eie_detect0	read-write, wr:oneToClear	1	0x0
[15]	eios_detect15	read-write, wr:oneToClear	1	0x0
[14]	eios_detect14	read-write, wr:oneToClear	1	0x0
[13]	eios_detect13	read-write, wr:oneToClear	1	0x0
[12]	eios_detect12	read-write, wr:oneToClear	1	0x0
[11]	eios_detect11	read-write, wr:oneToClear	1	0x0
[10]	eios_detect10	read-write, wr:oneToClear	1	0x0
[9]	eios_detect9	read-write, wr:oneToClear	1	0x0
[8]	eios_detect8	read-write, wr:oneToClear	1	0x0
[7]	eios_detect7	read-write, wr:oneToClear	1	0x0
[6]	eios_detect6	read-write, wr:oneToClear	1	0x0
[5]	eios_detect5	read-write, wr:oneToClear	1	0x0
[4]	eios_detect4	read-write, wr:oneToClear	1	0x0
[3]	eios_detect3	read-write, wr:oneToClear	1	0x0
[2]	eios_detect2	read-write, wr:oneToClear	1	0x0
[1]	eios_detect1	read-write, wr:oneToClear	1	0x0
[0]	eios_detect0	read-write, wr:oneToClear	1	0x0

eie_detect15

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect14

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect13

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect12

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect11

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect10

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect9

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect8

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect7

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect6

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect5

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect4

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect3

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect2

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect1

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eie_detect0

eie_detect[i] is set to 1 when a EIE ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect15

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect14

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect13

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect12

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect11

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect10

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect9

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect8

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect7

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect6

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect5

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect4

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect3

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect2

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect1

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

eios_detect0

eios_detect[i] is set to 1 when a EIOS ordered set is received on Lane[i]. 0 otherwise. This bit stays asserted once set. Write 1 to clear.

2.4.1.2.31. debug_self_crosslink Register 0xc0

LTSSM self crosslink

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

Self crosslink enable.

- 0 – Otherwise.
- 1 – For debug use only, configure LTSSM so that it links with itself when core Tx is externally looped back to core Rx.

2.4.1.2.32. debug_rx_det Register 0xc4

LTSSM receiver detection bypass configuration.

Field	Name	Access	Width	Reset
[31:17]	*reserved*	read-only	15	0x0
[16]	inhibit	read-write	1	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	bypass	read-write	1	0x0

inhibit

Link receiver detection inhibit.

- 0 – Perform receiver detection and use result to determine whether to include/exclude lanes from the link.
- 1 – Skip receiver detection and assume receivers are not present on all lanes.

bypass

Link receiver detection bypass. If both bypass and inhibit are asserted, bypass takes precedence.

- 0 – Perform receiver detection and use result to determine whether to include/exclude lanes from the link.
- 1 – Skip receiver detection and assume receivers are present on all lanes.

2.4.1.2.33. debug_force_tx Register 0xc8

PIPE Tx Debug.

Field	Name	Access	Width	Reset
[31:10]	*reserved*	read-only	22	0x0
[9]	deemph_5g_enable	read-write	1	0x0
[8]	deemph_5g_3_5db_6db_n	read-write	1	0x0
[7:4]	*reserved*	read-only	4	0x0
[3]	margin_enable	read-write	1	0x0
[2:0]	margin_value	read-write	3	0x0

deemph_5g_enable

For 5G capable cores only: Force pipe_tx_deemph at 5G enable.

- 0 – Disable.
- 1 – Enable. Force phy_tx_deemph at 5G speed to the value specified by deemph_5g_6db_3_5db_n. The force is applied at 5G speed except during Polling.Compliance, where for compatibility with PCI SIG Workshop Electrical Testing, the force is not applied.

deemph_5g_3_5db_6db_n

For 5G capable cores only: Force pipe_tx_deemph at 5G value.

- 0 – -6dB
- 1 – -3.5dB

margin_enable

Force pipe_tx_margin enable.

- 0 – Drive pipe_tx_margin per PCIe Specification.
- 1 – Drive pipe_tx_margin to value.

margin_value

Force pipe_tx_margin Value.

2.4.1.2.34. debug_direct_scramble_off Register 0xcc

Scambling disable control for less than or equal to 5G line rates.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

LTSSM direct scrambling disabled at 2.5G and 5G.

- 0 – Otherwise.
- 1 – Direct to disable scrambling at 2.5G and 5G during Configuration.Complete.

2.4.1.2.35. debug_pipe_rx Register 0xe0

PIPE Interface Debug status.

Field	Name	Access	Width	Reset
[31:16]	polarity	read-only	16	0x0
[15:0]	valid	read-only	16	0x0

polarity

PHY PIPE Interface pipe_rx_polarity current value. For each lane:

- 0 – Otherwise.
- 1 – PHY lane has been instructed to invert its receiver polarity to compensate for serial rx_p and rx_n being swapped.

valid

PHY PIPE Interface pipe_rx_valid current value. For each lane:

- 0 – Otherwise.
- 1 – PHY lane is locked to data stream.

2.4.1.2.36. debug_direct_to_loopback Register 0x100

LTSSM master loopback enable

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

LTSSM master loopback enable.

- 0 – Otherwise.
- 1 – Direct LTSSM to Loopback.Master. Before this field is set to 1, all relevant regsiters containing Master Loopback control options must be set to their desired values. When mgmt_tlb_debug_direct_to_loopback == 1 no Master Loopback control options may be changed.

2.4.1.2.37. debug_loopback_control Register 0x104

Loopback control

Field	Name	Access	Width	Reset
[31:28]	inject_err_lane_select	read-write	4	0x0
[27]	inject_rx_2bit_data_err	read-write	1	0x0
[26]	inject_rx_1bit_data_err	read-write	1	0x0
[25]	inject_rx_valid_err	read-write	1	0x0
[24]	inject_rx_skp_err	read-write	1	0x0
[23:19]	*reserved*	read-only	5	0x0
[18:16]	pattern	read-write	3	0x0
[15:9]	*reserved*	read-only	7	0x0
[8]	tx_comp_receive	read-write	1	0x0
[7:2]	*reserved*	read-only	6	0x0
[1:0]	speed	read-write	2	0x0

inject_err_lane_select

Lane selection to inject error in Loopback. Only lanes configured by the core may be programmed. 0 = Lane 0 .. 15 = Lane 15.

inject_rx_2bit_data_err

When enabled during loopback, the rising edge of inject_rx_2bit_data_err bit injects back-to-back errors on the received loopback data. This simulates the PHY losing lock and increments the counter by one, and the Loopback Master restarts the loopback pattern so that the PHY can recover symbol lock.

inject_rx_1bit_data_err

When enabled during loopback, the rising edge of inject_rx_1bit_data_err bit injects a single clk error on the received loopback data. This causes the error count to increment by 1 for each received data byte.

inject_rx_valid_err

When enabled during loopback, the rising edge of inject_rx_valid_err bit injects a single clk error on the received PIPE PHY interface phy_rx_valid signal. This simulates the PHY losing lock during Loopback Master operation which causes the error count to increment by one, and the Loopback Master restarts the loopback pattern so that the PHY can recover symbol lock.

inject_rx_skp_err

When enabled during loopback, the rising edge of inject_rx_skp_err bit injects a single clk error on the next received SKP Ordered Set. When a SKP Ordered Set is corrupted, that lane's RX descrambling LFSR goes out of sync with the transmitter lane's scrambling LFSR causing all the subsequent data checks to fail. This simulates the PHY losing lock and increments the counter by one, and the Loopback Master restarts the loopback pattern so that the PHY can recover symbol lock.

pattern

Loopback data pattern.

- 0 – Unscrambled PRBS31 Polynomial Pattern using Galois implementation with non-inverted output. The polynomial representation is G(x) = X31 + X28 +1.

tx_comp_receive

Loopback compliance receive behavior.

- 0 – Loopback Master does not assert Compliance Receive (recommended default)
- 1 – Loopback Master asserts Compliance Receive in TS sets transmitted during Loopback Entry

speed

Desired speed in loopback. Only speeds supported by the core may be programmed. A speed change is only implemented if Loopback is entered from Configuration; if entered from Recovery, then the speed is not changed.

- 0 – 2.5 G
- 1 – 5 G

2.4.1.2.38. debug_loopback_master_5g Register 0x108

Loopback master control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	deemph	read-write	1	0x0

deemph

Select Deemphasis value used by Loopback Master when Loopback.Active occurs at 5G data rate.

- 0 – -6.0dB
- 1 – -3.5dB

2.4.1.2.39. debug_loopback_slave_5g Register 0x10c

Loopback slave control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	deemph	read-write	1	0x0

deemph

Select Deemphasis value transmitted in TS sets for the Slave to use when Loopback.Active occurs at 5G data rate.

- 0 – -6.0dB
- 1 – -3.5dB

2.4.1.2.40. debug_direct_to_loopback_status Register 0x118

Master loopback status.

Field	Name	Access	Width	Reset
[31:16]	sync	read-only	16	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	cfg_entry	read-only	1	0x0

sync

Loopback per lane sync to data pattern indicator. For each lane:

- 0 – Not locked to loopback pattern.
- 1 – Locked to loopback pattern.

cfg_entry

Loopback entered from Configuration or Recovery indicator.

- 0 – Loopback entry was from Recovery.
- 1 – Loopback entry was from Configuration.

2.4.1.2.41. debug_loopback_err_reset Register 0x11c

Master loopback error reset

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

Loopback error counter reset.

- 0 – Master Loopback error count increments as errors are detected during Master Loopback – saturating at maximum value.
- 1 – Reset the master loopback error count on all lanes to 0x0. The reset stays in force for as long as mgmt_tlb_debug_loopback_err_reset_enable remains at 1.

2.4.1.2.42. debug_loopback_err Register 0x120

Master loopback error count

Field	Name	Access	Width	Reset
[255:0]	count	read-only	256	0x0

count

Loopback per lane error count – 16 bits per lane. Errors are counted only after the lane is locked to the loopback pattern

2.4.1.2.43. phy_control Register 0x140

LTSSM PIPE Interface configuration

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	pipe_tx_swing	read-write	1	0x0

pipe_tx_swing

Directly controls the value of pipe_tx_swing which sets PHY 2.5G/5G Transmitter Amplitude. Full Swing is required for most applications. Reduced Swing is useful to support low power form factors which encourage or require reduced transmitter amplitudes.

- 0 – Full Swing
- 1 – Reduced Swing

2.4.1.2.44. phy_eq_tx_max Register 0x14c

For >= 8G capable cores only: Local PHY maximum allowed coefficient values

Field	Name	Access	Width	Reset
[31:30]	*reserved*	read-only	2	0x0
[29:24]	pre	read-write	6	0x0
[23:22]	*reserved*	read-only	2	0x0
[21:16]	post	read-write	6	0x0
[15:0]	*reserved*	read-only	16	0x0

pre

Local PHY transmitter maximum pre-cursor[5:0] coefficient value. If a coefficient requests exceeds mgmt_tlb_phy_eq_tx_max_pre, then the coefficient is limited to mgmt_tlb_phy_eq_tx_max_pre before being passed to the PHY.

post

Local PHY transmitter maximum post-cursor[5:0] coefficient value. If a coefficient requests exceeds mgmt_tlb_phy_eq_tx_max_post, then the coefficient is limited to mgmt_tlb_phy_eq_tx_max_post before being passed to the PHY.

2.4.1.2.45. pl_tx_skp Register 0x344

Physical Layer Transmit SKP Period Control.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	2	0x0
[15:8]	period_sris_8b10b	read-write	8	0x0
[7:0]	period_srns_8b10b	read-write	8	0x0

period_sris_8b10b

The transmit SKP period used when operating at <= 5G with the SRIS capability enabled and configured for SRIS = period_sris_8b10b Symbol Times. PCIe Spec. is < 154 Symbol Times. 0 is a special case that selects 146 Symbol Times. This register must be configured for a PCIe Spec. compliant value. The number of symbol times selected must be a multiple of the PHY per lane symbol data width or the lower bits are truncated. For example: For 16-bit per lane PHY, period_sris_8b10b[0] is always treated as 0. For 32-bit per lane PHY, period_sris_8b10b[1:0] are always treated as 00. For 64-bit per lane PHY, period_sris_8b10b[2:0] are always treated as 000.

period_srns_8b10b

The transmit SKP period used when operating at <= 5G with the SRIS capability disabled or with SRIS enabled but configured for SRNS = (256 + period_srns_8b10b) * 4 Symbol Times. PCIe Spec. is 1180-1538 Symbol Times. 0 is a special case that selects 44 == 1200 Symbol Times. This register must be configured for a PCIe Spec. compliant value. The number of symbol times selected must be a multiple of the PHY per lane symbol data width or the lower bits are truncated. For example: For 64-bit per lane PHY period_srns_8b10b[0] is always treated as 0. For 32, 16, and 8-bit per lane PHY, all of period_srns_8b10b is relevant.

2.4.1.2.46. pl_tx_debug Register 0x348

Physical Layer Debug Control

Field	Name	Access	Width	Reset
[31:3]	*reserved*	read-only	29	0x0
[2]	inject_margin_crc_error	read-write	1	0x0
[1]	inject_margin_parity_error	read-write	1	0x0
[0]	inject_data_parity_error	read-write	1	0x0

inject_margin_crc_error

Setting this to 1 injects errors into the margin crc value of the control skp ordered set on all lanes

inject_margin_parity_error

Setting this to 1 injects errors into the margin parity bit of the control skp ordered set on all lanes

inject_data_parity_error

Setting this to 1 injects errors into the data parity bit of the control skp ordered set on all lanes

2.4.1.2.47. pl_ctrl Register 0x34c

Physical Layer Control

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	8b10b_err_rec_entry_sel	read-write	1	0x0

8b10b_err_rec_entry_sel

Selects the Physical Layer error threshold required to be received in L0, when operating with 8b10b encoding (2.5G/5G speed), before the link is directed to Recovery.

- 0 – When in L0 and using 8b10b encoding, direct the link to Recovery after receiving a single Physical Layer Error. This is the more conservative setting, but has the disadvantage of causing Recovery entry on all L0 Physical Layer errors – even those errors that the link would be able to recover from on its own without having to go through Recovery.
- 1 – When in L0 and using 8b10b encoding, direct the link to Recovery only after receiving a massive burst of errors (which typically is an indication that there is a persistent problem, such as PHY loss of lock, for which Recovery entry is required to fix). The core implements an error counter. For each enabled PHY Rx clock cycle, the error counter is incremented when a clock cycle contains a Physical Layer error and the error counter is decremented when a clock cycle contains no Physical Layer errors. If the counter reaches 1023, then the link is directed to Recovery.

2.4.1.2.48. pl_ts_matching Register 0x350

Physical Layer TS Match Control

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	legacy_mode	read-write	1	0x1

Legacy_mode

Setting this to 1 compares all symbols when matching TS sets (legacy behavior). Setting this to 0 compares only the symbols required to meet Spec.

2.4.1.2.49. dl_retry_timeout Register 0x380

Replay Timeout Control.

Field	Name	Access	Width	Reset
[31:24]	pcie4_symt_sync	read-write	8	0x0
[23:16]	pcie4_symt_sync_n	read-write	8	0x0
[15]	pcie4_enable	read-write	1	0x1
[14:1]	l0s_adj	read-write	14	0x180
[0]	mult_enable	read-write	1	0x0

pcie4_symt_sync

Replay Timeout Timer PCIe 4.0 Simplified REPLAY_TIMER Limits Extended Sync==1 Value. {pcie4_symt_sync, 10'h0} = Symbol times to use for Replay Timer when pcie4_enable==1 & Extended Synch==1. 0 is a special case selecting 8'd80.

pcie4_symt_sync_n

Replay Timeout Timer PCIe 4.0 Simplified REPLAY_TIMER Limits Extended Sync==0 Value. {pcie4_symt_sync_n, 10'h0} = Symbol times to use for Replay Timer when pcie4_enable==1 & Extended Synch==0. 0 is a special case selecting 8'd24.

pcie4_enable

Replay Timeout Timer PCIe 4.0 Simplified REPLAY_TIMER Limits Enable.

- 0 – Use PCIe 3.0 Specification and prior REPLAY_TIMER Limits from UNADJUSTED REPLAY_TIMER LIMITS tables in the PCIe Specification.
- 1 – Use PCIe 4.0 Specification Simplified REPLAY_TIMER Limits.

l0s_adj

Replay Timeout L0s Adjustment. The number of symbol times to add to the recommended PCIe Replay Timer timeout period to compensate for the remote link having to exit L0s before it can send an ACK/NAK DLLP. l0s_adj is added to the Replay Timer timeout period after the optional doubling controlled by mult_enable is applied. Not applicable when pcie4_enable==1.

mult_enable

Replay Timeout Timer Multiplier Enable. Not applicable when pcie4_enable==1.

- 0 – The Replay Timer timeout period implemented is the recommended (-0%) value in the PCIe Spec. UNADJUSTED REPLAY_TIMER LIMITS FOR 2.5/5.0/GT/S MODE OPERATION BY LINK WIDTH AND MAX_PAYLOAD_SIZE tables.
- 1 – The Replay Timer timeout period implemented is 2 times the recommended (-0%) value in the PCIe Spec. UNADJUSTED REPLAY_TIMER LIMITS FOR 2.5/5.0/GT/S MODE OPERATION BY LINK WIDTH AND MAX_PAYLOAD_SIZE tables.

2.4.1.2.50. dl_ack_timeout_div Register 0x384

ACK Timer Control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

ACK Timer Control.

- 0 – Ack according to Spec.
- 1 – Ack twice as often as recommended by Spec.

2.4.1.2.51. dl_ctrl Register 0x390

Data Link Layer Control.

Field	Name	Access	Width	Reset
[31:26]	*reserved*	read-only	6	0x0
[25]	tx_pfx_par_inject_en	read-write	1	0x0
[24]	rx_early_forward_disable	read-write	1	0x0
[23]	*reserved*	read-only	1	0x0
[22]	tx_gap_inject_en	read-write	1	0x0
[21]	rx_malf_inject_en	read-write	1	0x0
[20]	rx_lcrc_inject_en	read-write	1	0x0
[19]	*reserved*	read-only	1	0x0
[18]	rx_dl_active_disable	read-write	1	0x0
[17]	rx_inhibit_tlp	read-write	1	0x0
[16]	rx_inhibit_ack_nak	read-write	1	0x0
[15]	*reserved*	read-only	1	0x0
[14]	tx_par2_report_disable	read-write	1	0x0
[13]	tx_par2_handle_disable	read-write	1	0x0
[12]	tx_par2_inject_en	read-write	1	0x0
[11:9]	*reserved*	read-only	3	0x0
[8]	tx_par1_inject_en	read-write	1	0x0
[7]	*reserved*	read-only	1	0x0
[6]	tx_replay_ecc2_report_disable	read-write	1	0x0
[5]	tx_replay_ecc2_handle_disable	read-write	1	0x0
[4]	tx_replay_ecc2_inject_en	read-write	1	0x0
[3]	*reserved*	read-only	1	0x0
[2]	tx_replay_ecc1_report_disable	read-write	1	0x0
[1]	tx_replay_ecc1_handle_disable	read-write	1	0x0
[0]	tx_replay_ecc1_inject_en	read-write	1	0x0

tx_pfx_par_inject_en

Transmit Data Link Layer Prefix Parity Error Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single prefix parity error injection, applied prior to assigning the TLP sequence number, is scheduled and injected at the next opportunity (TLP transmit). Tx prefix parity error Handling and Reporting are governed by tx_par2_handle_disable and tx_par2_report_disable

rx_early_forward_disable

Receive Data Link Layer down-trained early forwarding disable.

- 0 – When down-trained, forward Rx Data Link Layer data for processing whenever a TLP/DLLP end occurs without a following TLP/DLLP start the same clock cycle. This setting results in lower Rx TLP/DLLP latency.
- 1 – When down-trained, always aggregate Rx Data Link Layer data to full width before forwarding the data. For example, a x16 core operating at x1 receives and aggregates 16 clock cycles of 1 lane data before outputting one clock cycle of 16 lane data for further processing.

tx_gap_inject_en

Transmit Data Link Layer TX Valid Gap Injection Enable.

- 0 – Do not inject gap.
- 1 – On the rising edge, a single clock bp_tx_valid gap is scheduled and injected at the next opportunity (within a TLP). This gap in the bp_tx_valid can cause a data underflow at the Physical Layer.

rx_malf_inject_en

Receive Data Link Layer Malformed Length TLP Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single malformed TLP error injection is scheduled and injected at the next opportunity (Tx TLP EOP). The TLP is malformed by deleting its end of TLP indicator causing the TLP to end at the incorrect location.

rx_lcrc_inject_en

Receive Data Link Layer LCRC Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single LCRC TLP error injection is scheduled and injected at the next opportunity (Tx TLP EOP). The LCRC is corrupted by inverting LCRC bit 0 of the received TLP.

rx_dl_active_disable

Control the use of DL_Active to block reception of TLPs.

- 0 – Block reception of TLPs when dl_active is low.
- 1 – Do not block TLP reception based on dl_active.

rx_inhibit_tlp

Receive Data Link Layer TLP Rx Inhibit Enable.

- 0 – Process received TLPs per PCIe Spec. Required setting for compliant PCIe operation.
- 1 – For test purposes only, discard and do not accept received TLPs. Received TLPs are processed as if their Sequence Number was one greater than received. This prevents the TLP with the current expected Sequence Number and all following TLPs from being received. This causes the link partner to do TLP replays because received TLPs are NAKed due to perceived Sequence Number errors.

rx_inhibit_ack_nak

Receive Data Link Layer ACK/NAK Inhibit Enable.

- 0 – Process received ACK and NAK DLLPs per PCIe Spec. Required setting for compliant PCIe operation.
- 1 – For test purposes only, discard and do not process received ACK and NAK DLLPs. This causes the core to do TLP replays because TLP acknowledgements is not received.

tx_par2_report_disable

Transmit Data Link Layer Parity 2 Error Reporting Disable.

- 0 – Enable reporting.
- 1 – Disable reporting of Data Link Layer transmit parity errors.

tx_par2_handle_disable

Transmit Data Link Layer Parity 2 Error Handling Disable.

- 0 – Enable handling. TLPs with errors are nullfied and not retransmitted.
- 1 – Disable handling of Data Link Layer transmit parity errors. When error handling is disabled, TLPs with parity errors continue to be transmitted.

tx_par2_inject_en

Transmit Data Link Layer Parity 2 Error Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single parity error injection, applied after assigning the TLP sequence number, is scheduled and injected at the next opportunity (TLP transmit).

tx_par1_inject_en

Transmit Data Link Layer Parity 1 Error Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single parity error injection, applied prior to assigning the TLP sequence number, is scheduled and injected at the next opportunity (TLP transmit).

tx_replay_ecc2_report_disable

Transmit Replay Buffer ECC 2-bit Error Reporting Disable.

- 0 – Enable reporting.
- 1 – Disable reporting of ECC 2-bit errors.

tx_replay_ecc2_handle_disable

Transmit Replay Buffer ECC 2-bit Error Handling Disable.

- 0 – Enable handling. TLPs with errors are nullfied and not retransmitted.
- 1 – Disable hanlding of ECC 2-bit errors. When error handling is disabled, TLPs with ECC 2-bit errors continue to be transmitted.

tx_replay_ecc2_inject_en

Transmit Replay Buffer ECC 2-bit Error Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single ECC 2-bit error injection is scheduled and injected at the next opportunity (Replay Buffer RAM write). The error is only seen if a Replay occurs of the TLP receiving the error injection.

tx_replay_ecc1_report_disable

Transmit Replay Buffer ECC 1-bit Error Reporting Disable.

- 0 – Enable reporting.
- 1 – Disable reporting of ECC 1-bit errors.

tx_replay_ecc1_handle_disable

Transmit Replay Buffer ECC 1-bit Error Handling Disable.

- 0 – Enable correction.
- 1 – Disable correction of ECC 1-bit errors. When error correction is disabled, ECC 1-bit errors are treated the same as uncorrectable ECC 2-bit errors.

tx_replay_ecc1_inject_en

Transmit Replay Buffer ECC 1-bit Error Injection Enable.

- 0 – Do not inject error.
- 1 – On the rising edge, a single ECC 1-bit error injection is scheduled and injected at the next opportunity (Replay Buffer RAM write). The error is only seen if a Replay occurs of the TLP receiving the error injection.

2.4.1.2.52. dl_stat Register 0x394

Data Link Layer Status.

Field	Name	Access	Width	Reset
[31]	info_bad_tlp_null_err	read-write, wr:oneToClear	1	0x0
[30]	info_bad_tlp_phy_err	read-write, wr:oneToClear	1	0x0
[29]	info_bad_tlp_malf_err	read-write, wr:oneToClear	1	0x0
[28]	info_bad_tlp_ecrc_err	read-write, wr:oneToClear	1	0x0
[27]	info_schedule_dupl_ack	read-write, wr:oneToClear	1	0x0
[26]	info_bad_tlp_seq_err	read-write, wr:oneToClear	1	0x0
[25]	info_bad_tlp_crc_err	read-write, wr:oneToClear	1	0x0
[24]	info_nak_received	read-write, wr:oneToClear	1	0x0
[23]	info_deskew_overflow_error	read-write, wr:oneToClear	1	0x0
[22]	info_tx_data_underflow	read-write, wr:oneToClear	1	0x0
[21]	info_replay_started	read-write, wr:oneToClear	1	0x0
[20]	*reserved*	read-only	1	0x0
[19]	err_aer_tx_par2	read-write, wr:oneToClear	1	0x0
[18]	*reserved*	read-only	1	0x0
[17]	err_aer_tx_replay_ecc2	read-write, wr:oneToClear	1	0x0
[16]	err_aer_tx_replay_ecc1	read-write, wr:oneToClear	1	0x0
[15:8]	*reserved*	read-only	8	0x0
[7]	*reserved*	read-only	1	0x0
[6]	err_aer_surprise_down	read-write, wr:oneToClear	1	0x0
[5]	err_aer_dl_protocol_error	read-write, wr:oneToClear	1	0x0
[4]	err_aer_replay_timer_timeout	read-write, wr:oneToClear	1	0x0
[3]	err_aer_replay_num_rollover	read-write, wr:oneToClear	1	0x0
[2]	err_aer_bad_dllp	read-write, wr:oneToClear	1	0x0
[1]	err_aer_bad_tlp	read-write, wr:oneToClear	1	0x0
[0]	err_aer_receiver_error	read-write, wr:oneToClear	1	0x0

info_bad_tlp_null_err

Nullified TLP Received Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_bad_tlp_phy_err

TLP PHY Error Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_bad_tlp_malf_err

Malformed TLP Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_bad_tlp_ecrc_err

TLP ECRC Mismatch Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_schedule_dupl_ack

Duplicate TLP Received Status. This is not a reported error, but is useful information to store for debug. Duplicate TLPs are received during TLP Replay.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_bad_tlp_seq_err

TLP Sequence Number Mismatch Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_bad_tlp_crc_err

TLP LCRC Mismatch Status. This is not a reported error, but is useful information to store for debug. This is a sub-class of err_aer_bad_tlp that provides more information as to why the TLP was bad.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_nak_received

NAK Received Status. This is not a reported error, but is useful information to store for debug. Receiving a NAK indicates that the link partner requested a Replay.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_deskew_overflow_error

Rx Deskew FIFO Overflow Error Status. The lane-lane skew of Rx data on one or more lanes was so latent from the other lanes that the deskew range of the Rx Deskew FIFO was exceeded. This is a correctable error since the core drives the link to Recovery to fix this issue.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_tx_data_underflow

Physical Layer TLP Transmit Underflow Error Status. A TLP was being transmitted by the physical layer and more data was needed to continue the transmission but no data was provided. This error is normally caused by the Transaction Layer failing to provide TLP data at >= PCIe Line Rate.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

info_replay_started

A Replay was started. This is not a reported error, but is useful information to store for debug. Indicates a Replay occurred on local TX interface due to either Ack Timeout or Nack Reception.

- 0 – Otherwise.
- 1 – Event occurred. Write 1 to clear.

err_aer_tx_par2

Transmit Data Link Layer Parity 2 Error Status. Indicates that a Data Link Layer transmit parity error was detected after sequence number application.

- 0 – Otherwise.
- 1 – Error occurred. Write 1 to clear.

err_aer_tx_replay_ecc2

Transmit Replay Buffer ECC 2-bit Error Status. Indicates that an uncorrectable ECC error occurred during Replay.

- 0 – Otherwise.
- 1 – Error occurred. Write 1 to clear.

err_aer_tx_replay_ecc1

Transmit Replay Buffer ECC 1-bit Error Status. Indicates that a correctable ECC error occurred during Replay.

- 0 – Otherwise.
- 1 – Error occurred. Write 1 to clear.

err_aer_surprise_down

Surprise Down Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_dl_protocol_error

DL Protocol Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_replay_timer_timeout

ReplayTimer Timeout Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_replay_num_rollover

Replay Num Rollover Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_bad_dllp

Bad DLLP Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_bad_tlp

Bad TLP Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

err_aer_receiver_error

Receiver Error Status.

- 0 – Otherwise.
- 1 – Error occurred. Errors of this type must be logged in the Transaction Layer AER Capability. Write 1 to clear.

2.4.1.2.53. dl_ack_to_nak Register 0x398

ACK-to-NAK error injection controls.

Field	Name	Access	Width	Reset
[31]	enable	read-write	1	0x0
[30:24]	*reserved*	read-only	7	0x0
[23:16]	count	read-write	8	0x0
[15:12]	*reserved*	read-only	4	0x0
[11:0]	seq_num	read-write	12	0x0

enable

Enable ACK-to-NAK injection. The write to this field must occur after writes to the other fields in this register have established the desired parameters of the injection.

- 0 – Do nothing
- 1 – Enable injection and load parameters into the ACK-to-NAK injector.

count

Number of times to replace the ACK with a NAK.

seq_num

Sequence Number of ACK to be changed to a NAK.

2.4.1.2.54. dl_inject Register 0x39c

DLLP CRC/TLP ECRC error injection controls.

Field	Name	Access	Width	Reset
[31]	dllp_crc_err_enable	read-write	1	0x0
[30:28]	*reserved*	read-only	3	0x0
[27:16]	dllp_crc_err_rate	read-write	12	0x0
[15:13]	*reserved*	read-only	3	0x0
[12]	dllp_inject_enable	read-write	1	0x0
[11:9]	*reserved*	read-only	3	0x0
[8]	tlp_seq_err_enable	read-write	1	0x0
[7:4]	*reserved*	read-only	4	0x0
[3]	tlp_lcrc_err_enable	read-write	1	0x0
[2:0]	tlp_lcrc_err_rate	read-write	3	0x0

dllp_crc_err_enable

Enable DLLP CRC error injection. The write to this field must occur after writes to the other fields in this register have established the desired parameters of the injection.

- 0 – Disable the DLLP CRC Error injector.
- 1 – Enable DLLP CRC Error injector.

dllp_crc_err_rate

Rate at which DLLP CRC errors are to be injected. A value of 0 injects a single DLLP CRC error. A non-zero value injects errors at intervals of Rate*256*clk_period. This field may not be changed while dllp_crc_err_enable==1.

dllp_inject_enable

Inject a DLLP (transmit) using the data in the dllp_inject_data register. A single DLLP is injected after each rising edge of this signal.

tlp_seq_err_enable

Modify the sequence number in the next transmitted TLP to an invalid value (bad sequence number error). A single TLP is altered after each rising edge of this signal.

tlp_lcrc_err_enable

Enable TLP LCRC error injection. The write to this field must occur after writes to the other fields in this register have established the desired parameters of the injection.

- 0 – Disable the TLP LCRC Error injector
- 1 – Enable TLP LCRC Error injector

tlp_lcrc_err_rate

Rate at which TLP LCRC errors are to be injected. A value of 0 injects LCRC errors into all TLPs. A non-zero vlaue injects an error into a TLP and then pass Rate TLPs without error, and then repeat. This field may not be changed while tlp_lcrc_err_enable==1.

2.4.1.2.55. dllp_inject Register 0x3a0

DLLP Injector Data.

Field	Name	Access	Width	Reset
[31:0]	data	read-write	32	0x0

data

Data to include in injected DLLP. This field may not be changed while dl_inject_dllp_inject_enable==1.

2.4.1.3. mgmt_ptl (0x03000)

mgmt_ptl_BASE 0x3000

2.4.1.3.1. simulation Register 0x0

Partial Transaction Layer simulation speed reduction.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	pm_reduce_timeouts	read-write	1	0x0

pm_reduce_timeouts

Reduce Power Management State Machine timeouts from their value in ms to their value in uS to shorten simulation time.

- 0 – Disable
- 1 – Enable

2.4.1.3.2. pm_aspm_l0s Register 0x40

Power Management State Machine ASPM L0s entry control.

Field	Name	Access	Width	Reset
[31:16]	entry_time	read-write	16	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	enable	read-write	1	0x1

entry_time

ASPM L0s TX Entry Time in uS. 0 is a special case == 6.9uS.

enable

Enable Power Management State Machine to enter ASPM L0s.

- 0 – Disable
- 1 – Enable

2.4.1.3.3. pm_aspm_l1 Register 0x50

Power Management State Machine ASPM L1 entry control.

Field	Name	Access	Width	Reset
[31:16]	entry_time	read-write	16	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	enable	read-write	1	0x1

entry_time

ASPM L1 TX Entry Time in mS. 0 is a special case == 1000 uS.

enable

Enable Power Management State Machine to enter ASPM L1.

- 0 – Disable
- 1 – Enable

2.4.1.3.4. pm_aspm_l1_min Register 0x54

Power Management State Machine ASPM L1 reentry control.

Field	Name	Access	Width	Reset
[31:30]	*reserved*	read-only	2	0x0
[29:16]	reentry_time	read-write	14	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	reentry_disable	read-write	1	0x0

reentry_time

When reentry_disable==0, specifies the minimum time between ASPM L1 requests in ns. 0 is a special case == 9500 ns (PCIe Spec. value).

reentry_disable

Disable enforcing a minimum time between ASPM L1 requests.

- 0 – Enable
- 1 – Disable

2.4.1.3.5. pm_l1 Register 0x60

Power Management State Machine L1 entry control.

Field	Name	Access	Width	Reset
[31:16]	us_port_ps_entry_time	read-write	16	0x0
[15:1]	*reserved*	read-only	15	0x0
[0]	enable	read-write	1	0x1

us_port_ps_entry_time

Upstream Ports only: Number of uS to wait for the transmission of the completion to the PowerState Cfg Write that initiated L1 entry, before beginning to block TLPs and enter L1. 0 is a special case == 4us.

enable

Enable Power Management State Machine to enter L1.

- 0 – Disable
- 1 – Enable

2.4.1.3.6. pm_l1_min Register 0x64

Power Management State Machine L1 reentry control.

Field	Name	Access	Width	Reset
[31:24]	*reserved*	read-only	8	0x0
[23:16]	ps_reentry_time	read-write	8	0x0
[15:0]	*reserved*	read-only	16	0x0

ps_reentry_time

Minimum number of uS to wait following an L1 exit when Power State != D0, before re-entering L1 due to Power State != D0. A wait time is needed to give the transaction layer time to process a Power State Cfg Write to D0 that caused L1 exit. 0 is a special case == 50 us.

2.4.1.3.7. pm_l1pmss Register 0x68

Power Management State Machine L1PMSS control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	ds_drive_clkreq	read-write	1	0x0

ds_drive_clkreq

Enable driveing the clkreq_n signal when operating as a downstream port.

- 0 – Disable
- 1 – Enable

2.4.1.3.8. pm_l2 Register 0x70

Power Management State Machine L2 entry control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x1

enable

Enable Power Management State Machine to enter L2.

- 0 – Disable
- 1 – Enable

2.4.1.3.9. pm_pme_to_ack_ep Register 0x80

Power Management State Machine Endpoint PME_TO_Ack control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	user_auto_n	read-write	1	0x0

user_auto_n

For Endpoints only: PME_TO_Ack message transmission scheduling method. Endpoints are required to respond to a PME_Turn_Off message with a PME_TO_Ack message when they are ready to allow power down.

- 0 – Schedule PME_TO_Ack message automatically on reception of PME_Turn_Off message.
- 1 – Schedule PME_TO_Ack message under user control via pm_l2_enter_ack rising edge.

2.4.1.3.10. pm_pme_to_ack_ds Register 0x84

Power Management State Machine Downstream Port PME_TO_Ack control.

Field	Name	Access	Width	Reset
[31:8]	reserved	read-only	24	0x0
[7:0]	timeout_threshold	read-write	8	0x0

timeout_threshold

For Root Port only: mS to wait for a transmitted PME_Turn_Off Message to be acknowledged by receipt of PME_TO_Ack message before continuing with L2/L3 entry. 0xFF is a special case that disables the timeout mechanism. 0x00 is a special case == 10 mS.

2.4.1.3.11. pm_pme Register 0x88

Power Management State Machine PM_PME control.

Field	Name	Access	Width	Reset
[31:12]	*reserved*	read-only	20	0x0
[11:0]	timeout_threshold	read-write	12	0x0

timeout_threshold

mS to wait for a transmitted PM_PME Message to be acknowledged, by clearing of the PME_Status register, before re-issuing the PM_PME message. 0xFFF is a special case that disables the timeout mechanism. 0x000 is a special case == 100 mS.

2.4.1.3.12. pm_status Register 0x90

Power Management State Machine Status.

Field	Name	Access	Width	Reset
[31:5]	*reserved*	read-only	27	0x0
[4:0]	state	read-only	5	0x0

state

Power Management State Machine State.

- 0 – IDLE
- 1 – L1_WAIT_IDLE
- 2 – L1_WAIT_REPLAY
- 3 – L1_READY
- 4 – L1_STOP_DLLP
- 5 – L1
- 6 – L1_1
- 7 – L1_2_ENTRY

- 8 – L1_2_IDLE
- 9 – L1_2_EXIT
- 10 – L1_EXIT
- 11 – L2_WAIT_IDLE
- 12 – L2_WAIT_REPLAY
- 13 – L23_READY
- 14 – L2_STOP_DLLP
- 15 – L2
- 16 – L0S

2.4.1.3.13. tlp_tx Register 0x1c4

TLP Transmit Control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	td1_means_add_has_n	read-write	1	0x0

td1_means_add_has_n

TLP Transmit TD==1 Header Field Interpretation.

- 0 – When a TLP is transmitted with TLP header bit TD==1, this means that the TLP already contains an ECRC. The core transmits the TLP with the TLP's existing ECRC and does not attempt to generate/append a new ECRC.
- 1 – When a TLP is transmitted with TLP header bit TD==1, this means that you are requesting that an ECRC be generated and appended to the TLP. In this setting, PCIe Configuration Register ECRC Generation Enable==1 is not used by the core. You are expected to consider ECRC Generation Enable when setting the TD bit in transmitted TLPs. When this field is set to 1, it is illegal for transmitted TLPs to already contain an ECRC; if a transmitted TLP already contains an ECRC when the core is instructed (via TLP Header TD==1) to add another ECRC, this results in a serious error such that the core is not able to properly transmit the TLP and the link is compromised.

2.4.1.3.14. fc_credit_init Register 0x1c8

FC Credit Init Control.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	redo	read-write	1	0x0

redo

Force the core to redo FC Credit Initialization without taking the link down. This is only possible if both ends of the link are instructed to redo the the initialization.

2.4.1.3.15. rx c Register 0x200

Receive Buffer aribtration and completion handling configuration.

Field	Name		Width	Reset
31..24	priority p starve thresh	read-write	8	10h
23..16	priority n starve thresh	read-write	8	10h
15..2	*reserved*	read only	14	0h
1	force_ro	read-write	1	0h
0	priority	read-write	1	0h

priority p starve thresh when priority == 1, priority p starve thresh == number of times a P TLP grant can be skipped before P priority is elevated to prevent it from starving.

priority n starve thresh when priority == 1, priority n starve thresh == number of times a N TLP grant can be skipped before N priority is elevated to prevent it from starving.

force ro

Force completion relaxed ordering (RO==1) behavior for all completion TLPs, even those with RO==0. Note that setting this register to 1 is not PCIe Spec. compliant but this may be ne for some designs since it is acceptable in many designs for C without RO==1 to pass prior P.

- 0 – Disable – Received completions are handled using their received RO attribute.
- 1 – Enable – All received completions are handled as if the RO attribute was 1.

priority

Completion priority enable.

- 0 – Disable. Arbitration between Posted, Non-Posted, and Completion TLPs is round robin.
- 1 – Enable. While arbitrating between putting pending received Posted, Non-Posted, and Completion TLPs on the user received TLP interface, completions are given highest priority. Posted and non-posted requests will transact only when a completion is not pending or as needed to prevent starving.

2.4.1.3.16. rx ctrl Register 0x208

Receive Buffer Control

Field	Name	Access	Width	Reset
31..23	*reserved*	read only	9	0h
22..20	max pl size supported max	read only	3	0h
19..16	*reserved*	read only	4	0h
15..10	*reserved*	read only	6	0h
9..8	adv ch cd sel	read-write	2	0h
7..3	*reserved*	read only	5	0h
2..1	fc update timer div	read-write	2	0h
0	fc update timer disable	read-write	1	0h

max pl size supported max

Maximum value recommended for max pl size supported. This status field is set to the TLP payload size that can be held by the lesser of 1/2 of the P or C TLP buffers and assumes that at least 1/2 of the P & C TLP buffer space is reserved to hold data credits (with the remaining space reserved to hold TLP headers). However, for sustained throughput performance it is better to configure the Rx Buffer to hold at least 3-4 max payload size TLPs in each of the P & C buffers.

- 0 – 128 Bytes
- 1 – 256 Bytes
- 2 – 512 Bytes
- 3 – 1024 Bytes
- 4 – 2048 Bytes
- 5 – 4096 Bytes
- 6 – Reserved
- 7 – Reserved

adv ch cd sel

PCIe Spec. requires CH & CD credit advertisements to be infinite for Endpoints and the finite (actual credit values) for Root Port and Switch Ports. ch cd sel may be configured to over-ride the default PCIe Spec. expected behavior.

- 0 – Implement CH, CD credit advertisements per port type: Endpoints == infinite, Root Port and Switch Ports == actual.
- 1 – Advertise actual CH, CD credits.
- 2 – Advertise Infinite CH, CD credits.

fc update timer div

Receive Buffer Flow Control Divider. Configures the FC Update frequency of the Receive Buffer when fc update timer disable==0.

- 0 – Use the PCIe Spec. recommended values
- 1 – Use the PCIe Spec. recommended values divided by 2
- 2 – Use the PCIe Spec. recommended values divided by 4
- 3 – Use the PCIe Spec. recommended values divided by 8

fc update timer disable

Receive Buffer Flow Control Disable

- 0 – Enable the FC Update Timer – schedule FC Updates in accordance with PCIe. Spec. recommended values.
- 1 – Disable FC Update Timer – schedule a FC Update on Every Consumed RX TLP.

2.4.1.3.17. p stat rx Register 0x210

Receive Buffer PCIe Clock Domain Input Status.

Field	Name	Access	Width	Reset
31..2	*reserved*	read only	30	0h
1	mps violation	read-write	1	0h
0	tlp valid	read-write	1	0h

mps violation

Receive Buffer discarded a TLP on PCIe clock domain at the input of the Receive Buffer for exceeding the Max Payload Size. Sticky, write 1 to clear. This field is referenced to clock p clk.

tlp valid

Receive Buffer received a valid TLP on PCIe clock domain at input of the Receive Buffer. Sticky, write 1 to clear. This field is referenced to clock p clk.

2.4.1.3.18. u stat rx Register 0x214

Receive Buffer User Clock Domain Output Status.

Field	Name	Access	Width	Reset
31..2	*reserved*	read only	28	0x0
1	err ucor	read-write	1	0x0
0	err cor	read-write	1	0x0

err ucor

Uncorrectable ECC error detected at output of Receive Buffer. Sticky, write 1 to clear. This field is referenced to clock u clk.

err cor

Correctable ECC error detected at output of Receive Buffer. Sticky, write 1 to clear. This field is referenced to clock u clk.

2.4.1.3.19. vc rx control Register 0x218

Receive Buffer Parity/ECC Control.

Field	Name	Access	Width	Reset
31..19	*reserved*	read only	13	0h
18	par report disable	read-write	1	0h
17	*reserved*	read only	1	0h
16	par inject en	read-write	1	0h
15	*reserved*	read only	1	0h
14	ecc2 report disable	read-write	1	0h
13	ecc2 handle disable	read-write	1	0h
12	ecc2 inject m 1 n	read-write	1	0h
11..9	ecc2 inject type	read-write	3	0h
8	ecc2 inject en	read-write	1	0h
7	*reserved*	read only	1	0h
6	ecc1 report disable	read-write	1	0h
5	ecc1 handle disable	read-write	1	0h
4	ecc1 inject m 1 n	read-write	1	0h
3..1	ecc1 inject type	read-write	3	0h
0	ecc1 inject en	read-write	1	0h

par report disable

Receive Buffer Parity Error Reporting Disable. This field is referenced to clock u clk.

- 0 – Enable reporting.
- 1 – Disable reporting of Receive Buffer detected parity errors.

par inject en

Receive Buffer Parity Error Injection Enable. This field is referenced to clock p clk.

- 0 – Do not inject error.
- 1 – When par inject en is written from 0 to 1, a parity error injection is scheduled and will be injected at the next opportunity (TLP receipt). The injection inverts the parity of TLP header/payload bytes being received in the clock cycle that the injection is performed.

ecc2 report disable

Receive Buffer ECC 2-bit Error Reporting Disable. This field is referenced to clock u clk.

- 0 – Enable reporting.
- 1 – Disable reporting of ECC 2-bit errors.

ecc2 handle disable

Receive Buffer ECC 2-bit Error Handling Disable. This field is referenced to clock u clk.

- 0 – Enable handling.
- 1 – Disable hanlding of ECC 2-bit errors.

ecc2 inject m 1 n

Receive Buffer ECC 2-bit Error Injection Multiple/Single Select. This field is referenced to clock p clk.

- 0 – Inject only 1 error when ecc2 inject en is written from 0 to 1.
- 1 – Keep injecting errors as long as ecc2 inject en == 1

ecc2 inject type

Receive Buffer ECC 2-bit Error Injection Type. This field is referenced to clock p clk.

- 0 – Inject error in Posted Data RAM
- 1 – Inject error in Non-posted Data RAM
- 2 – Inject error in Completion Data RAM
- 3 – Reserved. Do not use.
- 4 – Inject error in Posted Header RAM
- 5 – Inject error in Non-posted Header RAM
- 6 – Inject error in Completion Header RAM
- 7 – Reserved. Do not use.

ecc2 inject en

Receive Buffer ECC 2-bit Error Injection Enable. ecc2 inject en must not be written in the same write that changes the value of ecc2 inject m 1 n or ecc2 inject type. This field is referenced to clock p clk.

- 0 – Do not inject error.
- 1 – Inject ECC 2-bit error at the next opportunity (Receive Buffer RAM write).

ecc1 report disable

Receive Buffer ECC 1-bit Error Reporting Disable. This field is referenced to clock u clk.

- 0 – Enable reporting.
- 1 – Disable reporting of ECC 1-bit errors.

ecc1 handle disable

Receive Buffer ECC 1-bit Error Handling Disable. This field is referenced to clock u clk.

- 0 – Enable correction.
- 1 – Disable correction of ECC 1-bit errors. When error correction is disabled, ECC 1-bit errors are treated the same as uncorrectable ECC 2-bit errors.

ecc1 inject m 1 n

Receive Buffer ECC 1-bit Error Injection Multiple/Single Select. This field is referenced to clock p clk.

- 0 – Inject only 1 error when ecc1 inject en is written from 0 to 1.
- 1 – Keep injecting errors as long as ecc1 inject en == 1

ecc1 inject type

Receive Buffer ECC 1-bit Error Injection Type. This field is referenced to clock p clk.

- 0 – Inject error in Posted Data RAM
- 1 – Inject error in Non-posted Data RAM
- 2 – Inject error in Completion Data RAM
- 3 – Reserved. Do not use.
- 4 – Inject error in Posted Header RAM
- 5 – Inject error in Non-posted Header RAM
- 6 – Inject error in Completion Header RAM
- 7 – Reserved. Do not use.

ecc1 inject en

Receive Buffer ECC 1-bit Error Injection Enable. ecc1 inject en must not be written in the same write that changes the value of ecc1 inject m 1 n or ecc1 inject type. This field is referenced to clock p clk.

- 0 – Do not inject error.
- 1 – Inject ECC 1-bit error at the next opportunity (Receive Buffer RAM write).

2.4.1.3.20. vc rx status Register 0x21c

Receive Buffer Parity/ECC Status.

Field	Name	Access	Width	Reset
31..4	*reserved*	read only	28	0h
3	*reserved*	read only	1	0h
2	err par	read-write	1	0h
1	err ecc2	read-write	1	0h
0	err ecc1	read-write	1	0h

err par

Receive Buffer Parity Error Detection Status. This field is referenced to clock u clk.

- 0 – Otherwise.
- 1 – Error occurred.

err ecc2

Receive Buffer ECC 2-bit Error Detection Status. This field is referenced to clock u clk.

- 0 – Otherwise.
- 1 – Error occurred.

err ecc1

Receive Buffer ECC 1-bit Error Detection Status. This field is referenced to clock u clk.

- 0 – Otherwise.
- 1 – Error occurred.

2.4.1.3.21. u rx credit stat p init Register 0x220

Receive Buffer Posted Credit Initialization Status.

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	4	0h
27..16	h	read only	12	0h
15..0	d	read only	16	0h

h

Number of PH credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

d

Number of PD credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

2.4.1.3.22. u rx credit stat p curr Register 0x224

Receive Buffer Posted Credit Current Status.

Field	Name	Access	Width	Reset
31	lim h	read-write	1	0h
30	lim d	read-write	1	0h
29..28	*reserved*	read only	2	0h
27..16	h	read only	12	0h
15..0	d	read only	16	0h

lim h

Receive Buffer PH limited status. 1==Forwarding of TLPs was limited due to PH credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

lim d

Receive Buffer PD limited status. 1==Forwarding of TLPs was limited due to PD credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

h

Receive Buffer – Current PH credits. This field is referenced to clock u clk.

d

Receive Buffer – Current PD credits. This field is referenced to clock u clk.

2.4.1.3.23. u rx credit stat n init Register 0x228

Receive Buffer FC Update Timer Control

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	4	0h
27..16	h	read only	12	0h
15..0	d	read only	16	0h

h

Number of NH credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

d

Number of ND credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

2.4.1.3.24. u rx credit stat n curr Register 0x22c

Receive Buffer Non-Posted Credit Current Status.

Field	Name	Access	Width	Reset
31	lim h	read-write	6	0x0
30	lim d	read-write	10	0x10
29..28	*reserved*	read only	7	0x0
27..16	h	read only	1	0x0
15..0	d	read only	7	0x0

lim h

Receive Buffer NH limited status. 1==Forwarding of TLPs was limited due to NH credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

lim d

Receive Buffer ND limited status. 1==Forwarding of TLPs was limited due to ND credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

h

Receive Buffer – Current NH credits. This field is referenced to clock u clk.

d

Receive Buffer – Current ND credits. This field is referenced to clock u clk.

2.4.1.3.25. u rx credit stat c init Register 0x230

Receive Buffer Completion Credit Initialization Status.

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	4	0h
27..16	h	read only	12	0h
15..0	d	read only	16	0h

h

Number of CH credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

d

Number of CD credits with which Receive Buffer was initialized. This field is referenced to clock u clk.

2.4.1.3.26. u rx credit stat c curr Register 0x234

Receive Buffer Completion Credit Current Status.

Field	Name	Access	Width	Reset
31	lim h	read-write	1	0h
30	lim d	read-write	1	0h
29..28	*reserved*	read only	2	0h
27..16	h	read only	12	0h
15..0	d	read only	16	0h

lim h

Receive Buffer CH limited status. 1==Forwarding of TLPs was limited due to CH credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

lim d

Receive Buffer CD limited status. 1==Forwarding of TLPs was limited due to CD credits. 0==Otherwise. Sticky, write 1 to clear. This field is referenced to clock u clk.

h

Receive Buffer - Current CH credits. This field is referenced to clock u clk.

d

Receive Buffer - Current CD credits. This field is referenced to clock u clk.

2.4.1.3.27. rx alloc size p Register 0x240

Receive Buffer Posted Storage Status.

Field	Name	Access	Width	Reset
31..24	hdr	read only	8	0h
23..0	storage	read only	24	0h

hdr

Number of bytes required to store 1 PH credit.

storage

Receive Buffer P RAM storage size in bytes.

2.4.1.3.28. rx alloc size n Register 0x244

Receive Buffer Non-Posted Storage Status.

Field	Name	Access	Width	Reset
31..24	hdr	read only	8	0h
23..0	storage	read only	24	0h

Hdr

Number of bytes required to store 1 NH credit.

Storage

Receive Buffer N RAM storage size in bytes.

2.4.1.3.29. rx alloc size c Register 0x248

Receive Buffer Completion Storage Status.

Field	Name	Access	Width	Reset
31..24	hdr	read only	8	0h
23..0	storage	read only	24	0h

d hdr

Number of bytes required to store 1 CH credit.

Storage

Receive Buffer C RAM storage size in bytes.

2.4.1.3.30. rx alloc lim Register 0x24c

Receive Buffer Allocation Limit Status.

Field	Name	Access	Width	Reset
31..24	max ch	read only	8	0h
23..16	max nh	read only	8	0h
15..8	max ph	read only	8	0h
7..0	min d multiple	read only	8	0h

max ch

Receive Buffer maximum number of CH credits which may be allocated == $2^{\text{mgmt ptl rx alloc lim max ch}}$.

max nh

Receive Buffer maximum number of NH credits which may be allocated == $2^{\text{mgmt ptl rx alloc lim max nh}}$.

max ph

Receive Buffer maximum number of PH credits which may be allocated == $2^{\text{mgmt ptl rx alloc lim max ph}}$.

min d multiple

Receive Buffer minimum multiple of D credits that can be allocated. <= 128-bit cores must allocate D credits in multiples of 1, 256-bit cores in multiples of 2, and 512-bit cores in in multiples of 4.

2.4.1.3.31. rx alloc p Register 0x250

Receive Buffer Posted Credit Allocation.

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	4	0h
27..16	h	read-write	12	10h
15..0	d	read-write	16	6ch

h

Number of PH credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space.

d

Number of PD credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested PH & PD credits must not exceed the P RAM storage space.

2.4.1.3.32. rx alloc n Register 0x254

Receive Buffer Non-Posted Credit Allocation.

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	31	0x0
27..16	h	read-write	1	0x0
15..0	d	read-write		

h

Number of NH credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space.

d

Number of ND credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested NH & ND credits must not exceed the N RAM storage space.

2.4.1.3.33. rx alloc c Register 0x258

Receive Buffer Completion Credit Allocation.

Field	Name	Access	Width	Reset
31..28	*reserved*	read only	4	0h
27..16	h	read-write	12	20h
15..0	d	read-write	16	60h

h

Number of CH credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space.

d

Number of CD credits to allocate to the Receive Buffer. The number of bytes required to allocate the requested CH & CD credits must not exceed the C RAM storage space.

2.4.1.3.34. rx alloc sel Register 0x25c

Receive Buffer Credit Allocation Selection.

Field	Name	Access	Width	Reset
31..1	*reserved*	read only	31	0h
0	en	read-write	1	0h

en

Receive Buffer credit allocation selection.

- 0 – Credits are allocated by the hardware design. Credits are allocated in a balanced fash- ion using all available RAM. For designs supporting bifurcation, the hardware credit allocation automatically adapts to the current bifurcation.
- 10 – Allocate credits under user control using the rx alloc p/n/ch and rx alloc p/n/cd registers

2.4.1.3.35. rx alloc error Register 0x260

Receive Buffer Allocation Error Status.

Field	Name	Access	Width	Reset
31..24	*reserved*	read only	8	0h
23	*reserved*	read only	1	0h
22	max ch	read only	1	0h
21	max nh	read only	1	0h
20	max ph	read only	1	0h
19..16	*reserved*	read only	4	0h
15	*reserved*	read only	1	0h
14	c sum	read only	1	0h
13	n sum	read only	1	0h
12	p sum	read only	1	0h
11	*reserved*	read only	1	0h
10	cd mult	read only	1	0h
9	nd mult	read only	1	0h
8	pd mult	read only	1	0h
7	*reserved*	read only	1	0h
6	min cd	read only	1	0h
5	min nd	read only	1	0h
4	min pd	read only	1	0h
3	*reserved*	read only	1	0h
2	min ch	read only	1	0h
1	min nh	read only	1	0h
0	min ph	read only	1	0h

max ch

mgmt ptl rx alloc ch must be <= $(2^{\text{mgmt ptl rx alloc lim max ch}})$.

- 0 – No Error
- 1 – Error

max nh

mgmt ptl rx alloc nh must be <= $(2^{\text{mgmt ptl rx alloc lim max nh}})$.

- 0 – No Error
- 1 – Error

max ph

mgmt ptl rx alloc ph must be <= $(2^{\text{mgmt ptl rx alloc lim max ph}})$.

- 0 – No Error
- 1 – Error

c sum

Storage space required for mgmt ptl rx alloc ch + mgmt ptl rx alloc cd exceeds C RAM storage space.

- 0 – No Error
- 1 – Error

n sum

Storage space required for mgmt ptl rx alloc nh + mgmt ptl rx alloc nd exceeds N RAM storage space.

- 0 – No Error
- 1 – Error

p sum

Storage space required for mgmt ptl rx alloc ph + mgmt ptl rx alloc pd exceeds P RAM storage space.

- 0 – No Error
- 1 – Error

cd mult

mgmt ptl rx alloc cd[1:0] must be 0 for 512-bit and mgmt ptl rx alloc cd[0] must be 0 for 256-bit.

- 0 – No Error
- 1 – Error

nd mult

mgmt ptl rx alloc nd[1:0] must be 0 for 512-bit and mgmt ptl rx alloc nd[0] must be 0 for 256-bit.

- 0 – No Error
- 1 – Error

pd mult

mgmt ptl rx alloc pd[1:0] must be 0 for 512-bit and mgmt ptl rx alloc pd[0] must be 0 for 256-bit.

- 0 – No Error
- 1 – Error

min cd

mgmt ptl rx alloc cd must be >= Max Payload Size Supported.

- 0 – No Error
- 1 – Error

min nd

mgmt ptl rx alloc nd must be >= 2.

- 0 – No Error
- 1 – Error

min pd

mgmt ptl rx alloc pd must be >= Max Payload Size Supported.

- 0 – No Error
- 1 – Error

min ch

mgmt ptl rx alloc ch must be > 0.

- 0 – No Error
- 1 – Error

min nh

mgmt ptl rx alloc nh must be > 0.

- 0 – No Error
- 1 – Error

min ph

mgmt ptl rx alloc ph must be > 0.

- 0 – No Error
- 1 – Error

2.4.1.3.36. tx c Register 0x280

Transmit Buffer arbitration and completion handling configuration.

Field	Name	Access	Width	Reset
31..24	priority p starve thresh	read-write	8	10h
23..16	priority n starve thresh	read-write	8	10h
15..2	*reserved*	read only	14	0h
1	force ro	read-write	1	0h
0	priority	read-write	1	0h

priority p starve thresh

when priority == 1, priority p starve thresh == number of times a P TLP grant can be skipped before P priority is elevated to prevent it from starving.

priority n starve thresh

when priority == 1, priority n starve thresh == number of times a N TLP grant can be skipped before N priority is elevated to prevent it from starving.

force ro

Force completion relaxed ordering (RO==1) behavior for all completion TLPs, even those with RO==0. Note that setting this register to 1 is not PCIe Spec. compliant but this may be fine for some designs since it is acceptable in many designs for C without RO==1 to pass prior P.

- 0 – Disable - transmitted completions are handled using their transmitted RO attribute.
- 1 – Enable - all transmitted completions are handled as if the RO attribute was 1.

priority

Completion priority enable.

- 0 – Disable. Arbitration between Posted, Non-Posted, and Completion TLPs is round robin.
- 1 – Enable. While arbitrating between putting pending Transmitted Posted, Non-Posted, and Completion TLPs on the user Transmitted TLP interface, completions are given highest priority. Posted and non-posted requests will transact only when a completion is not pending or as needed to prevent starving.

2.4.1.3.37. tx ctrl Register 0x284

Transmit Buffer Control.

Field	Name	Access	Width	Reset
31..23	*reserved*	read only	9	0h
22..20	max pl size supported max	read only	3	0h
19..16	*reserved*	read only	4	0h
15..8	*reserved*	read only	8	0h
7..0	*reserved*	read only	8	0h

max pl size supported max

Maximum value recommended for max pl size supported due to Tx Buffer size restrictions. This status field is set to the TLP payload size that can be held by the lesser of 1/2 of the P or C TLP buffers and assumes that at least 1/2 of the P & C TLP buffer space is reserved to hold data credits (with the remaining space reserved to hold TLP headers). However, for sustained throughput performance it is better to configure the Rx Buffer to hold at least 3-4 max payload size TLPs in each of the P & C buffers.

- 0 – 128 Bytes
- 1 – 256 Bytes
- 2 – 512 Bytes
- 3 – 1024 Bytes
- 4 – 2048 Bytes
- 5 – 4096 Bytes
- 6 – Reserved
- 7 – Reserved

2.4.1.3.38. vc tx credit cleanup Register 0x288

TLP transmit error credit cleanup control.

Field	Name	Access	Width	Reset
31..1	reserved	read only	31	0h
0	method	read-write	1	0h

method

TLP Transmit Credit Cleanup Method.

- 0 – Use the headers of the cleaned-up TLPs to recover the credits. The credits in TLPs with corrupted headers will not be recovered.
- 1 – Use a credit lookup table based on the ID assigned to the TLP. This table is implemented in pcie user if.

2.4.1.4. mgmt_ftl (0x04000) (Function 0)

mgmt_ftl_BASE 0x4000

2.4.1.4.1. simulation Register 0x0

For simulation only: Full Transaction Layer simulation speed reduction.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	reduce_timeouts	read-write	1	0x0

reduce_timeouts

Reduce timeouts to shorten simulation time. When enabled ms timeouts are shortened to their value in us.

- 0 – Disable
- 1 – Enable

2.4.1.4.2. decode Register 0x10

Transaction Layer Decode Configuration.

Field	Name	Access	Width	Reset
[31:24]	*reserved*	read-only	8	0x0
[23:18]	*reserved*	read-only	6	0x0
[17]	tx_bypass_decode_en	read-write	1	0x0
[16]	rx_bypass_decode_en	read-write	1	0x0
[15:11]	*reserved*	read-only	5	0x0
[10]	tx_convert_ur_to_ca	read-write	1	0x0
[9]	rx_convert_ur_to_ca	read-write	1	0x0
[8]	t0_rx_bypass_msg_dec	read-write	1	0x0
[7:3]	*reserved*	read-only	5	0x0
[2]	vendor0_ur	read-write	1	0x1
[1]	target_only	read-write	1	0x0
[0]	ignore_poison	read-write	1	0x1

tx_bypass_decode_en

Bypass the TLP decode block in the Transmit path.

- 0 – Decode_in_path module is enabled.
- 1 – Decode_in_path module is bypassed.

rx_bypass_decode_en

Bypass the TLP decode block in the Receive path.

- 0 – Decode_in_path module is enabled.
- 1 – Decode_in_path module is bypassed.

tx_convert_ur_to_ca

When decoding TX packets convert Unsupported Request (UR) packets to Completer Abort (CA).

- 0 – Normal Operation.
- 1 – Convert UR to CA.

rx_convert_ur_to_ca

When decoding RX packets convert Unsupported Request (UR) packets to Completer Abort (CA).

- 0 – Normal Operation.
- 1 – Convert UR to CA.

t0_rx_bypass_msg_dec

When implementing Type 0 Configuration Space (Endpoint) – Bypass RX Message TLP Decode Enable.

- 0 – Normal operation. The core claims and does not forward Mesage TLPs to the TLP Receive Interface. Received messages may still be snooped by monitoring the Message Interface for the relevant message types.
- 1 – All valid Msg TLPs received on PCIe (except Routed by ID & Routed by Address which are routed according to their routing type) are forwarded to the TLP Receive Interface.

vendor0_ur

Vendor Type 0 Messages received from PCIe are reported as UR.

- 0 – Do not report received Vendor Type 0 Messages as Unsupported Request (UR).
- 1 – Report received Vendor Type 0 Messages as Unsupported Request (UR).

target_only

Target Only. Enable for user designs that implement purely target-only functionality. When enabled all received completions are considered Unexpected Completions and are not forwarded to the TLP Receive Interface.

- 0 – Disable
- 1 – Enable

ignore_poison

Ignore Poison – Set to 1 to have the core ignore the EP poison indicator for received TLPs with data payload that do not terminate in the core. When set to 1, the core passes all poisoned TLPs to you the same way it would pass the TLP if the TLP was not poisoned. Note that the Ignore Poison control is forced to 1 by the core when the core is configured as a Root-Port.

Note that the following TLP types ignore the setting of this bit.

Poisoned Configuration Type 0 writes are terminated in the core in all cases, independent of the Ignore Poison bit setting. A completion with UR status is generated and the appropriate error message, ERR COR or ERR FAT, is generated if not masked. Note that Poisoned Configuration Type 0 reads are always treated as if they were not poisoned. The read completes with successful completion status and an optional Advisory Non-Fatal Error status is set provided the severity level is set to NON FATAL. Poisoned packets without data payload are passed to you in all cases since EP should not be set on packets without data payload and these packets should generally be handled as if they were not poisoned or alternatively handled as Advisory Non-Fatal Errors by user logic. Poisoned Vendor-defined Type 1 messages with data payload are always passed to you and, if ignore poison is 0, additionally an Advisory Non-Fatal Error status is set provided the severity level is set to NON FATAL.

When Ignore Poison is set to 0, the core handles the remaining poisoned TLPs with data payload as follows.

Poisoned Write request and poisoned read completions with data TLPs are consumed by the core and handled as TLP Poisoned errors that generate the appropriate poison, ERR NON FATAL or ERR FATAL, depending upon the error severity register error message. Poisoned Message with data payload (other than vendor-defined type 1) are consumed by the core and handled as TLP Poisoned errors that generate the appropriate posion, ERR NON FATAL or ERR FATAL depending upon the error severity register, error message.

The recommended default for target-only endpoints is to set Ignore Poison == 0 and to have user logic ignore the EP header bit on TLPs that it receives. In this case, poisoned TLPs with data payload (other than config 0 writes and vendor-defined type 1 messages) generate a NON FATAL error message and discarded by the core. Poisoned TLPs without data payload (for which EP does not apply) are processed as if they were not poisoned.

- 0 – Disable
- 1 – Enable

2.4.1.4.3. decode_t1 Register 0x14

Type 1 Configuration Space Transaction Layer Decode Configuration.

Field	Name	Access	Width	Reset
[31:24]	*reserved*	read-only	8	0x0
[23:16]	*reserved*	read-only	8	0x0
[15:11]	*reserved*	read-only	5	0x0
[10]	bypass_addr_dec	read-write	1	0x0
[9]	tx_bypass_msg_dec	read-write	1	0x0
[8]	rx_bypass_msg_dec	read-write	1	0x0
[7:0]	*reserved*	read-only	8	0x0

bypass_addr_dec

When implementing Type 1 Configuration Space (Root Port): Bypass TLP Address Decode Enable. bypass_addr_dec controls the decoding of received Memory and I/O Request TLPs on both the Primary and Secondary sides of the core.

- 0 – Normal PCI Express compliant address validity checks are performed. Transactions must target a valid region to be forwarded. Transactions with an invalid address cause an error response to be generated. Recommended value unless the customer application requires a value of 1 to be used.)
- 1 – The address validity checks for IO, Memory, and Prefetchable Memory regions are bypassed. Memory Requests, I/O Requests, and Messages Routed by Address are considered valid regardless of address. bypass_addr_dec does not affect the other validity checks (Memory/IO regions enabled, in a Power State where transactions can be accepted, link is up, not poisoned, etc.). Transactions which fail these other validity checks still cause an error response to be generated. When bypass_addr_dec == 1, Memory Requests, which appear on the Receive Interface indicates a hit to the Prefetchable Memory Window and I/O Requests which appear on the Receive Interface indicate a hit to the I/O Window.

tx_bypass_msg_dec

When implementing Type 1 Configuration Space (Root Port): Bypass TX Message TLP Decode Enable.

- 0 – Normal operation. Claim and do not forward Local Routing messages and Error Messages that are transmitted without error propagation being enabled. Other message types are left in the stream.
- 1 – All valid Message TLPs transmitted towards PCIe (except Routed by ID and Routed by Address which are routed according to their routing type) are forwarded to PCIe.

rx_bypass_msg_dec

When implementing Type 1 Configuration Space (Root Port): Bypass RX Message TLP Decode Enable.

- 0 – Normal operation. Claim and do not forward Local Routing messages and Error Messages that are transmitted without error propagation being enabled. Other message types are left in the stream.
- 1 – All valid Message TLPs received from PCIe (except Routed by ID & Routed by Address which are routed according to their routing type) are forwarded to the TLP Receive Interface.

2.4.1.4.4. tlp_processing Register 0x18

Transaction Layer TLP Processing Configuration.

Field	Name	Access	Width	Reset
[31:24]	*reserved*	read-only	8	0x0
[23:16]	*reserved*	read-only	8	0x0
[15:8]	*reserved*	read-only	8	0x0
[7:2]	*reserved*	read-only	6	0x0
[1]	ignore_ecrc	read-write	1	0x0
[0]	crs_enable	read-write	1	0x0

ignore_ecrc

Ignore ECRC Error Enable. When enabled ECRC errors are ignored for TLPs passed to you on the TLP Receive Interface.

- 0 – Disable
- 1 – Enable

crs_enable

Configuration Request Retry Status Enable.

- 0 – Disable. Type 0 Configuration Writes and Reads are performed normally.
- 1 – Enable. Type 0 Configuration Writes and Reads return Configuration Request Retry Status.

2.4.1.4.5. initial Register 0x20

Initial speed and width configuration.

Field	Name	Access	Width	Reset
[31:19]	*reserved*	read-only	13	0x0
[18:16]	max_link_width	read-write	3	0x1
[15:2]	*reserved*	read-only	14	0x0
[1:0]	target_link_speed	read-write	2	0x3

max_link_width

Max Link Width Override. This setting if different from zero, overrides the value of Maximum Link Width in the PCIe Link Capabilities register.

- 0 – Maximum core lane width
- 1 – 1 lane

target_link_speed

Initial value of Target Link Speed Configuration Register. Determines the maximum initial link speed which can be reached during initial training. Must be set to the lesser of the maximum speed supported by the core and the maximum speed at which you want the core to operate.

- 0 – 2.5G
- 1 – 5.0G

2.4.1.4.6. cfg Register 0x30

Configuration Register type.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	type1_type0_n	read-write	1	0x0

type1_type0_n

Determines the type of Configuration Registers implemented by the core.

- 0 – Type 0 – Endpoint
- 1 – Type 1 – Root

2.4.1.4.7. ds_port Register 0x34

Downstream Port configuration.

Field	Name	Access	Width	Reset
[31:17]	*reserved*	read-only	15	0x0
[16]	rcb	read-write	1	0x0
[15:0]	id	read-write	16	0x0

rcb

Read Completion Boundary (RCB). RCB value advertised when the core is operating as a Root Port.

id

Root Port ID. This 16 bit field is used to define the ID used for PCIe Requester ID and Completer ID when the core is operating as a Root Port.

2.4.1.4.8. us_port Register 0x38

Upstream Port Configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	adv_target_link_speed	read-write	1	0x0

adv_target_link_speed

For an upstrea port, advertise the link speeds specified by the target_link_speed field rather than the maximum supported speed.

2.4.1.4.9. id1 Register 0x40

ID1 configuration.

Field	Name	Access	Width	Reset
[31:16]	device_id	read-write	16	0xe004
[15:0]	vendor_id	read-write	16	0x19aa

device_id

Value returned when the Device ID Configuration Register is read.

vendor_id

Value returned when the Vendor ID Configuration Register is read.

2.4.1.4.10. id2 Register 0x44

ID2 configuration.

Field	Name	Access	Width	Reset
[31:16]	subsystem_id	read-write	16	0xe004
[15:0]	subsystem_vendor_id	read-write	16	0x19aa

subsystem_id

Value returned when the Subsystem ID Configuration Register is read.

subsystem_vendor_id

Value returned when the Subsystem Vendor ID Configuration Register is read.

2.4.1.4.11. id3 Register 0x48

ID3 configuration.

Field	Name	Access	Width	Reset
[31:8]	class_code	read-write	24	0x118000
[7:0]	revision_id	read-write	8	0x4

class_code

Value returned when the Class Code Configuration Register is read. Must be set to the correct value for the type of device being implemented; see PCI Local Bus Specification Revision 2.3 Appendix D for details on setting Class Code.

revision_id

Value returned when the Revision ID Configuration Register is read.

2.4.1.4.12. cardbus Register 0x4c

Cardbus configuration.

Field	Name	Access	Width	Reset
[31:0]	cis_pointer	read-write	32	0x0

cis_pointer

Value returned when the Cardbus CIS Pointer Configuration Register is read. Set to 0x00000000 unless a Cardbus CIS structure is implemented in memory (which is rare), in which case set to the address of the CIS Structure.

2.4.1.4.13. interrupt Register 0x50

Interrupt configuration.

Field	Name	Access	Width	Reset
[31:10]	*reserved*	read-only	22	0x0
[9:8]	pin	read-write	2	0x0
[7:1]	*reserved*	read-only	7	0x0
[0]	disable	read-write	1	0x0

pin

Selects which legacy interrupt is used.

- 0 – INTA
- 1 – INTB
- 2 – INTC
- 3 – INTD

disable

Disable support for interrupts.

- 0 – Enable
- 1 – Disable

2.4.1.4.14. bar0 Register 0x60

BAR0 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffff000c

cfg

Configuration of BAR0 (Cfg address 0x10). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.15. bar1 Register 0x64

BAR1 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar1 (Cfg address 0x14). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.16. bar2 Register 0x68

BAR2 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffe00c

cfg

Configuration of bar2 (Cfg address 0x18). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.17. bar3 Register 0x6c

BAR3 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar3 (Cfg address 0x1C). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.18. bar4 Register 0x70

BAR4 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffe00c

cfg

Configuration of bar4 (Cfg address 0x20). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.19. bar5 Register 0x74

BAR5 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar5 (Cfg address 0x24). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.4.20. exp_rom Register 0x78

Expansion ROM configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0x0

cfg

Configuration of exp_rom. Use to define a 32-bit Memory Expansion ROM region. If an Expansion ROM region is defined, then the region must map to PCIe-compliant Expansion ROM code or the device may fail to boot.

2.4.1.4.21. pcie_cap Register 0x80

PCI Express Capbabilities configuration.

Field	Name	Access	Width	Reset
[31:14]	*reserved*	read-only	18	0x0
[13:9]	interrupt_message_number	read-write	5	0x0
[8]	slot_implemented	read-write	1	0x0
[7:4]	device_port_type	read-write	4	0x0
[3:0]	capability_version	read-write	4	0x2

interrupt_message_number

MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of the PCI Express Capability structure.

slot_implemented

Indicates that the Link associated with this Port is connected to a slot. This field is valid for Downstream Ports only.

device_port_type

Indicates the specific type of this PCI Express Function.

- 0 – PCI Express Endpoint
- 1 – Legacy PCI Express Endpoint
- 2 – Reserved
- 3 – Reserved
- 4 – Root Port of PCI Express Root Complex
- 5 – Upstream Port of PCI Express Switch
- 6 – Downstream Port of PCI Express Switch
- 7 – PCI Express to PCI/PCI-X Bridge
- 8 – PCI/PCI-X to PCI Express Bridge
- 9 – Root Complex Integrated Endpoint
- 10 – Root Complex Event Collector
- 11 – Reserved
- 12 – Reserved
- 13 – Reserved
- 14 – Reserved
- 15 – Reserved

capability_version

Indicates PCI-SIG defined PCI Express Capability structure version number. Must be set to 0x2.

2.4.1.4.22. pcie_dev_cap Register 0x84

PCI Express Device Capbabilities configuration.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28]	disable_flr_capability	read-write	1	0x0
[27:26]	*reserved*	read-only	2	0x0
[25:18]	*reserved*	read-only	8	0x0
[17:16]	*reserved*	read-only	2	0x0
[15]	*reserved*	read-only	1	0x0
[14:13]	*reserved*	read-only	2	0x0
[12]	extended_tag_field_en_default	read-write	1	0x1
[11:9]	endpoint_l1_acceptable_latency	read-write	3	0x0
[8:6]	endpoint_l0s_acceptable_latency	read-write	3	0x0
[5]	extended_tag_field_supported	read-write	1	0x1
[4:3]	phantom_functions_supported	read-write	2	0x0
[2:0]	max_payload_size_supported	read-write	3	0x2

disable_flr_capability

Function Level Reset Capability

- 0 – Enable
- 1 – Disable

extended_tag_field_en_default

Extended Tag Field Enable Default Value. PCIe Spec. allows the Extended Tag Field Enable register to reset to either 1 or 0. This register determines the reset value.

- 0 – 5-bit Tag field enabled on reset
- 1 – 8-bit Tag field enabled on reset

endpoint_l1_acceptable_latency

Endpoint L1 Acceptable Latency

- 0 – Maximum of 1 us. Must be 0 when not an Endpoint.
- 1 – Maximum of 2 us
- 2 – Maximum of 4 us
- 3 – Maximum of 8 us
- 4 – Maximum of 16 us
- 5 – Maximum of 32 us
- 6 – Maximum of 64 us
- 7 – No limit

endpoint_l0s_acceptable_latency

Endpoint L0s Acceptable Latency

- 0 – Maximum of 64 ns. Must be 0 when not an Endpoint.
- 1 – Maximum of 128 ns
- 2 – Maximum of 256 ns
- 3 – Maximum of 512 ns
- 4 – Maximum of 1 us
- 5 – Maximum of 2 us
- 6 – Maximum of 4 us
- 7 – No limit

extended_tag_field_supported

Extended Tag Field Supported

- 0 – 5-bit Tag field supported
- 1 – 8-bit Tag field supported

phantom_functions_supported

Phantom Functions Supported

- 0 – No Function Number bits are used for Phantom Functions
- 1 – The most significant bit of the Function number in Requester ID is used for Phantom Functions
- 2 – The two most significant bits of Function Number in Requester ID are used for Phantom Functions
- 3 – All 3 bits of Function Number in Requester ID used for Phantom Functions.

max_payload_size_supported

Max Payload Size Supported

- 0 – 128 Bytes
- 1 – 256 Bytes
- 2 – 512 Bytes
- 3 – 1024 Bytes
- 4 – 2048 Bytes
- 5 – 4096 Bytes
- 6 – Reserved
- 7 – Reserved

2.4.1.4.23. pcie_link_cap Register 0x88

PCI Express Link Capbabilities configuration.

Field	Name	Access	Width	Reset
[31:24]	port_number	read-write	8	0x0
[23:18]	*reserved*	read-only	6	0x0
[17:15]	l1_exit_latency	read-write	3	0x7
[14:12]	l0s_exit_latency	read-write	3	0x7
[11:10]	aspm_support	read-write	2	0x3
[9:0]	*reserved*	read-only	10	0x0

port_number

Port Number. Indicates the PCI Express Port number for the PCI Express Link.

l1_exit_latency

L1 Exit Latency. The value reported indicates the length of time this Port requires to complete transition from ASPM L1 to L0.

- 0 – Less than 1 us
- 1 – 1 us to less than 2 us
- 2 – 2 us to less than 4 us
- 3 – 4 us to less than 8 us
- 4 – 8 us to less than 16 us
- 5 – 16 us to less than 32 us
- 6 – 32 us to 64 us
- 7 – More than 64 us

l0s_exit_latency

L0s Exit Latency. The value reported indicates the length of time this Port requires to complete transition from ASPM L0s to L0.

- 0 – Less than 64 ns
- 1 – 64 ns to less than 128 ns
- 2 – 128 ns to less than 256 ns
- 3 – 256 ns to less than 512 ns
- 4 – 512 ns to less than 1 us
- 5 – 1 us to less than 2 us
- 6 – 2 us to 4 us
- 7 – More than 4 us

aspm_support

Active State Power Management (ASPM) Support

- 0 – No ASPM Support
- 1 – L0s Supported
- 2 – L1 Supported
- 3 – L0s and L1 Supported

2.4.1.4.24. pcie_link_stat Register 0x8c

PCI Express Link Status configuration.

Field	Name	Access	Width	Reset
[31:13]	*reserved*	read-only	19	0x0
[12]	slot_clock_configuration	read-write	1	0x1
[11:0]	*reserved*	read-only	12	0x0

slot_clock_configuration

Indicates whether the component uses the physical reference clock that the platform provides on the connector.

- 0 – Using independent reference clock.
- 1 – Using reference clock provided by slot.

2.4.1.4.25. pcie_slot_cap Register 0x90

PCI Express Slot Capabilities configuration.

Field	Name	Access	Width	Reset
[31:19]	physical_slot_number	read-write	13	0x1
[18]	no_command_completed_support	read-write	1	0x0
[17]	em_interlock_present	read-write	1	0x0
[16:15]	slot_power_limit_scale	read-write	2	0x0
[14:7]	slot_power_limit_value	read-write	8	0xa
[6]	hot_plug_capable	read-write	1	0x0
[5]	hot_plug_surprise	read-write	1	0x0
[4]	power_indicator_present	read-write	1	0x0
[3]	attention_indicator_present	read-write	1	0x0
[2]	mrl_sensor_present	read-write	1	0x0
[1]	power_controller_present	read-write	1	0x0
[0]	attention_button_present	read-write	1	0x0

physical_slot_number

Indicates whether the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is unique within the chassis, regardless of the form factor associated with the slot. This field must be initialized to zero for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Root Port.

no_command_completed_support

Indicates whether the slot generates software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be 1 if the hot-plug capable Port is able to accept writes to all fields of the Slot Control register without delay between successive writes.

- 0 – Software notification provided.
- 1 – Software notification not provided.

em_interlock_present

Indicates whether an Electromechanical Interlock is implemented on the chassis for this slot.

- 0 – Not Supported
- 1 – Supported

slot_power_limit_scale

Slot Power Limit Scale. In combination with the Slot Power Limit Value, specifies the upper limit on power supplied by the slot or by other means to the adapter. See PCIe Spec. Section 6.9 for details.

slot_power_limit_value

Slot Power Limit Value. In combination with the Slot Power Limit Scale, specifies the upper limit on power supplied by the slot or by other means to the adapter. See PCIe Spec. Section 6.9 for details.

hot_plug_capable

Indicates whether this slot is capable of supporting hot-plug operations.

- 0 – Not Supported
- 1 – Supported

hot_plug_surprise

Indicates whether 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.

- 0 – Hot Plug Surprise not possible
- 1 – Hot Plug Surprise possible

power_indicator_present

Indicates whether a Power Indicator is electrically controlled by the chassis for this slot.

- 0 – Not Supported
- 1 – Supported

attention_indicator_present

Indicates whether an Attention Indicator is electrically controlled by the chassis.

- 0 – Not Supported
- 1 – Supported

mrl_sensor_present

Indicates whether a MRL Sensor is implemented on the chassis for this slot.

- 0 – Not Supported
- 1 – Supported

power_controller_present

Indicates whether a software programmable Power Controller is implemented for this slot/adapter.

- 0 – Not Supported
- 1 – Supported

attention_button_present

Indicates whether an Attention Button for this slot is electrically controlled by the chassis.

- 0 – Not Supported
- 1 – Supported

2.4.1.4.26. pcie_dev_cap2 Register 0x98

PCI Express Device Capbabilities 2 configuration.

Field	Name	Access	Width	Reset
[31:22]	*reserved*	read-only	10	0x0
[21]	end_end_prefixes_supported	read-write	1	0x0
[20:19]	*reserved*	read-only	2	0x0
[18]	obff_supported	read-write	1	0x0
[17:16]	*reserved*	read-only	2	0x0
[15:8]	*reserved*	read-only	8	0x0
[7:5]	*reserved*	read-only	3	0x0
[4]	cpl_timeout_disable_supported	read-write	1	0x1
[3:0]	cpl_timeout_ranges_supported	read-write	4	0x0

end_end_prefixes_supported

End-End TLP Prefix Supported

- 0 – Not Supported
- 1 – Supported

obff_supported

OBFF Supported

- 0 – OBFF Not Supported
- 1 – OBFF supported using Message signaling only

cpl_timeout_disable_supported

Completion Timeout Disable Supported. Completion timeout is not implemented by the core, so the advertised value must match the capabilities of the connected design which is implementing completion timeouts.

- 0 – Not Supported
- 1 – Supported

cpl_timeout_ranges_supported

Completion Timeout Ranges Supported advertised value. Completion timeout is not implemented by the core, so the advertised value must match the capabilities of the connected design which is implementing completion timeouts.

- 0 – Completion Timeout programming not supported. Timeout value in the range 50 us to 50 ms is used.
- 1 – Range A (50 us to 10 ms)
- 2 – Range B (10 ms to 250 ms)

- 3 – Range A (50 us to 10 ms) and B (10 ms to 250 ms)
- 4 – Range B (10 ms to 250 ms) and C (250 ms to 4 s)
- 5 – Range A (50 us to 10 ms) and B (10 ms to 250 ms) and C (250 ms to 4 s)
- 6 – Range B (10 ms to 250 ms) and C (250 ms to 4 s) and D (4 s to 64 s)
- 7 – Range A (50 us to 10 ms) and B (10 ms to 250 ms) and C (250 ms to 4 s) and D (4 s to 64 s)
- 8 – Reserved
- 9 – Reserved
- 10 – Reserved
- 11 – Reserved
- 12 – Reserved
- 13 – Reserved
- 14 – Reserved
- 15 – Reserved

2.4.1.4.27. pcie_link_ctl2 Register 0xa0

PCI Express Link Control 2 configuration.

Field	Name	Access	Width	Reset
[31:7]	*reserved*	read-only	25	0x0
[6]	selectable_deemphasis	read-write	1	0x0
[5:0]	*reserved*	read-only	6	0x0

selectable_deemphasis

Selectable Deemphasis setting for Root Port only: When the Link is operating at 5.0 GT/s speed, this bit is used to control the trasmit de-emphasis of the link.

- 0 – -6 dB
- 1 – -3.5 dB

2.4.1.4.28. pm_cap Register 0xc0

Power Management Capbabilities configuration.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15:11]	pme_support	read-write	5	0x1f
[10]	d2_support	read-write	1	0x1
[9]	d1_support	read-write	1	0x1
[8:6]	aux_current	read-write	3	0x0
[5]	dsi	read-write	1	0x0
[4]	*reserved*	read-only	1	0x0
[3]	pme_clock	read-write	1	0x0
[2:0]	version	read-write	3	0x3

pme_support

PME Support. Indicates the power states from which the function may generate a PME. For each power state {D3Cold, D3hot, D2, D1, D0} :

- 0 – PME# not supported
- 1 – PME# supported

d2_support

D2 Power Management State support.

- 0 – Not supported
- 1 – Supported

d1_support

D1 Power Management State support.

- 0 – Not supported
- 1 – Supported

aux_current

Aux Current. Reports the 3.3Vaux auxiliary current requirements for the PCI function. See PCIe Spec. for details.

- 0 – Self powered
- 1 – 55 mA
- 2 – 100 mA
- 3 – 160 mA
- 4 – 220 mA
- 5 – 270 mA
- 6 – 320 mA
- 7 – 375 mA

dsi

Device Specific Initialization. Indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.

- 0 – No Device Specific Initialization necessary.
- 1 – Function requires a device specific initialization sequence following transition to the D0 uninitialized state.

pme_clock

PME Clock. Does not apply to PCI Express and must be 0.

version

PCI Power Management Interface Specification Version. Must be set to 0x3 to indicate revision 1.2 of the PCI Power Management Interface Specification.

2.4.1.4.29. pm Register 0xc4

Power Management Control/Status configuration.

Field	Name	Access	Width	Reset
[31:24]	data	read-write	8	0x0
[23]	pmcsr_bus_p_c_en	read-write	1	0x0
[22]	pmcsr_b2_b3_support	read-write	1	0x0
[21:16]	*reserved*	read-only	6	0x0
[15]	*reserved*	read-only	1	0x0
[14:13]	cstat_data_scale	read-write	2	0x0
[12:9]	cstat_data_select	read-write	4	0x0
[8:0]	*reserved*	read-only	9	0x0

cstat_data_scale

Data Scale

- 0 – Unknown scale
- 1 – power = data * 0.1 Watts
- 2 – power = data * 0.01 Watts
- 3 – power = data * 0.001 Watts

cstat_data_select

Data Select

- 0 – D0 Power Consumed
- 1 – D1 Power Consumed
- 2 – D2 Power Consumed
- 3 – D3 Power Consumed
- 4 – D0 Power Dissipated
- 5 – D1 Power Dissipated
- 6 – D2 Power Dissipated
- 7 – D3 Power Dissipated
- 8 – Common logic power consumptio. For multifunction devices, reported in Function 0 only.
- 9 – Reserved
- 10 – Reserved
- 11 – Reserved
- 12 – Reserved
- 13 – Reserved
- 14 – Reserved
- 15 – Reserved

2.4.1.4.30. pm_aux Register 0xc8

Power Management Auxiliary Power configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	power_required	read-write	1	0x0

power_required

Identifies whether the design requires auxiliary power.

- 0 – Aux Power is not required.
- 1 – Aux Power is required. If Aux Power is required, PME is advertised supported from D3 Cold, or advertised aux_current != 0, then the value of Aux Power PM Enable is sticky and preserved through conventional reset when Aux Power is provided.

2.4.1.4.31. ari_cap Register 0xe0

ARI Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	disable	read-write	1	0x0

disable

ARI Capability Disable. When disabled, the ARI Capability does not appear in PCIe Configuration Space. Must be enabled when SR-IOV is enabled. Must be disabled for downstream ports, Root Complex Integrated Endpoints, and Root Complex Event Collectors.

- 0 – Enable
- 1 – Disable

2.4.1.4.32. msi_cap Register 0xe8

MSI Capability configuration.

Field	Name	Access	Width	Reset
[31:8]	*reserved*	read-only	24	0x0
[7]	*reserved*	read-only	1	0x0
[6:4]	mult_message_capable	read-write	3	0x5
[3:2]	*reserved*	read-only	2	0x0
[1]	vec_mask_capable	read-write	1	0x1
[0]	disable	read-write	1	0x0

mult_message_capable

Number of requested MSI vectors.

- 0 – 1
- 1 – 2
- 2 – 4
- 3 – 8
- 4 – 16
- 5 – 32
- 6 – Reserved
- 7 – Reserved

vec_mask_capable

MSI Capability Per Vector Mask Capable.

- 0 – Disable
- 1 – Enable

disable

MSI Capability Disable. When disabled, the MSI Capability does not appear in PCIe Configuration Space.

- 0 – Enable
- 1 – Disable

2.4.1.4.33. msix_cap Register 0xf0

MSI-X Capability configuration.

Field	Name	Access	Width	Reset
[31:27]	*reserved*	read-only	5	0x0
[26:16]	table_size	read-write	11	0x1f
[15:1]	*reserved*	read-only	15	0x0
[0]	disable	read-write	1	0x0

table_size

Number of requested MSI-X vectors == (table_size+1).

disable

MSI-X Capability Disable. When disabled, the MSI-X Capability does not appear in PCIe Configuration Space.

- 0 – Enable
- 1 – Disable

2.4.1.4.34. msix_table Register 0xf4

MSI-X Capability – MSI-X Table configuration.

Field	Name	Access	Width	Reset
[31:3]	offset	read-write	29	0xc00
[2:0]	bir	read-write	3	0x0

offset

{offset, 3'b000} == byte address offset, within the BAR selected by bir, at which the MSI-X Table begins.

bir

Indicates which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X Table into Memory Space.

- 0 – 0x10 (BAR0)
- 1 – 0x14 (BAR1)
- 2 – 0x18 (BAR2)
- 3 – 0x1C (BAR3)
- 4 – 0x20 (BAR4)
- 5 – 0x24 (BAR5)
- 6 – Reserved
- 7 – Reserved

2.4.1.4.35. msix_pba Register 0xf8

MSI-X Capability – MSI-X PBA configuration.

Field	Name	Access	Width	Reset
[31:3]	offset	read-write	29	0xe00
[2:0]	bir	read-write	3	0x0

offset

{offset, 3'b000} == byte address offset, within the BAR selected by bir, at which the MSI-X PBA begins.

bir

Indicates which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X PBA into Memory Space.

- 0 – 0x10 (BAR0)
- 1 – 0x14 (BAR1)
- 2 – 0x18 (BAR2)
- 3 – 0x1C (BAR3)
- 4 – 0x20 (BAR4)
- 5 – 0x24 (BAR5)
- 6 – Reserved
- 7 – Reserved

2.4.1.4.36. aer_cap Register 0x100

AER Capability configuration.

Field	Name	Access	Width	Reset
[31]	en_tlp_prefix_blocked	read-write	1	0x0
[30]	en_atomicop_egress_blocked	read-write	1	0x0
[29]	en_mc_blocked_tlp	read-write	1	0x0
[28]	en_ucorr_internal_error	read-write	1	0x0
[27]	en_acs_violation	read-write	1	0x0
[26]	en_receiver_overflow	read-write	1	0x0
[25]	en_completer_abort	read-write	1	0x0
[24]	en_completion_timeout	read-write	1	0x1
[23]	en_surprise_down_error	read-write	1	0x0
[22]	en_corr_internal_error	read-write	1	0x0
[21:16]	*reserved*	read-only	6	0x0
[15:2]	*reserved*	read-only	14	0x0
[1]	ecrc_gen_chk_capable	read-write	1	0x1
[0]	version	read-write	1	0x0

en_tlp_prefix_blocked

Enable TLP Prefix Blocked error reporting.

- 0 – Disable
- 1 – Enable

en_atomicop_egress_blocked

Enable AtomicOp Egress Blocked error reporting.

- 0 – Disable
- 1 – Enable

en_mc_blocked_tlp

Enable MC Blocked TLP error reporting. Not supported by core, so must be 0.

- 0 – Disable
- 1 – Enable

en_ucorr_internal_error

Enable Uncorrectable Internal Error.

- 0 – Disable
- 1 – Enable

en_acs_violation

Enable ACS Violation error reporting. Not supported by core, so must be 0.

- 0 – Disable
- 1 – Enable

en_receiver_overflow

Enable Receiver Overflow error reporting. Not supported by core, so must be 0.

- 0 – Disable
- 1 – Enable

en_completer_abort

Enable Completer Abort error reporting.

- 0 – Disable
- 1 – Enable

en_completion_timeout

Enable Completion Timeout error reporting.

- 0 – Disable
- 1 – Enable

en_surprise_down_error

Enable Surprise Down Error error reporting.

- 0 – Disable
- 1 – Enable

en_corr_internal_error

Enable Correctable Internal Error error reporting.

- 0 – Disable
- 1 – Enable

ecrc_gen_chk_capable

ECRC Generation/Checking Capable.

- 0 – Not supported
- 1 – Supported

version

AER Capability Version.

- 0 – Version 0x1
- 1 – Version 0x2

2.4.1.4.37. vsec_cap Register 0x110

Vendor-Specific Capability configuration.

Field	Name	Access	Width	Reset
[31:16]	id	read-write	16	0x1
[15:1]	*reserved*	read-only	15	0x0
[0]	enable	read-write	1	0x1

id

Vendor-Specific Capability ID.

enable

Vendor-Specific Capability Enable. When disabled, the VSEC Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.38. sris_cap Register 0x120

SRIS Capability configuration.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15:12]	low_skp_generation_speeds	read-write	4	0x0
[11:8]	low_skp_reception_speeds	read-write	4	0x0
[7:1]	*reserved*	read-only	7	0x0
[0]	enable	read-write	1	0x0

low_skp_generation_speeds

SRIS Lower SKP OS Generation Supported Speeds Vector advertisement.

low_skp_reception_speeds

SRIS Lower SKP OS Reception Supported Speeds Vector advertisement.

enable

SRIS Capability Enable. When disabled, the SRIS Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.39. dsn_cap Register 0x130

For DSN capable cores only: Device Serial Number Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

Device Serial Number Capability Enable. When disabled, the Device Serial Number Capability Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.40. dsn_serial Register 0x134

Device Serial Number Capability – Serial Number.

Field	Name	Access	Width	Reset
[63:0]	number	read-write	64	0x0

number

Device Serial Number.

2.4.1.4.41. pwr_budget_cap Register 0x150

Power Budgeting Capability configuration.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1]	sys_alloc	read-write	1	0x0
[0]	enable	read-write	1	0x0

sys_alloc

Power Budgeting System Allocated.

- 0 – Power Budget should use Power Budgeting Capability Values
- 1 – Power Budget is System Allocated

enable

Power Budgeting Capability Enable. When disabled, the Power Budgeting Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.42. dpa_cap Register 0x158

Dynamic Power Allocation Capability configuration.

Field	Name	Access	Width	Reset
[31:24]	xlcy1	read-write	8	0x0
[23:16]	xlcy0	read-write	8	0x0
[15:14]	*reserved*	read-only	2	0x0
[13:12]	pas	read-write	2	0x0
[11:10]	*reserved*	read-only	2	0x0
[9:8]	tlunit	read-write	2	0x0
[7:3]	substate_max	read-write	5	0x0
[2:1]	*reserved*	read-only	2	0x0
[0]	enable	read-write	1	0x0

xlcy1

Transition Latency Value 1. When the Transition Latency Indicator for a substate is 1, this value is multiplied by the Transition Latency Unit to determine the maximum Transition Latency for the substate.

xlcy0

Transition Latency Value 0. When the Transition Latency Indicator for a substate is 0, this value is multiplied by the Transition Latency Unit to determine the maximum Transition Latency for the substate.

pas

Power Allocation Scale. The value of the substate Power Allocation Register is multiplied by the decoded value of this field to determine the power allocation of the substate.

- 0 – 10x
- 1 – 1x
- 2 – 0.1x
- 3 – 0.01x

tlunit

Transition Latency Unit. The substate Transition Latency Value is multiplied by the decoded Transition Latency Unit to determine the maximum Transition Latency for the substate.

- 0 – 1 ms
- 1 – 10 ms
- 2 – 100 ms
- 3 – Reserved

substate_max

Substate_Max. Specifies the maximum substate number. Substates from [substate_max:0] are supported. For example, substate_max==0 indicates support for 1 substate.

enable

Dynamic Power Allocation (DPA) Capability Enable. When disabled, the Dynamic Power Allocation Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.43. dpa_xlcy Register 0x15c

Dynamic Power Allocation – Transition Latency.

Field	Name	Access	Width	Reset
[31:0]	indicator	read-write	32	0x0

indicator

Transition Latency Indicator. Indiates which Transition Latency Value applies to each substate. For each substate[i], indicator[i] indicates which Transition Latency Value applies:

- 0 – Use Transition Latency Value 0
- 1 – Use Transition Latency Value 1

2.4.1.4.44. dpa_alloc Register 0x160

Dynamic Power Allocation Capability – Dynamic Power Allocation Array.

Field	Name	Access	Width	Reset
[255:0]	array	read-write	256	0x0

array

Substate Power Allocation Array. For each substate[i], multiply array[(i*8)+7:(i*8)] times the Power Allocation Scale to determine the power allocation in Watts for the associated substate.

2.4.1.4.45. ltr_cap Register 0x180

Latency Tolerance Reporting Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

Latency Tolerance Reporting Capability Enable. When disabled, the Latency Tolerance Reporting Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.46. l1pmss_cap Register 0x188

L1 PM Substates Capability configuration.

Field	Name	Access	Width	Reset
[31:24]	cm_restore_time	read-write	8	0x0
[23:16]	port_cm_restore_time	read-write	8	0x0
[15:11]	port_tpower_on_value	read-write	5	0x0
[10]	*reserved*	read-only	1	0x0
[9:8]	port_tpower_on_scale	read-write	2	0x0
[7]	pcipm_l1_1_supported	read-write	1	0x1
[6]	pcipm_l1_2_supported	read-write	1	0x1
[5]	aspm_l1_1_supported	read-write	1	0x1
[4]	aspm_l1_2_supported	read-write	1	0x1
[3]	l1pm_supported	read-write	1	0x1
[2:1]	*reserved*	read-only	2	0x0
[0]	enable	read-write	1	0x0

cm_restore_time

Default Common Mode Restore Time. Default time, in microseconds, used by the Downstream Port for timing the re-establishment of common mode. See the L1 PM Substates ECN for further details.

port_cm_restore_time

Port Common Mode Restore Time. Time, in microseconds, required for this port to re-establish common mode. See the L1 PM Substates ECN for further details.

port_tpower_on_value

Port TPOWER_ON Value. Required for ports supporting PCI-PM L1.2 or ASPM L1.2. The value of TPOWER_ON is calculated by multiplying the value in this field by the decoded TPOWER_ON Scale field.

port_tpower_on_scale

Port TPOWER_ON Scale. Required for ports supporting PCI-PM L1.2 or ASPM L1.2.

- 0 – 2 us
- 1 – 10 us
- 2 – 100 us
- 3 – Reserved

pcipm_l1_1_supported

PCI-PM L1.1 Substate Supported. Must be set to 1 for all ports supporting L1 PM Substates.

- 0 – Not supported
- 1 – Supported

pcipm_l1_2_supported

PCI-PM L1.2 Substate Supported.

- 0 – Not supported
- 1 – Supported

aspm_l1_1_supported

ASPM L1.1 Substate Supported.

- 0 – Not supported
- 1 – Supported

aspm_l1_2_supported

ASPM L1.2 Substate Supported.

- 0 – Not supported
- 1 – Supported

l1pm_supported

L1 PM Substates Supported.

- 0 – Not supported
- 1 – Supported

enable

L1 PM Substates Capability Enable. When disabled, the L1 PM Substates Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.47. rbar_cap Register 0x1a0

Resizable BAR Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x1

enable

Resizable BAR Capability Enable. When disabled, the Resizable BAR Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.48. rbar_cfg0 Register 0x1a4

Resizable BAR Capability – BAR Configuration 0.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0xf
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x0

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.49. rbar_cfg1 Register 0x1a8

Resizable BAR Capability – BAR Configuration 1.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x1

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.50. rbar_cfg2 Register 0x1ac

Resizable BAR Capability – BAR Configuration 2.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x2

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.51. rbar_cfg3 Register 0x1b0

Resizable BAR Capability – BAR Configuration 3.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x3

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.52. rbar_cfg4 Register 0x1b4

Resizable BAR Capability – BAR Configuration 4.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x4

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.53. rbar_cfg5 Register 0x1b8

Resizable BAR Capability – BAR Configuration 5.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x5

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.4.54. ats_cap Register 0x1c0

For ATS capable cores only: ATS Capability configuration.

Field	Name	Access	Width	Reset
[31:17]	*reserved*	read-only	15	0x0
[16]	global_inval_support	read-write	1	0x1
[15:13]	*reserved*	read-only	3	0x0
[12:8]	inval_q_depth	read-write	5	0x0
[7:1]	*reserved*	read-only	7	0x0
[0]	enable	read-write	1	0x0

global_inval_support

Cores with both ATS & PASID support only: ATS/PASID Global Invalidate Support. If set to 1, the function supports Invalidate Requests with the Global Invalidate bit set.

inval_q_depth

ATS Invalidate Queue Depth. Number of invalidate requests that can be queued. 0 is a special case that indicates a queue depth of 32.

enable

ATS Capability Enable. When disabled, the ATS Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.4.55. atomic_op_cap Register 0x1cc

Atomic Op Capability configuration.

Field	Name	Access	Width	Reset
[31:6]	*reserved*	read-only	26	0x0
[5]	rp_completer_enable	read-write	1	0x0
[4]	completer_128_supported	read-write	1	0x0
[3]	completer_64_supported	read-write	1	0x0
[2]	completer_32_supported	read-write	1	0x0
[1]	routing_supported	read-write	1	0x0
[0]	enable	read-write	1	0x0

rp_completer_enable

Enable Root Port to be an Atomic Op Completer which means that the Root Port completes rather than forwards Atomic Op TLPs.

- 0 – Disable
- 1 – Enable

completer_128_supported

Atomic Op Completer 128-bit Operand Support.

- 0 – NotSupported
- 1 – Supported

completer_64_supported

Atomic Op Completer 64-bit Operand Support.

- 0 – NotSupported
- 1 – Supported

completer_32_supported

Atomic Op Completer 32-bit Operand Support.

- 0 – NotSupported
- 1 – Supported

routing_supported

Atomic Op Routing Supported.

- 0 – NotSupported
- 1 – Supported

enable

Atomic Op Capability Enable. When disabled, the Atomic Op Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.5. mgmt_ftl_mf[3:1] (0x07000, 0x06000, 0x05000) (Function 1-3)

mgmt_ftl_mf1_BASE	0x5000
mgmt_ftl_mf2_BASE	0x6000
mgmt_ftl_mf3_BASE	0x7000

2.4.1.5.1. function Register 0x8

Function disable for Functions[3:1]. Function[0] may not be disabled.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	disable	read-write	1	0x0

disable

Function disable for Functions[3:1]. Function[0] may not be disabled.

- 0 – Enable
- 1 – Disable

2.4.1.5.2. us_port Register 0x38

Upstream Port Configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	adv_target_link_speed	read-write	1	0x0

adv_target_link_speed

For an upstream port, advertise the link speeds specified by the target_link_speed field rather than the maximum supported speed.

2.4.1.5.3. id1 Register 0x40

ID1 configuration.

Field	Name	Access	Width	Reset
[31:16]	device_id	read-write	16	0xe004
[15:0]	vendor_id	read-write	16	0x19aa

device_id

Value returned when the Device ID Configuration Register is read.

vendor_id

Value returned when the Vendor ID Configuration Register is read.

2.4.1.5.4. id2 Register 0x44

ID2 configuration.

Field	Name	Access	Width	Reset
[31:16]	subsystem_id	read-write	16	0xe004
[15:0]	subsystem_vendor_id	read-write	16	0x19aa

subsystem_id

Value returned when the Subsystem ID Configuration Register is read.

subsystem_vendor_id

Value returned when the Subsystem Vendor ID Configuration Register is read.

2.4.1.5.5. id3 Register 0x48

ID3 configuration.

Field	Name	Access	Width	Reset
[31:8]	class_code	read-write	24	0x118000
[7:0]	revision_id	read-write	8	0x4

class_code

Value returned when the Class Code Configuration Register is read. Must be set to the correct value for the type of device being implemented; see PCI Local Bus Specification Revision 2.3 Appendix D for details on setting Class Code.

revision_id

Value returned when the Revision ID Configuration Register is read.

2.4.1.5.6. cardbus Register 0x4c

Cardbus configuration.

Field	Name	Access	Width	Reset
[31:0]	cis_pointer	read-write	32	0x0

cis_pointer

Value returned when the Cardbus CIS Pointer Configuration Register is read. Set to 0x00000000 unless a Cardbus CIS structure is implemented in memory (which is rare), in which case set to the address of the CIS Structure.

2.4.1.5.7. interrupt Register 0x50

Interrupt configuration.

Field	Name	Access	Width	Reset
[31:10]	*reserved*	read-only	22	0x0
[9:8]	pin	read-write	2	0x0
[7:1]	*reserved*	read-only	7	0x0
[0]	disable	read-write	1	0x0

pin

Selects which legacy interrupt is used.

- 0 – INTA
- 1 – INTB
- 2 – INTC
- 3 – INTD

disable

Disable support for interrupts.

- 0 – Enable
- 1 – Disable

2.4.1.5.8. bar0 Register 0x60

BAR0 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffff000c

cfg

Configuration of BAR0 (Cfg address 0x10). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.9. bar1 Register 0x64

BAR1 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar1 (Cfg address 0x14). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.10. bar2 Register 0x68

BAR2 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffe00c

cfg

Configuration of bar2 (Cfg address 0x18). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.11. bar3 Register 0x6c

BAR3 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar3 (Cfg address 0x1C). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.12. bar4 Register 0x70

BAR4 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffe00c

cfg

Configuration of bar4 (Cfg address 0x20). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.13. bar5 Register 0x74

BAR5 configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0xffffffff

cfg

Configuration of bar5 (Cfg address 0x24). Use to define a 32-bit Memory or IO region or combine with an adjacent BAR to define a 64-bit Memory region.

2.4.1.5.14. exp_rom Register 0x78

Expansion ROM configuration.

Field	Name	Access	Width	Reset
[31:0]	cfg	read-write	32	0x0

cfg

Configuration of exp_rom. Use to define a 32-bit Memory Expansion ROM region. If an Expansion ROM region is defined, then the region must map to PCIe-compliant Expansion ROM code or the device may fail to boot.

2.4.1.5.15. msi_cap Register 0xe8

MSI Capability configuration.

Field	Name	Access	Width	Reset
[31:8]	*reserved*	read-only	24	0x0
[7]	*reserved*	read-only	1	0x0
[6:4]	mult_message_capable	read-write	3	0x5
[3:2]	*reserved*	read-only	2	0x0
[1]	vec_mask_capable	read-write	1	0x1
[0]	disable	read-write	1	0x0

mult_message_capable

Number of requested MSI vectors.

- 0 – 1
- 1 – 2
- 2 – 4
- 3 – 8
- 4 – 16
- 5 – 32
- 6 – Reserved
- 7 – Reserved

vec_mask_capable

MSI Capability Per Vector Mask Capable.

- 0 – Disable
- 1 – Enable

disable

MSI Capability Disable. When disabled, the MSI Capability does not appear in PCIe Configuration Space.

- 0 – Enable
- 1 – Disable

2.4.1.5.16. msix_cap Register 0xf0

MSI-X Capability configuration.

Field	Name	Access	Width	Reset
[31:27]	*reserved*	read-only	5	0x0
[26:16]	table_size	read-write	11	0x1f
[15:1]	*reserved*	read-only	15	0x0
[0]	disable	read-write	1	0x0

table_size

Number of requested MSI-X vectors == (table_size+1).

disable

MSI-X Capability Disable. When disabled, the MSI-X Capability does not appear in PCIe Configuration Space.

- 0 – Enable
- 1 – Disable

2.4.1.5.17. msix_table Register 0xf4

MSI-X Capability – MSI-X Table configuration.

Field	Name	Access	Width	Reset
[31:3]	offset	read-write	29	0xc00
[2:0]	bir	read-write	3	0x0

offset

{offset, 3'b000} == byte address offset, within the BAR selected by bir, at which the MSI-X Table begins.

bir

Indicates which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X Table into Memory Space.

- 0 – 0x10 (BAR0)
- 1 – 0x14 (BAR1)
- 2 – 0x18 (BAR2)
- 3 – 0x1C (BAR3)
- 4 – 0x20 (BAR4)
- 5 – 0x24 (BAR5)
- 6 – Reserved
- 7 – Reserved

2.4.1.5.18. msix_pba Register 0xf8

MSI-X Capability – MSI-X PBA configuration.

Field	Name	Access	Width	Reset
[31:3]	offset	read-write	29	0xe00
[2:0]	bir	read-write	3	0x0

offset

{offset, 3'b000} == byte address offset, within the BAR selected by bir, at which the MSI-X PBA begins.

bir

Indicates which Base Address register, located beginning at 10h in Configuration Space, is used to map the MSI-X PBA into Memory Space.

- 0 – 0x10 (BAR0)
- 1 – 0x14 (BAR1)
- 2 – 0x18 (BAR2)
- 3 – 0x1C (BAR3)
- 4 – 0x20 (BAR4)
- 5 – 0x24 (BAR5)
- 6 – Reserved
- 7 – Reserved

2.4.1.5.19. dsn_cap Register 0x130

For DSN capable cores only: Device Serial Number Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x0

enable

Device Serial Number Capability Enable. When disabled, the Device Serial Number Capability Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.5.20. dsn_serial Register 0x134

Device Serial Number Capability – Serial Number.

Field	Name	Access	Width	Reset
[63:0]	number	read-write	64	0x0

number

Device Serial Number.

2.4.1.5.21. rbar_cap Register 0x1a0

Resizable BAR Capability configuration.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	enable	read-write	1	0x1

enable

Resizable BAR Capability Enable. When disabled, the Resizable BAR Capability does not appear in PCIe Configuration Space.

- 0 – Disable
- 1 – Enable

2.4.1.5.22. rbar_cfg0 Register 0x1a4

Resizable BAR Capability – BAR Configuration 0.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0xf
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x0

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.5.23. rbar_cfg1 Register 0x1a8

Resizable BAR Capability – BAR Configuration 1.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x1

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.5.24. rbar_cfg2 Register 0x1ac

Resizable BAR Capability – BAR Configuration 2.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x2

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.5.25. rbar_cfg3 Register 0x1b0

Resizable BAR Capability – BAR Configuration 3.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x3

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.5.26. rbar_cfg4 Register 0x1b4

Resizable BAR Capability – BAR Configuration 4.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x4

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.5.27. rbar_cfg5 Register 0x1b8

Resizable BAR Capability – BAR Configuration 5.

Field	Name	Access	Width	Reset
[31:29]	*reserved*	read-only	3	0x0
[28:24]	size	read-write	5	0x0
[23:4]	supported_sizes	read-write	20	0x0
[3]	*reserved*	read-only	1	0x0
[2:0]	bar_index	read-write	3	0x5

size

Default BAR Size. Indicates the default size after reset for this BAR. BAR Size == 2size+20 bytes. For example, if this field is set to a value of 3, that indicates this BAR has a default size of 2^23=8MB. The max value is 19 (2^39==512 GB). The default value must be one of the supported BAR sizes indicated by supported_sizes

supported_sizes

Supported BAR Sizes. supported_sizes[i] indicates a BAR Size of 2i+20 is supported for this BAR. For example. If supported_sizes[0] is set, then a BAR size of 2^20=2MB is supported.

bar_index

BAR Index. BAR offset for which this configuration is valid. For a 64-bit BAR, this index must indicate the lower DWORD used for the BAR.

- 0 – BAR located at Configuration Register address offset 0x10.
- 1 – BAR located at Configuration Register address offset 0x14.
- 2 – BAR located at Configuration Register address offset 0x18.
- 3 – BAR located at Configuration Register address offset 0x1C.
- 4 – BAR located at Configuration Register address offset 0x20.
- 5 – BAR located at Configuration Register address offset 0x24.
- 6 – Reserved
- 7 – Reserved

2.4.1.6. pcie_ll (0x0F000)

pcie_ll_BASE 0xF000

2.4.1.6.1. main_ctrl_0 Register 0x0

Main Control 0 register.

Field	Name	Access	Width	Reset
[31]	en_user_write	read-write	1	0x1
[30:17]	*reserved*	read-only	14	0x0
[16]	disable_csr_reset_port	read-write	1	0x0
[15:6]	*reserved*	read-only	10	0x0
[5]	sel_pclk_div2	read-write	1	0x1
[4:2]	num_lanes	read-write	3	0x1
[1]	core_bypass	read-write	1	0x0
[0]	core_enable	read-write	1	0x1

en_user_write

This option allows you to modify the values of this register (excluding this field). By default, you have write and read access to this register.

- 0 – User has read only access
- 1 – User has read/write access

disable_csr_reset_port

Disables the reset of configuration and status registers (CSR) through reset port.

- 0 – Asserting the usr_lmmi_resetn_i resets the CSRs
- 1 – Disable reset port. (Users can still use soft reset by writing to the reset register pcie_ll_main_ctrl_2[0])

sel_pclk_div2

This field selects the clock output on port clk_usr_o.

- 0 – pclk (250 MHz)
- 1 – pclk_div2 (125 MHz)

num_lanes

This field indicates the maximum number of lanes that are used when PCIe LL core is enabled.

- 1 – 1 Lane

core_bypass

This option allows you to bypass the PCIe Link Layer and directly connect to the PMA using the shared user interface. The bypass mode is applicable for protocols other than PCIe which is out of scope of this user guide.

- 0 – Normal
- 1 – Bypass PCIe Link Layer

core_enable

Enable or disable the PCIe Link Layer Core.

- 0 – Disable
- 1 – Enable

2.4.1.6.2. main_ctrl_1 Register 0x4

Main Control 1 register.

Field	Name	Access	Width	Reset
[31:17]	*reserved*	read-only	15	0x0
[16]	hold_reset	read-write	1	0x0
[15:9]	*reserved*	read-only	7	0x0
[8]	pipe_reset	read-write	1	0x0
[7:1]	*reserved*	read-only	7	0x0
[0]	core_reset	read-write	1	0x0

hold_reset

Controls the core_reset and pipe_reset if it remains asserted or automatically deassert.

- 0 – writing 1 to core_reset/pipe_reset field toggles the PCIe Link Layer core reset or PIPE reset for 1 clock cycle
- 1 – Hold the core_reset/pipe_reset (core_reset/pip_reset does not automatically deassert unless 0 is written to the corresponding field)

pipe_reset

This field controls the PIPE reset (PCS reset). The behaviour of pipe_reset depends on the hold_reset field.

- 0 – Deassert PIPE Reset (Normal operation)
- 1 – Assert PIPE Reset

core_reset

This field controls the PCIe Link Layer core reset. The behaviour of core_reset depends on the hold_reset field.

- 0 – Deassert Core Reset (Normal operation)
- 1 – Assert Core Reset

2.4.1.6.3. main_ctrl_2 Register 0x8

Main Control 2 register.

Field	Name	Access	Width	Reset
[31:2]	*reserved*	read-only	30	0x0
[1]	ll_csr_reset	read-write	1	0x0
[0]	phy_csr_reset	read-write	1	0x0

ll_csr_reset

This field controls the reset of PCIe Link Layer mgmt_* configuration and status registers. Automatically returns to 0 after a write of 1.

- 0 – reserved
- 1 – Assert Link Layer CSR Reset, writing 1 to ll_csr_reset field toggles the Link Layer CSR reset for 1 clock cycle

phy_csr_reset

This field controls the reset of PHY configuration and status registers. Automatically returns to 0 after a write of 1.

- 0 – reserved
- 1 – Assert PHY CSR Reset, writing 1 to phy_csr_reset field toggles the PHY CSR reset for 1 clock cycle

2.4.1.6.4. main_ctrl_3 Register 0xC

Main Control 3 register.

Field	Name	Access	Width	Reset
[31:16]	u_clk_period_in_ps	read-write	16	0x1F40
[15:0]	p_clk_period_in_ps	read-write	16	0xFA0

u_clk_period_in_ps

The current period of clk_usr in picoseconds. This is used to time events with fixed time duration such as LTSSM state machine timeouts. Default is 8000 ps (125 MHz).

p_clk_period_in_ps

The current period of pclk in picoseconds. This is used to time events with fixed time duration such as LTSSM state machine timeouts. Default is 4000 ps (250 MHz).

2.4.1.6.5. main_ctrl_4 Register 0x10

Main Control 4 register.

Field	Name	Access	Width	Reset
[31:16]	aux_clk_period_in_ps	read-write	16	0xF424
[15:3]	*reserved*	read-only	13	0x0
[2]	merge_cfgreg_lmmi_rdata	read-write	1	0x0
[1]	en_port_mgmt_interrupt_leg	read-write	1	0x1
[0]	en_port_mgmt_ltssm_disable	read-write	1	0x0

aux_clk_period_in_ps

The current period of phy aux_clk in picoseconds. This is used to time events with fixed time duration such as LTSSM state machine timeouts. Default is 62500 ps (16 MHz).

merge_cfgreg_lmmi_rdata

This option is provided to allow the reduction of ports and merge the PCIe Configuration Register read data port (ucfg_rd_data_o) with the LMMI read data (usr_lmmi_rdata_o) port. When enabled, it is expected that you will not issue a simultaneous read access on CSR and PCIe Configuration Registers.

- 0 – Disable
- 1 – Enable

en_port_mgmt_interrupt_leg

Enables the input port mgmt_interrupt_leg, otherwise use register access.

- 0 – Disable
- 1 – Enable

en_port_mgmt_ltssm_disable

Enables the input port u_ltssm_disable_i, otherwise use register access (see register pcie_ll_conv_port_0).

- 0 – Disable
- 1 – Enable

2.4.1.6.6. main_ctrl_5 Register 0x14

Main Control 5 register.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	en_pipe_if_ctrl	read-write	1	0x0

en_pipe_if_ctrl

When enabled, allows you to control the following pipe interface signals:

– pipe_pclkreq_n, pipe_rx_ei_disable, pipe_tx_cm_disable, pipe_power_down. This should not be enabled during normal operation.

- 0 – Disable
- 1 – Enable

2.4.1.6.7. conv_port_0 Register 0x100

Converted Port 0 register.

Field	Name	Access	Width	Reset
[31:1]	*reserved*	read-only	31	0x0
[0]	mgmt_ltssm_disable	read-write	1	0x0

mgmt_ltssm_disable

(see also: register pcie_ll_main_ctrl_4)

The LTSSM does not transition from Detect.Quiet to Detect. Active to begin LTSSM training while mgmt_ltssm_disable ==1. mgmt_ltssm_disable may be used to delay the start of LTSSM training which otherwise begins as soon as rst_usr_n is deasserted. mgmt_ltssm_disable must be set to 1 relatively soon (within a few mS) after rst_usr_n is released as the system allocates a finite amount of time for devices to initialize before it begins to scan for devices. If mgmt_ltssm_disable is held for too long, software may scan for the device before it becomes operational and assume that no device is present.

2.4.1.6.8. conv_port_1 Register 0x104

Converted Port 1 register.

Field	Name	Access	Width	Reset
[31:4]	*reserved*	read-only	28	0x0
[3:0]	mgmt_interrupt_leg	read-write	4	0x0

mgmt_interrupt_leg

(see also: register pcie_ll_main_ctrl_4)

When Legacy Interrupt Mode is enabled, mgmt_interrupt_leg implements one level-sensitive interrupt (INTA, INTB, INTC, or INTD) for each Base Function. Each functions' interrupt sources must be logically ORed together and input as mgmt_interrupt_leg[i] for a given function i. Each interrupt source must continue to drive a 1 until it has been serviced and cleared by software at which time it must switch to driving 0. The core ORs together INTA/B/C/D from all functions to create an aggregated INTA/INTB/INTC/INTD. The core monitors high and low transitions on the aggregated INTA/B/C/D and sends an Interrupt Assert message on each 0 to 1 transition and an Interrupt De-Assert Message on each 1 to 0 transition of the aggregated INTA/B/C/D. Transitions which occur too close together to be independently transmitted are merged.

2.4.1.6.9. conv_port_2 Register 0x108

Converted Port 2 register.

Field	Name	Access	Width	Reset
[31:5]	*reserved*	read-only	27	0x0
[4:3]	pipe_power_down	read-write	2	0x0
[2]	pipe_tx_cm_disable	read-write	1	0x0
[1]	pipe_rx_ei_disable	read-write	1	0x0
[0]	pipe_pclkreq_n	read-write	1	0x0

pipe_power_down

Applicable if en_pipe_if_ctrl == 1 (pcie_ll_main_ctrl_5[0]).

Set this register to force drive the pipe interface signal. Power up or down the transceiver.

- 00 – P0, normal operation
- 01 – P0s, low recovery time latency power saving state
- 10 – P1, longer recovery time latency power saving state
- 11 – P2, lowest power state

pipe_tx_cm_disable

Applicable if en_pipe_if_ctrl == 1 (pcie_ll_main_ctrl_5[0]).

Set this register to force drive the pipe interface signal. L1 substate disable Tx common mode voltage. Through this signal the Link Layer effectively configure the Tx driver into Hi-Z (power down) and move the PHY to L1.2 (Tx common mode voltage is disabled).

pipe_rx_ei_disable

Applicable if en_pipe_if_ctrl == 1 (pcie_ll_main_ctrl_5[0]).

Set this register to force drive the pipe interface signal. L1 substate disable activity detector. Through this signal the Link Layer effectively disable the activity detector circuit (power down) and move the PHY to either L1.1 (Tx common mode voltage is still valid) or L1.2 (Tx common mode voltage is disabled).

pipe_pclkreq_n

Applicable if en_pipe_if_ctrl == 1 (pcie_ll_main_ctrl_5[0]).

Set this register to force drive the pipe interface signal. L1 substate request. Active low request to enter L1 substate. The Link Layer should wait for pipe_pclkack_n assertion (low) before to effectively gate the reference clock on the board through CLKREQ# out of band signal.

2.4.1.6.10. stat_port_0 Register 0x200

Status Port 0 register.

Field	Name	Access	Width	Reset
[31:16]	*reserved*	read-only	16	0x0
[15:14]	phy_sts_pipe_power_down	read-only	2	0x0
[13]	phy_sts_pipe_tx_cm_disable	read-only	1	0x0
[12]	phy_sts_pipe_rx_ei_disable	read-only	1	0x0
[11]	phy_sts_pipe_pclkack_n	read-only	1	0x0
[10]	phy_sts_pipe_pclkreq_n	read-only	1	0x0
[9]	phy_sts_pipe_phy_status	read-write	1	0x0
[8]	phy_sts_pipe_rstn	read-write	1	0x0
[7:6]	*reserved*	read-only	2	0x0
[5]	phy_sts_arxpllstable	read-only	1	0x0
[4]	phy_sts_atxpllstable	read-only	1	0x0
[3]	phy_sts_acdrdiagout	read-only	1	0x0
[2]	phy_sts_atrandet	read-only	1	0x0
[1]	phy_sts_acdrpllrstb	read-only	1	0x0
[0]	phy_sts_txpllrstb	read-only	1	0x0

phy_sts_pipe_power_down

Signal from PIPE interface. This register may not reflect the current value due to synchronization.

Power up or down the transceiver.

- 00 – P0, normal operation
- 01 – P0s, low recovery time latency power saving state
- 10 – P1, longer recovery time latency power saving state
- 11 – P2, lowest power state

phy_sts_pipe_tx_cm_disable

Signal from PIPE interface. This register may not reflect the current value due to synchronization.

L1 substate disable Tx common mode voltage. Through this signal the LL effectively configure the Tx driver into Hi-Z (power down) and move the PHY to L1.2 (Tx common mode voltage is disabled).

phy_sts_pipe_rx_ei_disable

Signal from PIPE interface. This register may not reflect the current value due to synchronization.

L1 substate disable Tx common mode voltage. Through this signal the LL effectively configure the Tx driver into Hi-Z (power down) and move the PHY to L1.2 (Tx common mode voltage is disabled).

phy_sts_pipe_pclkack_n

Signal from PIPE interface. This register may not reflect the current value due to synchronization.

L1 substate acknowledge. Active low acknowledge signal to enter L1 substate. The Link Layer should wait for pipe_pclkack_n assertion (low) before to effectively gate the reference clock on the board through CLKREQ# out of band signal.

phy_sts_pipe_pclkreq_n

Signal from PIPE interface. This register may not reflect the current value due to synchronization.

L1 substate request. Active low request signal to enter L1 substate. The Link Layer should wait for pipe_pclkack_n assertion (low) before to effectively gate the reference clock on the board through CLKREQ# out of band signal.

phy_sts_pipe_phy_status

Signal from PIPE interface.

- 0 – otherwise
- 1 – pipe_phy_status was asserted. Write 1 to clear.

phy_sts_pipe_rstn

Signal from PIPE interface.

- 0 – otherwise
- 1 – pipe_rstn was asserted. Write 1 to clear.

phy_sts_arxpllstable

Signal from PMA interface. Rx PLL locked.

phy_sts_atxpllstable

Signal from PMA interface. Tx PLL locked.

phy_sts_acdrdiagout

Signal from PMA interface. CDR PLL locked on data.

phy_sts_atrandet

Signal from PMA interface. Activity detected.

phy_sts_acdrpllrstb

Signal from PMA interface. Rx PLL reset.

phy_sts_txpllrstb

Signal from PMA interface. Tx PLL reset.

2.4.2. PCI Express Configuration Space Registers

The Lattice PCIe X1 Core implements Header Type 00 and Header Type 01 Configuration Registers, including Capability and Extended Capability Items, as detailed in the PCI Express Base Specification, Rev 3.0, PCI Local Interface Specification Revision 3.0, and PCI Bus Power Management Interface Specification Revision 1.2.

Type 00 and Type 01 Configuration Registers implement the first 64 bytes of Configuration Space differently:

- Type 00 – Implemented by Endpoints; see Table 2.30
- Type 01 – Implemented by Root Ports; see Table 2.31

Capability and Extended Capability Items are located at the same addresses regardless of which header type is implemented; see Table 2.32, Table 2.30, Table 2.31, and Table 2.32 illustrate the core's PCIe Configuration Register map.

The core's Configuration Registers are highly configurable. In dual-mode (Root-Port/Endpoint) applications the registers configure themselves according to the mode of operation – changing between Type 00 and Type 01 for instance when changing between an Endpoint and Root Port design.

The Configuration Regsiters provide the ability for standard PCI/PCIe BIOS/OS software to discover the device, determine its capabilities, and configure the core's features. Since there are a tremendous variety of applications, the core's Configiruation Registers are highly configurable.

2.4.2.1. Type 00 Configuration Header

Table 2.30. Type 00 Configuration Header

Addr	Byte3	Byte2	Byte1	Byte0
00	Device ID		Vendor ID	
04	Status		Command	
08	Class Code			Revision ID
0C	BIST	Header Type	Latency Timer	Cache Line Size
10	Base Address Register 0			
14	Base Address Register 1			
18	Base Address Register 2			
1C	Base Address Register 3			
20	Base Address Register 4			
24	Base Address Register 5			
28	Cardbus CIS Pointer			
2C	Subsystem ID		Subsystem Vendor ID	
30	Expansion ROM Base Address			
34	Reserved			Capabilities Pointer
38	Reserved			
3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

2.4.2.2. Type 01 Configuration Header

Table 2.31. Type 01 Configuration Header

Addr	Byte3	Byte2	Byte1	Byte0
00	Device ID		Vendor ID	
04	Status		Command	
08	Class Code			Revision ID
0C	BIST	Header Type	Primary Latency Timer	Cache Line Size
10	Base Address Register 0			
14	Base Address Register 1			
18	Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number
1C	Secondary Status		I/O Limit	I/O Base
20	Memory Limit		Memory Base	
24	Prefetchable Memory Limit		Prefetchable Memory Base	
28	Prefetchable Base Upper 32 Bits			
2C	Prefetchable Limit Upper 32 Bits			
30	I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits	
34	Reserved			Capability Pointer
38	Expansion ROM Base Address			
3C	Bridge Control		Interrupt Pin	Interrupt Line

2.4.2.3. Capability and Extended Capability Address Locations

Table 2.32. Capability and Extended Capability Items

Addr	Byte3	Byte2	Byte1	Byte0
7B-40	PCI Express Capability			
7F-7C	Reserved			
87-80	Power Management Capability			
8F-88	Reserved			
9B-90	MSI-X Capability			
9F-9C	Reserved			
AD-A0	MSI Capability			
FF-AE	Reserved			
147-100	Advanced Error Reporting Capability			
14F-148	ARI Capability			
173-150	Vendor-Specific Extended Capability			
17f-174	Reserved			
1AB-180	Secondary PCI Express Extended Capability			
1FF-1AC	Reserved			
207-200	ATS Capability			
20F-208	Reserved			
21B-210	DSN Capability			
26B-240	Reserved			
2BF-280	Resizable BAR Capability			
38F-2C0	Reserved			
39F-390	Power Budgeting Capability			
3CF-3A0	Dynamic Power Allocation (DPA) Capability			
3DF-3D0	L1 PM Substates Extended Capability			
3E7-3E0	Latency Tolerance Reporting (LTR) Capability			
FFF-3E8	Reserved			

2.4.2.4. Type 00 Configuration Registers

Table 2.33. Type 00 Configuration Registers

Addr	Config Register	Register Description
01 – 00	Vendor ID	Read Only: This field identifies the manufacturer of the device.
03 – 02	Device ID	Read Only: This field identifies the particular device.
05 – 04	Command Register	Command Register Bits: Bits 10, 8, 6, and 2..0 are Read/Write. Bits[15:11] = 00000. Not implemented. Bit[10] – Interrupt Disable – If set, interrupts are disabled and cannot be generated; if clear interrupts are enabled Bit[9] = 0. Not implemented. Bit[8] – SERR Enable – When set enables the reporting of fatal and non-fatal errors detected by the device to the root complex. Bit[7] = 0. Not implemented. Bit[6] – Parity Error Enable – Affects the mapping of PCI Express errors to legacy PCI errors. See *PCI Express Base Specification Rev1.1*, Section 6.2 for details. Bit[5] = 0. Not implemented. Bit[4] = 0. Not implemented. Bit[3] = 0. Not implemented. Bit[2] – Bus Master Enable – Memory and I/O Requests can only be generated on the Transaction Layer Interface if this bit is set. Bit[1] – Memory Space Enable – If set, the core decodes packets to determine memory BAR hits; if clear, memory BARs are disabled Bit[0] – I/O Space Enable – If set, the core decodes packets to determine I/O BAR hits; if clear, I/O BARs are disabled
07 – 06	Status Register	Status Register Bits: Bits 15..11 and 8 are Read/Write. Writing a 1 to a bit location clears that bit. Writing a 0 to a bit location has no affect. Bit[15] – Set by a device whenever it receives a Poisoned TLP. Bit[14] – Set when a device sends an ERR_FATAL or ERR_NONFATAL Message and the SERR Enable bit in the Command Register is set. Bit[13] – Set when a requestor receives a completion with Unrecognized Request Completion Status Bit[12] – Set when a requestor receives a completion with Completer Abort Completion Status Bit[11] – Set when a device completes a request using Completer Abort Completion Status Bits[10:9] = 00. Not implemented. Bit[8] – Master Data Parity Error – This bit is set by a Requestor if its Parity Error Enable bit is set and either a Completion is received that is marked poisoned or the requestor poisons a write request. Bits[7:5] = 000. Not implemented. Bit[4] = 1 to indicate the presence of a Capabilities List. Bit[3] – Interrupt Status – Reflects the value of mgmt_interrupt Bits[2:0] = 000. Reserved.
08	Revision ID	Read Only: This register specifies the device specific revision identifier.
0B – 09	Class Code	Read Only: The Class Code identifies the generic function of the device.
0C	0x0C: Cache Line Size	Read/Write: Cache Line Size is not used with PCI Express but is still implemented as read/write register for legacy compatibility purposes.
0D	0x0D: Latency Timer	Read Only returning 0x00.
0E	0x0E: Header Type	Read Only: This register reads 0x00 to indicate that the core complies to the standard PCI configuration register mapping and that it is a single function device.
0F	0x0F: BIST	Not implemented. Reads return 0x00.

Addr	Config Register	Register Description
13 – 10	Base Address Register 0	Read/Write: Base Address Register0, Base Address Register1, Base Address Register2, Base Address Register3, Base Address Register4, and Base Address Register5 inform system software of the device's resource requirements and are subsequently programmed to allocate memory and I/O resources to the device.
17 – 14	Base Address Register 1	See Base Address Register 0 description
1B – 18	Base Address Register 2	See Base Address Register 0 description
1F – 1C	Base Address Register 3	See Base Address Register 0 description
23 – 20	Base Address Register 4	See Base Address Register 0 description
27 – 24	Base Address Register 5	See Base Address Register 0 description
2B – 28	Card Bus CIS Pointer	Read Only: Reads return the value of the Cardbus CIS Pointer.
2D – 2C	Subsystem Vendor ID	Read Only: Additional vendor information. Reads return the value of the Subsystem Vendor ID.
2F – 2E	Subsystem ID	Read Only: Additional device information. Reads return the value of the Subsystem ID.
33 – 30	Expansion ROM Base Addr. Reg.	Informs system software of the device's Expansion ROM resource requirements and is subsequently programmed to allocate memory resources to the device. Read/Write: Expansion ROM Base Address Register Bits[31:11] – Written to specify where to locate this region in memory space Bits[10:1] = 0..0 Reserved Bit[0] = Set by S/W to enable decoding the Expansion ROM and clear to disable
34	Capabilities Pointer	Read Only: Reads return 0x40 which is the beginning address of the PCI Express Capabilities Item.
37 – 35	Reserved	Not implemented. Reads return 0x000000.
3B – 38	Reserved	Not implemented. Reads return 0x00000000.
3C	Interrupt Line	Legacy Interrupt support is enabled/disabled by CSR register. When interrupts are enabled, Interrupt Line is read/write and is written by configuration software to indicate to which system interrupt line INTAn is connected. When interrupts are disabled Interrupt Line is read only and returns 0x00.
3D	Interrupt Pin	Interrupt support is enabled/disabled by CSR register. When interrupts are enabled, Interrupt Pin returns 0x01 indicating the core implements INTA# and when interrupts are disabled, Interrupt Pin returns 0x00 indicating no interrupts are used.
3E	Minimum Grant	Read Only: Returns 0x00.
3F	Maximum Latency	Read Only: Returns 0x00.

2.4.2.5. PCI Express Capability

Table 2.34. PCI Express Capability

Addr	Config Register	Register Description
40	PCI Express Capability ID	Read Only = 0x10 (Beginning of PCI Express Capability Item)
41	Next Capability Pointer	Read Only = 0x80 (Pointer to beginning of Power Management Capability)
43-42	PCI Express Capabilities	Read Only • Bits[15:14] – Reserved = 00 • Bits[13:9] – Interrupt Message Number[4:0]; MSI/MSI-X interrupt vector associated with interrupts generated by Configuration Register events (change in link bandwidth, Root Port error, etc.) • Bit[8] – Slot Implemented; Downstream Switch/Root Port only • Bits[7:4] – Device/Port Type – Must match the core application since the value programmed enables/hides Configuration Registers and functionality that is only applicable to some Device/Port types: • 0000 – PCI Express Endpoint • Required for Endpoint applications • 0001 – Legacy PCI Express Endpoint • 0100 – Root Port of PCI Express Root Complex* • Requried for Root Port applications • 0101 – Upstream Port of PCI Express Switch • Required for Upstream Switch Ports • 0110 – Downstream Port of PCI Express Switch • Required for Downstream Switch Ports • 0111 – PCI Express to PCI/PCI-X Bridge • 1000 – PCI/PCI-X to PCI Express Bridge* • 1001 – Root Complex Integrated Endpoint • 1010 – Root Complex Event Collector • Bits[3:0] – Capability Version – Must be 0x2 for PCIe 3.0
47-44	Device Capabilities Register	Read Only • Bits[31:29] – Reserved. • Bit[28] – Function Level Reset Capability • 1 – Capability Present • 0 – Capability Not Present • Bits[27:26] – Captured Slot Power Limit Scale • Bits[25:18] – Captured Slot Power Limit Value • Bits[17:16] = 00. Reserved. • Bit[15] = 1. Role-based Error Reporting • Bit[14] = 0 – Reserved • Bit[13] = 0 – Reserved • Bit[12] = 0 – Reserved • Bits[11:9] – Endpoint L1 Acceptable Latency • Bit[8:6] – Endpoint L0s Acceptable Latency • Bit[5] – Extended Tag Field Supported • Bits[4:3] – Phantom Functions Supported • Bits[2:0] – Max Payload Size Supported • 000 – 128 bytes max payload size • 001 – 256 bytes max payload size • 010 – 512 bytes max payload size • 011 – 1024 bytes max payload size • 100 – 2048 bytes max payload size • 101 – 4096 bytes max payload size

Addr	Config Register	Register Description
		• 110 – Reserved • 111 – Reserved
49-48	Device Control Register	Read/Write • Bit[15] – Bridge Configuration Retry Enable/Initiate Function Level Reset • Bits[14:12] – Max Read Request Size; the Transmit Interface may not transmit a read request TLP with a length larger than the size indicated by Max Read Request Size: • 000 == 128 bytes • 001 == 256 bytes • 010 == 512 bytes • 011 == 1024 bytes • 100 == 2048 bytes • 101 == 4096 bytes • 110 == Reserved • 111 == Reserved • Bit[11] – Enable No Snoop • Bit[10] – Aux Power PM Enable • Bit[9] – Phantom Functions Enable • Bit[8] – Extended Tag Field Enable • Bits[7:5] – Max Payload Size; the Transmit Interface may not transmit a TLP with a payload larger than the size indicated by Max Payload Size: • 000 == 128 bytes • 001 == 256 bytes • 010 == 512 bytes • 011 == 1024 bytes • 100 == 2048 bytes • 101 == 4096 bytes • 110 == Reserved • 111 == Reserved • Bit[4] – Enable Relaxed Ordering • Bit[3] – Unsupported Request Reporting Enable • Bit[2] – Fatal Error Reporting Enable • Bit[1] – Non-Fatal Error Reporting Enable • Bit[0] – Correctable Error Reporting Enable
4B-4A	Device Status Register	Bits[15:4] are Read Only. Bits[3:0] are cleared by writing a 1 to the corresponding bit location. • Bits[15:6] = 0000000000. Reserved • Bit[5] – Transactions Pending • Bit[4] – AUX Power Detected • Bit[3] – Unsupported Request Detected • Bit[2] – Fatal Error Detected • Bit[1] – Non-Fatal Error Detected • Bit[0] – Correctable Error Detected
4F-4C	Link Capabilities Register	Read Only. • Bits[31:24] – Port Number • Bits[22] = 1. ASPM Optionalit Compliance • Bit[21] – Link Bandwidth Notification Capability • == 1 when operating as a Downstream Port; else 0 • Bit[20] – Data Link Layer Active Reporting Capable • == 1 when operating as a Downstream Port; else 0 • Bit[19] – Surprise Down Error Reporting Capable • == 1 when operating as a Downstream Port; else 0 • Bit[18] = 0. Clock Power Management

Addr	Config Register	Register Description
		• Bits[17:15] – L1 Exit Latency • Bits[14:12] – L0s Exit Latency • Bits[11:10] – Active State Power Management (ASPM) Support • 00 – No ASMP Support • 01 – L0s Supported • 10 – L1 Supported • 11 – L0s and L1 Supported • Bits[9:4] – Maximum Link Width • 000001 – x1 • 000010 – x2 • 000100 – x4 • 001000 – x8 • 010000 – x16 • Bits[3:0] – Maximum Link Speed • 0001 (2.5GT/s) • 0010 (5GT/s) • 0011 (8GT/s)
51-50	Link Control Register	Read/Write • Bits[15:12] = 0000. Reserved. • Bit[11] – Link Autonmous Bandwidth Interrupt Enable • Bit[10] – Link Bandwidth Management Interrupt Enable • Bit[9] – Hardware Autonomous Width Disable • Bit[8] = 0. Enable Clock Power Management • Bit[7] – Extended Sync • Bit[6] – Common Clock Configuration • Bit[5] – Retrain Link • Bit[4] – Link Disable • Bit[3] – Read Completion Boundary (RCB) • 0 == 64 bytes • 1 == 128 bytes • Bit[2] = 0. Reserved. • Bits[1:0] – Active State Power Management (ASPM) Control • 00 – Disabled • 01 – L0s Enabled • 10 – L1 Enabled • 11 – L0s and L1 Enabled
53-52	Link Status Register	Read Only • Bit[15] – Link Autonomous Bandwidth Status • Bit[14] – Link Bandwidth Management Status • Bit[13] – Data Link Layer Active • Bit[12] – Slot Clock Configuration • Bit[11] – Link Training • Bit[10] = 0. Reserved. • Bits[9:4] Negotiated Link Width – indicates the number of lanes currently in use • 010000 = x16 • 001000 = x8 • 000100 = x4 • 000010 = x2 • 000001 = x1 • Bits[3:0] Link Speed • 0001 (2.5 GT/s) • 0010 (5.0 GT/s) • 0011 (8.0 GT/s)

Addr	Config Register	Register Description
57-54	Slot Capabilities Root Port/Switch Only	Normally Read Only; Writable when HW Init Write Enable == 1 (see the Vendor-Specific Extended Capability section.) • Bits[31:19] – Physical Slot Number • Bit[18] – No Command Completed Support • Bit[17] – Electromechanical Interlock Present • Bits[16:15] – Slot Power Limit Scale[1:0] • Bits[14:7] – Slot Power Limit Value[7:0] • Bit[6] – Hot-Plug Capable • Bit[5] – Hot-Plug Surprise • Bit[4] – Power Indicator Present • Bit[3] – Attention Indicator Present • Bit[2] – MRL Sensor Present • Bit[1] – Power Controller Present • Bit[0] – Attention Button Present
59-58	Slot Control Root Port/Switch Only	Read Only • Bits[15:13] = 0. Reserved. • Bit[12] – Data Link Layer State Changed Enable • Bit[11] = 0. Electromechanical Interlock Control • Bit[10] – Power Controller Control • Bit[9:8] – Power Indicator Control • Bit[7:6] – Attention Indicator Control • Bit[5] – Hot-Plug Interrupt Enable • Bit[4] – Command Completed Interrupt Enable • Bit[3] – Presence Detect Changed Enable • Bit[2] – MRL Sensor Changed Enable • Bit[1] – Power Fault Detected Enable • Bit[0] – Attention Button Pressed Enable
5b-5a	Slot Status Root Port/Switch Only	Read Only • Bits[15:9] = 0. Reserved. • Bit[8] – Data Link Layer State Changed • Bit[7] – Electromechanical Interlock Status • Bit[6] – Presence Detect State • Bit[5] – MRL Sensor State • Bit[4] – Command Completed • Bit[3] – Presence Detect Changed • Bit[2] – MRL Sensor Changed • Bit[1] – Power Fault Detected • Bit[0] – Attention Button Pressed
5d-5c	Root Control Root Port Only	Read/Write • Bits[15:5] = 0. Reserved. • Bit[4] = CRS Software Visibility Enable • Bit[3] – PME Interrupt Enable • Bit[2] – System Error on Fatal Error Enable • Bit[1] – System Error on Non-Fatal Error Enable • Bit[0] – System Error on Correctable Error Enable
5f-5e	Root Capabilities Root Port Only	Read Only; Bit[16] – Write 1 to clear. • Bits[15:1] = 0. Reserved • Bit[0] = 1. CRS Software Visibility supported.

Addr	Config Register	Register Description
63-60	Root Status Root Port Only	Read Only; Bit[16] – Write 1 to clear. • Bits[31:18] = 0. Reserved • Bit[17] – PME Pending • Bit[16] – PME Status • Bits[15:0] – PME Requester ID
67-64	Device Capabilities 2	Read Only • Bits[31:24] = 0. Reserved • Bits[23:22] = 00. Max End-End TLP Prefixes • Bit[21] = 0. End-End TLP Prefix Supported • Bit[20] = 0. Extended Fmt Field Supported • Bit[19:18] = 00. OBFF Supported • Bits[17:14] = 0000. Reserved • Bits[13:12] = 00. TPH Completer Supported • Bit[11] = LTR Mechanism Supported • Bit[10] = 0. No RO-enabled PR-PR Passing • Bit[9] = 0. 128-bit CAS Completer Supported • Bit[8] = 0. 64-bit AtomicOp Completer Supported • Bit[7] = 0. 32-bit AtomicOp Completer Supported • Bit[6] = 0. AtomicOp Routing Supported • Bit[5] = 0. ARI Forwarding Supported • Bit[4] – Completion Timeout Disable Supported • Bits[3:0] – Completion Timeout Ranges Supported
69-68	Device Control 2	Read/Write • Bit[15] – End-End TLP Prefix Blocking • Bits[14:13] – OBFF Enable; not supported • Bits[12:11] = 00. Reserved. • Bit[10] – LTR Mechanism Enable • Bit[9] – IDO Completion Enable • Bit[8] – IDO Request Enable • Bit[7] – AtomicOp Egress Blocking • Bit[6] – AtomicOp Request Enable • Bit[5] – ARI Forwarding Enable • Bit[4] – Completion Timeout Disable – Set by system software to disable this device from generating completion timeouts. Users must disable completion timeout error generation when this bit is set. • Bits[3:0] – Completion Timeout Value – Set by system software to select the completion timeout range which must be used by users which are implementing completion timeouts. See PCI Express Specification Table 7-24 for details.
6B-6A	Device Status 2	Reserved by PCI SIG for future use. Reads return 0x00000000.
6F-6C	Link Capabilities 2	Read Only • Bits[31:23] – Reserved • Bits[22:16] – Lower SKP OS Reception Supported Speeds Vector • Bits[15:9] – Lower SKP OS Generation Supported Speeds Vector • Bit[8] – Crosslink Supported • Bits[7:1] – Supported Link Speeds Vector • Bit[0] = 0. Reserved
71-70	LinkControl 2	Read/Write • Bit[15:12] – Compliance Preset/De-emphasis[3:0] • Bit[11] – Compliance SOS • Bit[10] – Enter Modified Compliance • Bits[9:7] – Transmit Margin • Bit[6] – Selectable De-emphasis

Addr	Config Register	Register Description
		• Bit[5] – Hardware Autonomous Speed Disable • Bit[4] – Enter Compliance • Bits[3:0] – Target Link Speed[3:0] • 0001 (2.5 GT/s) • 0010 (5.0 GT/s) • 0011 (8.0 GT/s)
73-72	Link Status 2	Read Only; Bit[5] – write 1 to clear: • Bits[15:6] = 0000000000. Reserved. • Bit[5] – Link Equalization Reset • Bit[4] – Equalization Phase 3 Successful • Bit[3] – Equalization Phase 2 Successful • Bit[2] – Equalization Phase 1 Successful • Bit[1] – Equalization Complete • Bit[0] – Current De-emphasis Level • 1==-3.5dB • 0==-6dB
77-74	Slot Capabilities 2 Root Port/Switch Only	Reserved by PCI SIG for future use. Reads return 0x00000000.
79-78	Slot Control 2 Root Port/Switch Only	Reserved by PCI SIG for future use. Reads return 0x00000000.
7b-7a	Slot Status 2 Root Port/Switch Only	Reserved by PCI SIG for future use. Reads return 0x00000000.
7F-7C	Reserved	Reads return 0x00000000.

2.4.2.6. Power Management Capability

Table 2.35. Power Management Capability

Addr	Config Register	Register Description
80	Power Management Capability ID	Read Only = 0x01 (Beginning of Power Management Capability Item)
81	Next Capability Pointer	Read Only. Pointer to next Capability Item in the list.
83-82	Power Management Capabilities	Read Only. • Bits[15:11] – PME Support; recommended default == 0 • Bits[10] – D2 Support (1) Yes (0) No; this bit must be set for the core to allow Power State to be written to D2; recommended default == 0 • Bit[9] – D1 Support (1) Yes (0) No; this bit must be set for the core to allow Power State to be written to D1; recommended default == 0 • Bit[8:6] – Aux Current; recommended default = 0 • Bit[5] – Device Specific Initialization(DSI); recommended default = 0 • Bit[4] – Reserved; set to 0 • Bit[3] – PME Clock; recommended default = 0 • Bits[2:0] – Version; set to 011 (complies with revision 1.2 of the PCI Power Management Interface Specification) See Section 2.1.8 for additional detail.
85-84	Power Management Control/Status	Read/Write. • Bit[15] – PME Status; if Power Management Capabilities[15] == 1 indicating that PME is generated from D3cold, then PME_Status is implemented by the core; otherwise PME_Status == 0. • Bits[14:13] – Data Scale; recommend == 0 (Data not implemented) • Bits[12:9] – Data Select; recommend == 0 (Data not implemented) • Bit[8] – PME En – ; if Power Management Capabilities[15:11] == 0 indicating that PME is not generated from any power state then PME_En == 0; is implemented by the core and written by system software to enable PME generation from D3cold; otherwise PME_En == 0. • Bits[7:4] – Reserved – set to 0 • Bit[3] – No Soft Reset – Core sets to 1 since the core is not reset when transitioning from D3hot to D0 purely due to power state changes. This bit is used by system software to know whether the device needs to be reinitialized when transitioning between D3hot and D0. • Bit[2] – Reserved; set to 0 • Bits[1:0] – Power State; software writes this field to transition a device into a different power state; increasing Dx numbers represent increasingly lower power states • 00 – D0; normal operation • 01 – D1; not allowed to be written unless D1 Support == 1 • 10 – D2; not allowed to be written unless D2 Support == 1 • 11 – D3hot; “off” See Section 2.1.8 for additional detail.
86	PMCSR PCI to PCI Bridge Support	Read Only. • Bit[7] – Bus Power/Clock Control Enable; set to 0 • Bit[6] – B2/B3 Support for D3bat; set to 0 • Bits[5:0] – Reserved; set to 0
87	Data	Read Only. • Bits[7:0] – Data; recommended default = 0; not implmented
8F-88	Reserved	Reads return 0x00000000.

2.4.2.7. MSI-X Capability

Table 2.36. MSI-X Capability

Addr	Config Register	Register Description
90	MSI-X Capability ID	Read Only = 0x11 (Beginning of MSI-X Capability Item) MSI-X Support may be enabled/disabled via CSR registers. If present, its capability is defined as follows. Otherwise, all of the following registers are read 0x0.
91	Next Capability Pointer	Read Only. Pointer to next Capability Item in the list.
93-92	Message Control	Only bits[15:14] are Read/Write. • Bit[15] – MSI-X Enable (Read/Write) • Bit[14] – Function Mask (Read/Write) • Bits[13:11] – Reserved – 000 (Read Only) • Bit[10:0] – Table Size[10:0] (Read Only) • The number of MSI-X vectors requested/supported by the user's design is Table Size + 1.
97-94	Table_Offset, Table_BIR	Read Only. Bits[31:3] – Table_Offset[31:3] • {Table_Offset[31:3], 000} is the offset into the BAR indicated by Table_BIR where the MSI-X Table begins. Bits[2:0] – Table BIR[2:0] • Indicates which BAR location contains the MSI-X Table. In the case of a 64-bit BAR Table BIR indicates the BAR that contains the lower 32-bit address: • 000 – BAR0 • 001 – BAR1 • 010 – BAR2 • 011 – BAR3 • 100 – BAR4 • 101 – BAR5 • 110, 111 – Reserved
9B-98	PBA_Offset, PBA_BIR	Read Only. Bits[31:3] – PBA_Offset[31:3] • Same as Table Offset above, but indicates the location of the PBA (Pending Bit Array). Bits[2:0] – PBA BIR[2:0] • Same as Table BIR above, but indicates the location of the PBA

2.4.2.8. MSI Capability

Table 2.37. MSI Capability

Addr	Config Register	Register Description
9F-9C	Reserved	Reads return 0x00000000.
A0	Message Capability ID	Read Only = 0x05 (Beginning of Message Capability Item); MSI Support is enabled/disabled by CSR registers. If present, its capability is defined as follows. Otherwise, all of the following registers are read 0x0.
A1	Next Capability Pointer	Read Only. Pointer to next Capability Item in the list.
A3-A2	Message Control	• Bits[6:4] and Bit[0] are Read/Write; remainder are Read Only • Bits[15:9] = 0x00. Reserved • Bit[8] = 0. Note per vector masking capable. • Bit[7] – 64-bit Address Capable = 1 (Capable of generating 64-bit messages) • Bits[6:4] – Multiple Message Enable – system software writes the number of allocated messages; 000==1, 001==2, 010==4, 011==8, 100==16, 101==32, 110 Reserved, 111 Reserved • Bits[3:1] – Multiple Message Capable – Number of messages requested by the device == 000 (1 Message) • Bit[0] – MSI Enable – System software sets this bit to enable MSI. When set, the core uses the MSI mechanism instead of the legacy interrupt mechanism to forward user interrupts on mgmt_interrupt to PCI Express
A7-A4	Message Address	Bits[31:2] are Read/Write; Bits[1:0] are Read Only • Bits[31:2] Message Address[31:2] • Bits[1:0] – Reserved – Message Address[1:0] is always 00
AB-A8	Message Upper Address	Read/Write • Bits[31:0] Message Address[63:32]; if Message Address[63:32] == 0, then the core uses only Message Address[31:0] and does 32-bit address MSI writes. If Message Address[63:32] != 0 then the core uses Message Address[63:0] and does 64-bit address MSI writes.
AD-AC	Message Data	Read/Write • Bits[15:0] Message Data[15:0] – An MSI Message is sent by writing Message Data to Message Address.

2.4.2.9. Advanced Error Reporting Extended Capability

Table 2.38. Advanced Error Reporting Extended Capability

Addr	Config Register	Register Description
103-100	Advanced Error Reporting Enhanced Capability Header	Beginning of Advanced Error Reporting (AER) Capability; the AER capability is only present if AER support is enabled in the design, however, AER support is a standard core feature that is present unless AER removal has been specifically requested to be excluded at core deliver time (which is unusual). • Bits[15:0] – Read Only = 0x0001 == AER Capability ID • Bits[19:16] – Read Only = 0x01 == AER Capability Version (PCIe 2.0/1.1) • Bits[31:20] – Read Only. Pointer to next Enhanced/Extended Capability Item in the list.
107-104	Uncorrectable Error Status	Read/Write: Bit set when corresponding error event occurs and the error is not masked by the Uncorrectable Error Mask register; clear set bits by writing a 1: • Bits[3:0] – Reserved == 0 • Bit[4] – DataLink_Protocol_Error_Status • Bit[5] – Surprise_Down_Error_Status • Bits[11:6] – Reserved == 0 • Bit[12] – Poisoned_TLP_Status • Bit[13] – Flow_Control_Protocol_Error_Status • Bit[14] – Completion_Timeout_Status

Addr	Config Register	Register Description
		• Bit[15] – Completer_Abort_Status • Bit[16] – Unexpected_Completion_Status • Bit[17] – Receiver_Overflow_Status • Bit[18] – Malformed_TLP_Status • Bit[19] – ECRC_Error_Status • Bit[20] – Unsupported_Request_Error_Status • Bit[21] – Reserved = 0 • Bit[22]- Uncorrectable Internal Error Status • Bits[31:23] – Reserved ==
10B-108	Uncorrectable Error Mask	Read/Write: Set corresponding bit to mask (not report) selected error events; clear to unmask (report): • Bits[3:0] – Reserved == 0 • Bit[4] – DataLink_Protocol_Error_Mask • Bit[5] – Surprise_Down_Error_ Mask • Bits[11:6] – Reserved == 0 • Bit[12] – Poisoned_TLP_ Mask • Bit[13] – Flow_Control_Protocol_Error_ Mask • Bit[14] – Completion_Timeout_ Mask • Bit[15] – Completer_Abort_ Mask • Bit[16] – Unexpected_Completion_ Mask • Bit[17] – Receiver_Overflow_ Mask • Bit[18] – Malformed_TLP_ Mask • Bit[19] – ECRC_Error_ Mask • Bit[20] – Unsupported_Request_Error_ Mask • Bit[21] – Reserved = 0 • Bit[22]- Uncorrectable Internal Error Mask • Bits[31:23] – Reserved == 0
10F-10C	Uncorrectable Error Severity	Read/Write: Set corresponding bit to mark selected error events as FATAL errors; clear to mark selected error events as NON-FATAL errors: • Bits[3:0] – Reserved == 0 • Bit[4] – DataLink_Protocol_Error_Severity • Bit[5] – Surprise_Down_Error_Severity • Bits[11:6] – Reserved == 0 • Bit[12] – Poisoned_TLP_Severity • Bit[13] – Flow_Control_Protocol_Error_Severity • Bit[14] – Completion_Timeout_Severity • Bit[15] – Completer_Abort_Severity • Bit[16] – Unexpected_Completion_Severity • Bit[17] – Receiver_Overflow_Severity • Bit[18] – Malformed_TLP_Severity • Bit[19] – ECRC_Error_Severity • Bit[20] – Unsupported_Request_Error_Severity • Bit[21] – Reserved = 0 • Bit[22]- Uncorrectable Internal Error Severity • Bits[31:23] – Reserved == 0
113-110	Correctable Error Status	Read/Write: Bit set when corresponding error event occurs and the error is not masked by the Correctable Error Mask register; clear set bits by writing a 1: • Bit[0] – Receiver_Error_Status • Bits[5:1] – Reserved == 0 • Bit[6] – Bad_TLP_Status • Bit[7] – Bad_DLLP_Status • Bit[8] – Replay_Num_Rollover_Status

Addr	Config Register	Register Description
		• Bits[11:9] – Reserved == 000 • Bit[12] – Replay_Timer_Timeout_Status • Bit[13] – Advisory_Non_Fatal_Error_Status • Bit[14] – Corrected Internal Error Status • Bit[15] – Header Log Overflow Status • Bits[31:16] – Reserved == 0
117-114	Correctable Error Mask	Read/Write: Set corresponding bit to mask (not report) selected error events; clear to unmask (report): • Bit[0] – Receiver_Error_Mask • Bits[5:1] – Reserved == 0 • Bit[6] – Bad_TLP_Mask • Bit[7] – Bad_DLLP_Mask • Bit[8] – Replay_Num_Rollover_Mask • Bits[11:9] – Reserved == 000 • Bit[12] = Replay_Timer_Timeout_Mask • Bit[13] = Advisory_Non_Fatal_Error_Mask • Bit[14] – Corrected Internal Error Mask • Bit[15] – Header Log Overflow Mask • Bits[31:16] – Reserved == 0
11B-118	Advanced Error Capabilities and Control	Read/Write: Misc Capabilities and Control • Bits[4:0] = Read Only – First_Error_Pointer[4:0] • Bit[5] = Read Only – ECRC_Generation_Capable • 1==Device is capable of generating ECRC; is set if the core includes ECRC generation logic (non-standard core option) • 0 == Device is not capable of generating ECRC • Bit[6] = Read/Write – ECRC_Generation_Enable • Software sets to control whether ECRCs are generated and inserted for TLPs transmitted by the core; if ECRC support is not implemented in the core, this bit is Read Only == 0 • 1== Generate and insert ECRC for TLPs transmitted by the core • 0 == Do not generate and insert ECRC • Bit[7] = Read Only – ECRC_Check_Capable • 1==Device is capable of checking ECRC; is set if the core includes ECRC generation logic (non-standard core option) • 0 == Device is not capable of checking ECRC • Bit[8] = Read/Write – ECRC_Check_Enable • Software sets to control whether ECRCs are checked for TLPs received by the core; if ECRC support is not implemented in the core, this bit is Read Only == 0 • 1== Check ECRC for all TLPs with ECRC received by the core • 0 == Do not check ECRC • Bits[31:9] – Reserved = 0
12B-11C	Header Log	Header[127:0] of the TLP associated with the error. TLP format is in same order as illustrated in PCIe Specification: • 0x11F-11C : {Byte0, Byte1, Byte2, Byte3} • 0x123-120 : {Byte4, Byte5, Byte6, Byte7} • 0x127-124 : {Byte8, Byte9, Byte10, Byte11} • 0x12B-0x128 : {Byte12, Byte13, Byte14, Byte15}
137-12C	Reserved	Only implemented by AER Root Ports. Reads return 0x00000000.
147-138	Reserved	TLP Prefix Log Register

2.4.2.10. ARI Extended Capability

Table 2.39. ARI Extended Capability

** ARI Is located at offset 0x148 unless AER is not present in which case it is moved to 0x100.

Addr	Config Register	Register Description
14B-148 or 103-100	ARI Capability Extended Capability Header	Beginning of ARI Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list: • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x000E == Capability ID
14D-14C or 105-104	ARI Capability Register	Read Only • Bits[15:8] – Next Function Number = 0 (not implememted) • Bits[7:2] – Reserved = 0 • Bit[1] – ACS Function Groups Capability = 0 (not implemented) • Bit[0] – MFVC Function Groups Capability = 0 (not implemented)
14F-14E or 107-106	ARI Control Register	Read Only • Bits[15:7] – Reserved • Bit[6:4] – Function Group = 0 (not implemented) • Bits[3:2] – Reserved = 0 • Bit[1] – ACS Function Groups Enable = 0 (not implemented) • Bit[0] – MFVC Function Groups Enable = 0 (not implemented)

2.4.2.11. Vendor-Specific Extended Capability

Table 2.40. Vendor-Specific Extended Capability

Addr	Config Register	Register Description
153-150	Vendor-Specifc PCI Express Extended Capability Header	Beginning of Vendor-Specific Extended Capability (VSEC) • Bits[31:20] – Read Only. Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] – Read Only = 0x1 == Capability Version • Bits[15:0] – Read Only = 0x000B == Capability ID
157-154	Vendor-Specific Header	Read Only • Bits[31:20] – VSEC Length = 0x24 (36 bytes) • Bits[19:16] – VSEC Rev = 0x1 • Bits[15:0] – VSEC ID
15B-158	HW Init	Read/Write • Bits[31:1] = 0. Reserved • Bit[0] – HW Init Write Enable – Used to allow software to write some Configuration Registers which are type “HWInit” when they would otherwise not be writable; default value == 0 • 1 – HW Init Write Enabled – Allow specific HW Init fields to be written by software; only relevant for Configuration Registers in this document which specifically state they are writable when HW Init Write Enable == 1 (for example PCI Express Capability: Slot Capabilities) • 0 – HW Init Write Disabled
15F-15C	Link Power Down Root Port / Downstream Switch Port Only	Read/Write – Used by system software in a Root Port or Downstream Switch Port application to cause a PME_Turn_Off Message to be transmitted on PCI Express to request that the downstream PCI Express heirarchy prepare for Power Down • Bits[31:3] = 0. Reserved • Bit[2] – L2 Request Timeout; indicates when an L2 Request completed due to a timeout; L2 Request Timeout is cleared (0) when L2_Request is written to 1 or when 1 is written to this register; L2 Request Timeout is set (1) when a PME_TO_ACK message is not received in response to a transmitted PME_Turn_Off within the expected 100 mS (100 uS for simulation when mgmt_short_sim == 1) timeout window • Bit[1] – L2 Request Status; indicates when an L2 Request has completed either due to receiving the expected PME_TO_Ack message response or due to timeout; L2 Request

Addr	Config Register	Register Description
		Status is cleared (0) when L2_Request is written to 1 or when 1 is written to this register; L2 Request Status is set (1) when a PME_TO_ACK message is received or a timeout occurs • Bit[0] – L2 Request; write to 1 to cause a PME_Turn_Off Message to be transmitted downstream to the PCI Express hierarchy; after all downstream devices have prepared for power-down the core should receive a PME_TO_Ack message in response indicating the downstream PCIe hierarchy is ready for removal of power; L2 Request stays set until a PME_TO_ACK message is received or a timeout occurs
163-160	Autonomous Recovery, Speed, and Width	Read/Write – Used by system software in an US Port to perform autonomous speed change, width change, or entry to recovery. Bits[31:16] – Lane Width Mask. A 1 indicates that the lane can be used. [16] = Lane 0 .. [31] = Lane 15. Bits[15:12] – Reserved Bits[11:8] – Target Speed. (1=2.5G, 2=5G,3=8G,4=16G). Bits[7:3] – Reserved Bit[2] – Autonomous Entry to Recovery Command. When Set to 1, Bits[1] and [0] must both be set to 0. Setting this bit to 1 causes the Link to immediately transition to Recovery. Bit[1] – Autonomous Width Change Command. When set to 1, the Link transitions to Recovery to perform a link width change, using the Lane Width Mask field. This bit is ignored if HW Autonomous Width Disable has been set in the Link Control register. Bit[0] – Autonomous Speed Change Command When set to 1, the Link transitions to Recovery to perform a speed change, using the Target Speed field. This bit is ignored if HW Autonomous Speed Disable has been set in the Link Control 2 register. Speed and Width Changes can be signaled together, however Entry to Recovery must be signaled independently from Speed or Width Changes.
173-164	Reserved	Reserved

2.4.2.12. Secondary PCI Express Extended Capability

Table 2.41. Secondary PCI Express Extended Capability

Addr	Config Register	Register Description
183-180	Secondary PCI Express Extended Capability Header	Beginning of Secondary PCI Express Extended Capability; this capability is only present if PCIe 3.0 support is enabled in the design. If the AER capability is not present, then this capability is located at offset 0x100 instead. Bits[31:20] – Read Only. Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] – Read Only = 0x1 == Capability Version • Bits[15:0] – Read Only = 0x0019 == Capability ID
187-184	Link Control 3	Read/Write • Bits[31:16] – Reserved = 0 • Bits[15:9] – Enable Lower SKP OS Generation Vector • Bits[8:2] – Reserved = 0 • Bit[1] – Link Equalization Request Interrupt Enable • Bit[0] – Perform Equalization
18B-188	Lane Error Status	Read Only: Indicates lane-specific error status • Bits[31:NUM_LANES] – Reserved = 0 • Bit[Lane#] – 1 == Error detected on lane[[Lane#]; 0 == no error
1AB-18C	Lane Equalization Control Register	Read Only: Control and status fields for link equalization; 16-bits per lane starting with Lane[0] with higher lane #s at higher addresses Per lane format is as follows: • Bit[15] – Reserved = 0 • Bits[14:12] – Upstream Port Receiver Preset Hint • Bits[11:8] –Upstream Port Transmitter Preset • Bit[7] – Reserved = 0 • Bits[6:4] – Downstream Port Receiver Preset Hint • Bits[3:0] –Downstream Port Transmitter Preset

2.4.2.13. ATS Extended Capability

Table 2.42. ATS Extended Capability

Addr	Config Register	Register Description
203-200	ATS Capability Extended Capability Header	Beginning of ATS Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x000F == Capability ID
205-204	ATS Capability Register	Read Only • Bits[4:0] – Invalidate Queue Depth • Bit[5]Page Aligned Request • Bits[15:6] – Reserved
207-206	ATS Control Register	Read/Write • Bits[4:0] – Smallest Translation Unit • Bits[14:5] – Reserved • Bit[15] – Enable

2.4.2.14. DSN Extended Capability

Table 2.43. DSN Extended Capability

Addr	Config Register	Register Description
213-210	DSN Capability Extended Capability Header	Beginning of DSN Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x0003 == Capability ID
21B-214	DSN Serial NUmber	Read Only • Bits[63:0] – DSN Serial Number

2.4.2.15. Resizable BAR Capability

Table 2.44. Resizable BAR Capability

Addr	Config Register	Register Description
283-280	Resizable BAR Extended Capability Header	Resizable BAR Capability Header – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x0015 == Capability ID
287-284	Resizable BAR Capability(0)	Read Only • Bits[31:24] – Reserved • Bits[23:4] – When Bit n is Set, The BAR Indicated by the BAR Index in the Control Register operates with BAR sized to 2^(n+16) Bytes. For example, bit[4] = 2^20 Bytes = 1MB • Bits[3:0] – Reserved
28B-288	Resizable Bar Control Register(0)	Read Only • Bits[31:13] – Reserved R/W • Bits[12:8] – BAR Size. Encoded Value for the Size this BAR should use. Read Only • Bits[7:5] – Number of Resizable BARs. Value must be between 1 and 6. These bits are only valid in the Resizable BAR Control Register (0). In Control Registers (1) or higher, these bits are Reserved. • Bits[2:0] – BAR Index for this BAR: 0 = BAR located at offset 0x10 1 = BAR located at offset 0x14 2 = BAR located at offset 0x18 3 = BAR located at offset 0x1C 4 = BAR located at offset 0x20 5 = BAR located at offset 0x24 Other values reserved. For a 64-bit BAR, this index should point to the lower DWORD.
2BF-28C	Resizable BAR Capability and Control Registers (1..6)	See Resizable BAR Capability (0) See Resizable Bar Control Register(0) The number of Implemented BAR Capability and Control Registers depends on the setting of “Number of Resizable BARs” Control Register (0).

2.4.2.16. Power Budgeting Capability

Table 2.45. Power Budgeting Capability

Addr	Config Register	Register Description
393-390	Power Budgeting Capability Extended Capability Header	Beginning of Power Budgeting Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x0004 == Capability ID
394	Data Select Register	Read/Write • Bits[7:0] – Data Select Register
397-395	Reserved	Reserved
39B-398	Data Register	Read Only • Bits[31:21] – Reserved • Bits[20:18] – Power Rail (0:12V,1:3.3V,2:1.5/1.8V,7:Thermal) • Bits[17:15] – Type (0:PME Aux,1:Aux,2:Idle,3:Sustained,7:Max) • Bits[14:13] – PM State (0:D0,1:D1,2:D2,3:D3) • Bits[12:10] – PM Sub State (0:Default,others: Device Specific) • Bits[9:8] – Data Scale (0:1x,1:0.1x,2:0.01x,3:0.001x) • Bits[7:0] – Base Power
39C	Capabilities Register	Read Only • Bits[7:1] – Reserved • Bit[0] – System Allocated : Set to 1 to indicate that the Power Budget Should be System Allocated, and the values from the Data Register should NOT be used for System Power Budgeting. Set to 0 to indicate that the values provided in the Data Register should be used for System Power Budgeting.

2.4.2.17. Dynamic Power Allocation Capability

Table 2.46. Dynamic Power Allocation (DPA) Capability

Addr	Config Register	Register Description
3A3-3A0	DPA Capability Extended Capability Header	Beginning of DPA Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x0016 == Capability ID
3A7-3A4	DPA Capability Register	Read Only • Bits[31:24] – Transition Latency Value1 (xlcy1) • Bits[23:16] – Transition Latency Value0 (xlcy0) • Bits[15:14] – Reserved • Bits[13:12] – Power Allocation Scale (PAS) • Bits[11:10] – Reserved • Bits[9:8] – Transition Latency Unit (tlunit) • Bits[7:5] –Reserved • Bits[4:0] – Substate_Max
3AB-3A8	DPA Latency Indicator Register	Read Only • Bits[31:Substate_Max+1] – Reserved • Bits[**Substate_Max**:0] – Transition Latency Indicator Bits
3AD-3AC	DPA Status Register	Read Only • Bits[15:9] – Reserved Read, Write 1 to Clear • Bits[8] – Substate Control Enabled Read Only • Bits[7:0] – Substate Status

Addr	Config Register	Register Description
3EF-3AE	DPA Control Register	Read Only • Bits[15:5] – Reserved Read/Write • Bits[4:0] – Substate Control
3CF-3B0	DPA Power Allocation Array	Read Only • Bits[7:0] – Substate Power Allocation Register Address 3B0 is for Substate 0 Address 3B1 is for Substate 1, etc.. up to Substate **Substate_Max**

2.4.2.18. L1 PM Substates Extended Capability

Table 2.47. L1 PM Substates Extended Capability

Addr	Config Register	Register Description
3D3-3D0	L1 PM Substates Capability Extended Capability Header	Beginning of L1 PM Substates Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x001E == Capability ID
3D7-3D4	L1 PM Substates Capabilities Register	HwInit • Bits[31:24] – Reserved • Bits[23:19] – Port T_{POWER_ON} Value • Bits [18] – Reserved • Bits[17:16] – Port T_{POWER_ON} Scale • Bits[15:8] – Port Common_Mode_Restore_Time (in µs) • Bits[7:5] – Reserved • Bit[4] – L1 PM Substates Supported • Bit[3] – ASPM L1.1 Supported • Bit[2] – ASPM L1.2 Supported • Bit[1] – PCI-PM L1.1 Supported • Bit[0] – PCI-PM L1.2 Supported
3DB-3D8	L1 PM Substates Control 1 Register	Read/Write • Bits[31:29] – LTR_L1.2_THRESHOLD_Scale • Bits[28:26] – Reserved • Bits[25:16] – LTR_L1.2_THRESHOLD_Value • Bits[15:8] – Common_Mode_Restore_Time • Bits[7:4] – Reserved • Bit[3] – ASPM L1.1 Enable • Bit[2] – ASPM L1.2 Enable • Bit[1] – PCI-PM L1.1 Enable • Bit[0] – PCI-PM L1.2 Enable
3DF-3DC	L1 PM Substates Control 2 Register	Read/Write • Bits[31:8] – Reserved • Bits[7:3] – T_{POWER_ON} Value • Bit[2] – Reserved • Bits[1:0] – T_{POWER_ON} Scale

2.4.2.19. Latency Tolerance Reporting Capability

Table 2.48. Latency Tolerance Reporting (LTR) Capability

Addr	Config Register	Register Description
3E3-3E0	LTR Capability Extended Capability Header	Beginning of LTR Extended Capability – Read Only • Bits[31:20] – Pointer to next Enhanced/Extended Capability Item in the list. • Bits[19:16] = 0x1 == Capability Version • Bits[15:0] = 0x0018 == Capability ID
3E5-3E4	Max Snoop Latency Register	R/W • Bits[15:13] – Reserved • Bits[12:10] – Max Snoop LatencyScale • Bits[9:0] – Max Snoop LatencyValue
3E7-3E6	Max No-Snoop Latency Register	R/W • Bits[15:13] Reserved • Bits[12:10] – Max No-Snoop LatencyScale • Bits[9:0] – Max No-Snoop LatencyValue

2.4.3. Soft IP Configuration, Control and Status Registers

These registers are accessible through the selected register interface (APB or AHB-Lite) with a base offset of 0x28000.

2.4.3.1. pcie_csr_baseadr Register 0x0

Configuration, Control and Status Register Base Address.

Field	Name	Access	Width	Reset
[31:19]	baddr	read-write	13	PCIE_CSR_BASEADR
[18:0]	*Reserved*	read-only	19	0x0

baddr

This is the upper 13 bits of the base address (512 KiB aligned) used to access both Hard IP and Soft IP registers as well as PCIe Configuration Space Register. Applicable only when the register interface is APB or AHB-Lite.

2.4.3.2. pcie_softip_rst Register 0x4

Configuration, Control and Status Register Base Address.

Field	Name	Access	Width	Reset
[31]	hold_softip_rst	read-write	1	0x0
[30:1]	*Reserved*	read-only	30	0x0
[0]	softip_rst	read-write	1	0x0

hold_softip_rst

Controls the softip_rst if it remains asserted or automatically deassert.

- 0 – writing 1 to softip_rst field toggles the PCIe Soft IP block reset for 1 clock cycle
- 1 – Hold the softip_rst (softip_rst does not automatically deassert unless 0 is written to the corresponding field)

softip_rst

This field controls the PCIe Soft IP block reset. This reset does not cause a PCIe link down as it only affects the Soft logic/application logic on top of the transaction layer. The behaviour of softip_rst depends on the hold_softip_rst field.

- 0 – Deassert Soft IP Reset (Normal operation)
- 1 – Assert Soft IP Reset

2.4.3.3. dma_support_reg0 Register 0xC

Descriptor Queue Base Address.

Field	Name	Access	Width	Reset
[31:12]	desc_baseadr	read-write	20	DESC_QUEUE_BASEADR
[11:0]	*Reserved*	read-only	12	0x0

desc_baseadr

This is the upper 20 bits of the descriptor queue base address (4 KiB aligned).

2.4.3.4. dma_support_reg1 Register 0x10

Status Queue Base Address.

Field	Name	Access	Width	Reset
[31:12]	stat_baseadr	read-write	20	STAT_QUEUE_BASEADR
[11:0]	*Reserved*	read-only	12	0x0

stat_baseadr

This is the upper 20 bits of the status queue base address (4 KiB aligned).

2.4.3.5. dma_support_reg2 Register 0x14

Descriptor and Status Queue Size.

Field	Name	Access	Width	Reset
[31:25]	*Reserved*	read-only	7	0x0
[24:16]	statq_size	read-write	9	STAT_QUEUE_SIZE
[15:9]	*Reserved*	read-only	7	0x0
[8:0]	descq_size	read-write	9	DESC_QUEUE_SIZE

statq_size

Status Queue Size. Indicates the maximum number of status entries. Valid values are from 2 to 256.

descq_size

Descriptor Queue Size. Indicates the maximum number of descriptor entries. Valid values are from 2 to 256.

2.4.3.6. dma_support_reg3 Register 0x18

Descriptor Queue Pointers.

Field	Name	Access	Width	Reset
[31:24]	*Reserved*	read-only	8	0x0
[23:16]	desc_rptr	read-only	8	0x0
[15:8]	*Reserved*	read-only	8	0x0
[7:0]	desc_wptr	read-write	8	0x0

desc_rptr

Descriptor Read Pointer. This field is updated by the Core whenever a descriptor is fetched from the Descriptor Queue. The DMA master should monitor this field to avoid an overflow condition. Pointers should rollover to 0 when it reaches the maximum value (i.e. desc_rptr == (descq_size-1)).

desc_wptr

Descriptor Write Pointer. This field is updated by DMA master whenever new entries are posted in the Descriptor Queue. Pointers should rollover to 0 when it reaches the maximum value (i.e. desc_wptr == (descq_size-1)).

The empty condition is when the desc_rptr and desc_wptr have the same value. The full condition is when the desc_wptr is 1 less than the desc_rptr. The DMA master should make sure that the maximum number of entries in the queue is always 1 less than the maximum descriptor queue size (descq_size-1) (i.e. desc_wptr +1 != desc_rptr).

2.4.3.7. dma_support_reg4 Register 0x1C

Status Queue Pointers.

Field	Name	Access	Width	Reset
[31:24]	*Reserved*	read-only	8	0x0
[23:16]	stat_wptr	read-only	8	0x0
[15:8]	*Reserved*	read-only	8	0x0
[7:0]	stat_rptr	read-write	8	0x0

stat_wptr

Status Write Pointer. This field is updated by the Core after posting an entry in the Status Queue. The DMA master should monitor this field to avoid blocking of Descriptor processing due to Status Queue full condition.

Pointers should rollover to 0 when it reaches the maximum value (i.e. stat_wptr == (statq_size-1)).

stat_rptr

Status Read Pointer. This field is updated by DMA master after processing a status entry. Status entries should be processed immediately so as not to block the processing of Descriptor entries due to Status Queue full condition. Pointers should rollover to 0 when it reaches the maximum value (i.e. stat_rptr == (statq_size-1)).

2.4.3.8. interrupt_type Register 0x20

Interrupt Type.

Field	Name	Access	Width	Reset
[31:7]	*Reserved*	read-only	25	0x0
[6]	stat_full	read-write	1	0x0
[5]	stat_empty	read-write	1	0x0
[4]	stat_updated	read-write	1	0x0
[3]	*Reserved*	read-only	1	0x0
[2]	desc_full	read-write	1	0x0
[1]	desc_empty	read-write	1	0x0
[0]	desc_updated	read-write	1	0x0

stat_full

Classifies the stat_full interrupt status as normal or critical.

- 0 – normal
- 1 – critical

int_normal interrupt port asserts when any of the enabled and unmasked interrupt status with normal type is asserted.

int_critical interrupt port asserts when any of the enabled and unmasked interrupt status with critical type is asserted.

stat_empty

Classifies the stat_empty interrupt status as normal or critical.

- 0 – normal
- 1 – critical

stat_updated

Classifies the stat_updated interrupt status as normal or critical.

- 0 – normal
- 1 – critical

desc_full

Classifies the desc_full interrupt status as normal or critical.

- 0 – normal
- 1 – critical

desc_empty

Classifies the desc_empty interrupt status as normal or critical.

- 0 – normal
- 1 – critical

desc_updated

Classifies the desc_updated interrupt status as normal or critical.

- 0 – normal
- 1 – critical

2.4.3.9. interrupt_enable Register 0x24

Interrupt Enable.

Field	Name	Access	Width	Reset
[31:7]	*Reserved*	read-only	25	0x0
[6]	stat_full	read-write	1	0x0
[5]	stat_empty	read-write	1	0x0
[4]	stat_updated	read-write	1	0x0
[3]	*Reserved*	read-only	1	0x0
[2]	desc_full	read-write	1	0x0
[1]	desc_empty	read-write	1	0x0
[0]	desc_updated	read-write	1	0x0

stat_full

Enables the stat_full interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

stat_empty

Enables the stat_empty interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

stat_updated

Enables the stat_updated interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

desc_full

Enables the desc_full interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

desc_empty

Enables the desc_empty interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

desc_updated

Enables the desc_updated interrupt status. Disabled interrupt status is tied to 0.

- 0 – disable
- 1 – enable

2.4.3.10. interrupt_mask Register 0x28

Interrupt Mask.

Field	Name	Access	Width	Reset
[31:7]	*Reserved*	read-only	25	0x0
[6]	stat_full	read-write	1	0x0
[5]	stat_empty	read-write	1	0x0
[4]	stat_updated	read-write	1	0x0
[3]	*Reserved*	read-only	1	0x0
[2]	desc_full	read-write	1	0x0
[1]	desc_empty	read-write	1	0x0
[0]	desc_updated	read-write	1	0x0

stat_full

Masks the stat_full interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

stat_empty

Masks the stat_empty interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

stat_updated

Masks the stat_updated interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

desc_full

Masks the desc_full interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

desc_empty

Masks the desc_empty interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

desc_updated

Masks the desc_updated interrupt status. A masked status does not cause the assertion of interrupt port.

- 0 – disable
- 1 – enable

2.4.3.11. interrupt_status Register 0x2C

Interrupt Status.

Field	Name	Access	Width	Reset
[31:7]	*Reserved*	read-only	25	0x0
[6]	stat_full	read-write1ToClear	1	0x0
[5]	stat_empty	read-write1ToClear	1	0x0
[4]	stat_updated	read-write1ToClear	1	0x0
[3]	*Reserved*	read-only	1	0x0
[2]	desc_full	read-write1ToClear	1	0x0
[1]	desc_empty	read-write1ToClear	1	0x0
[0]	desc_updated	read-write1ToClear	1	0x0

stat_full

Status Queue Full.

- 0 – otherwise
- 1 – indicates that status queue is full

stat_empty

Status Queue Empty.

- 0 – otherwise
- 1 – indicates that status queue is empty

stat_updated

Status Queue Updated.

- 0 – otherwise
- 1 – indicates that stat_wptr value has changed

desc_full

Descriptor Queue Full.

- 0 – otherwise
- 1 – indicates that descriptor queue is full

desc_empty

Descriptor Queue Empty.

- 0 – otherwise
- 1 – indicates that descriptor queue is empty

desc_updated

Descriptor Queue Updated.

- 0 – otherwise
- 1 – indicates that desc_rptr value has changed

2.4.3.12. pcie_functions Register 0x40

PCIe Functions Enabled.

Field	Name	Access	Width	Reset
[31:3]	*Reserved*	read-only	29	0x0
[2:0]	num_funcs_enabled	read-write	3	NUM_FUNCTIONS

num_funcs_enabled

Indicates the number of enabled physical functions.

2.4.3.13. ahbl_bus_config Register 0x44

AHB-Lite Bus Configuration.

Field	Name	Access	Width	Reset
[31:7]	*Reserved*	read-only	25	0x0
[6:0]	ahbl_max_burst	read-write	7	0x8

ahbl_max_burst

(One-hot) Indicates the maximum burst length allowed in AHB-Lite bus. Large request that exceeds the maximum burst setting is completed as multiple transactions.

- 7’h01 – 4 beats
- 7’h02 – 8 beats
- 7’h04 – 16 beats
- 7’h08 – 32 beats
- 7’h10 – 64 beats
- 7’h20 – 128 beats
- 7’h40 – 256 beats

3. IP Generation

This section provides information on how to generate and customize the Lattice PCIe X1 Core using the IP generation wizard of the Lattice Radiant Software. For more details on the Lattice Radiant Software, refer to the Lattice Radiant Software User Guide.

3.1. Licensing the IP Core

An IP core-specific license is required to enable full use of the Lattice PCIe X1 Core. The IP Core can be evaluated through functional simulation and implementation (synthesis, map, place and route) without an IP license. This IP core supports Lattice's IP hardware evaluation capability, which makes it possible to generate the IP core and operate in hardware for a limited time (approximately four hours) without requiring an IP license. See Hardware Evaluation section for further details. However, a license is required to enable timing simulation and to generate bitstream file that does not include the hardware evaluation timeout limitation.

3.2. Getting Started

The Lattice PCIe X1 Core is available for download from the Lattice IP server through the IP Catalog of Lattice Radiant Software.

1. Open Lattice Radiant Software Program and create a new project. (Refer to Lattice Radiant Software User Guide for details)
2. Select the **IP Catalog** tab then select **IP on Server**.
3. Select **PCIe_Genpoint** and install. After installation, the IP should be on the RadiantIPLocal directory and can be seen in the **IP on Local** tab.

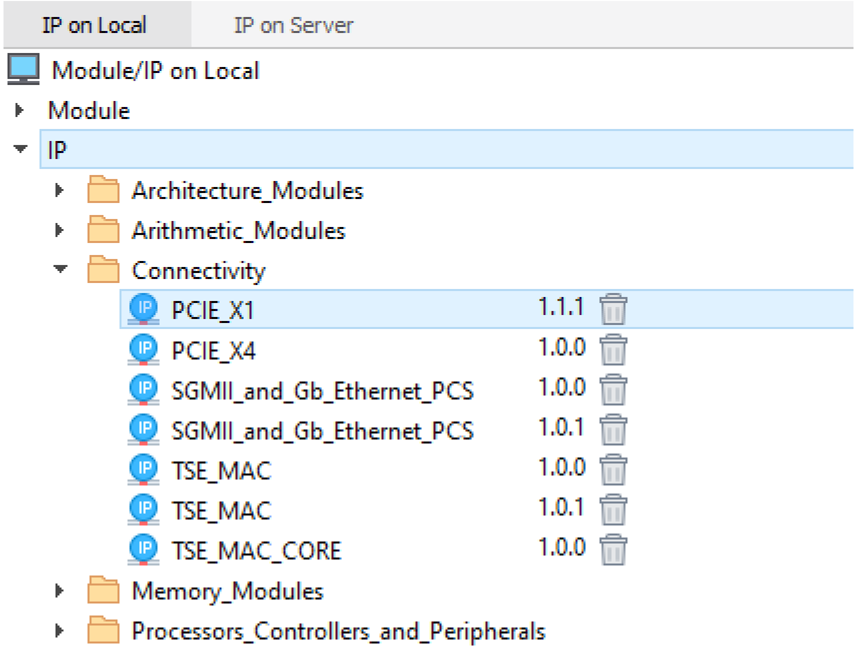
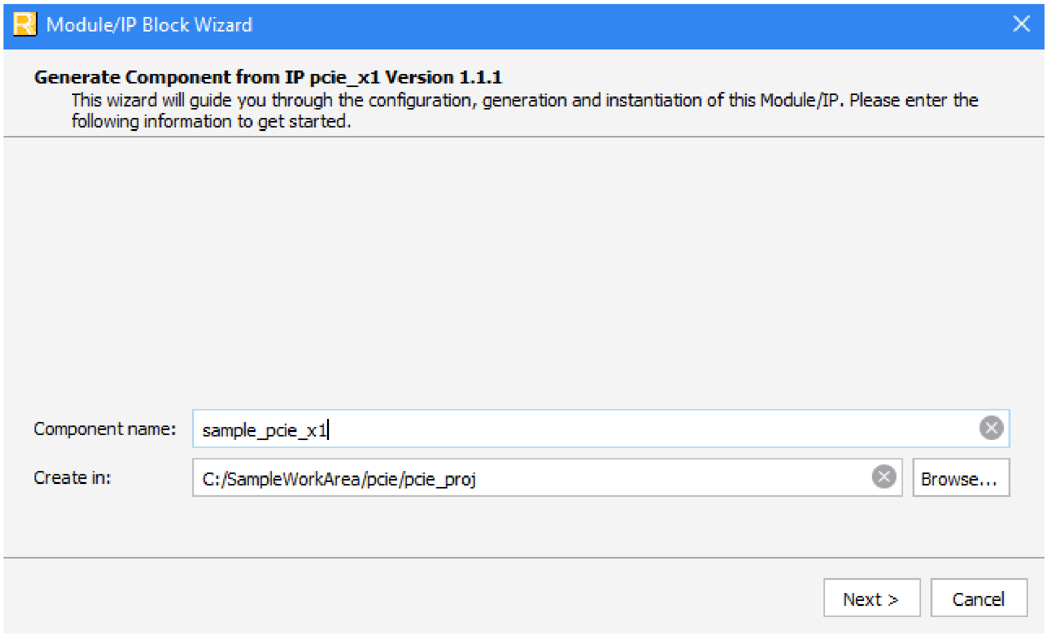


Figure 3.1. Select PCIe X1 IP

4. Double-click on *PCIe_Endpoint* to open the **Module/IP Block Wizard**.
5. Fill out the required information on the dialog box (such as component name and directory) and click **Next**.



Generate Component from IP pcie_x1 Version 1.1.1

This wizard will guide you through the configuration, generation and instantiation of this Module/IP. Please enter the following information to get started.

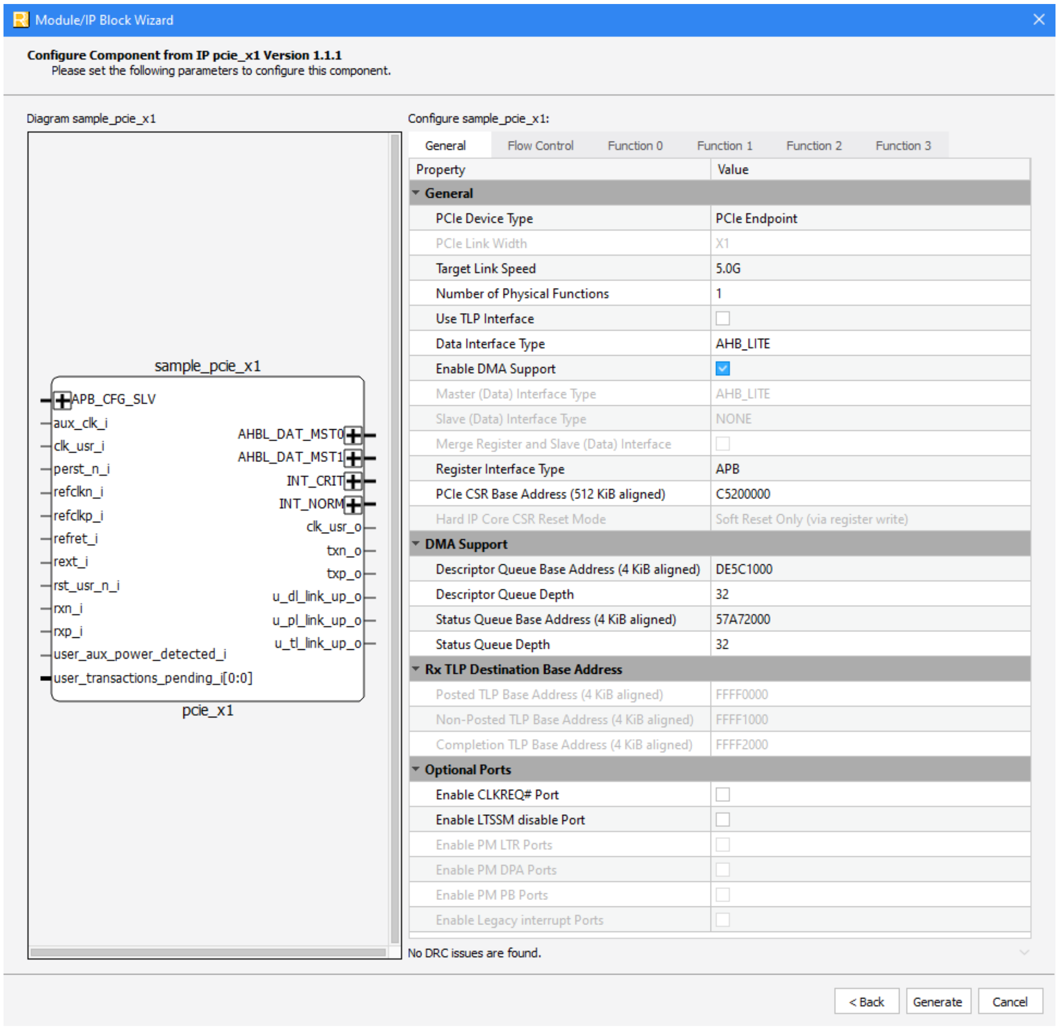
Component name: sample_pcie_x1

Create in: C:/SampleWorkArea/pcie/pcie_proj

Figure 3.2. Configure Module/IP Block Wizard

3.3. Configuring the IP Core

Figure 3.3 shows the IP Configuration interface where you can select and set the IP parameters.



Configure Component from IP pcie_x1 Version 1.1.1
Please set the following parameters to configure this component.

Configure sample_pcie_x1:

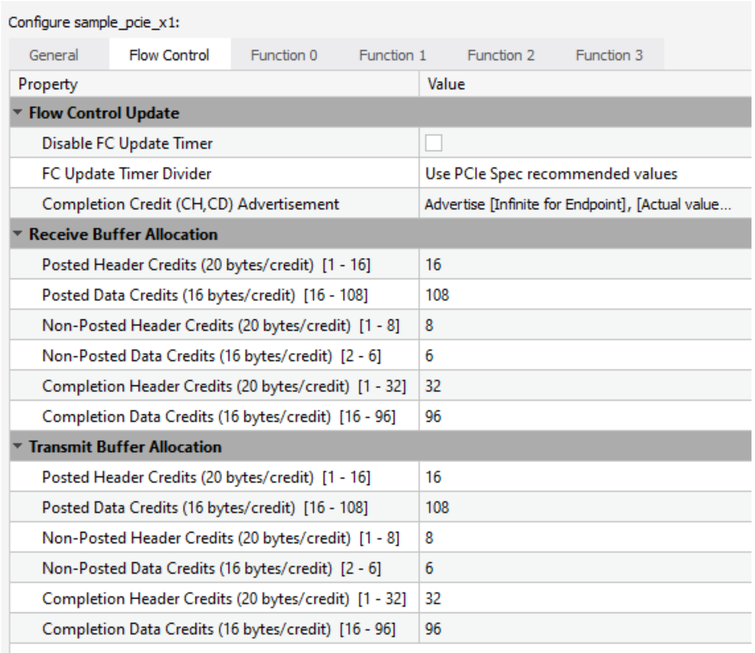
Property	Value
General	
PCIe Device Type	PCIe Endpoint
PCIe Link Width	X1
Target Link Speed	5.0G
Number of Physical Functions	1
Use TLP Interface	☐
Data Interface Type	AHB_LITE
Enable DMA Support	☑
Master (Data) Interface Type	AHB_LITE
Slave (Data) Interface Type	NONE
Merge Register and Slave (Data) Interface	☐
Register Interface Type	APB
PCIe CSR Base Address (512 KiB aligned)	C5200000
Hard IP Core CSR Reset Mode	Soft Reset Only (via register write)
DMA Support	
Descriptor Queue Base Address (4 KiB aligned)	DE5C1000
Descriptor Queue Depth	32
Status Queue Base Address (4 KiB aligned)	57A72000
Status Queue Depth	32
Rx TLP Destination Base Address	
Posted TLP Base Address (4 KiB aligned)	FFFF0000
Non-Posted TLP Base Address (4 KiB aligned)	FFFF1000
Completion TLP Base Address (4 KiB aligned)	FFFF2000
Optional Ports	
Enable CLKREQ# Port	☐
Enable LTSSM disable Port	☐
Enable PM LTR Ports	☐
Enable PM DPA Ports	☐
Enable PM PB Ports	☐
Enable Legacy interrupt Ports	☐

No DRC issues are found.

Figure 3.3. Lattice PCIe X1 Core Configuration User Interface (General Tab)

You may configure the IP by setting the user interface attributes applicable to their application. You may also configure the CSR through the register interface. It is recommended to use the Configuration interface to ensure that only valid parameter values are set. The details of attributes and settings are described in 2.3.

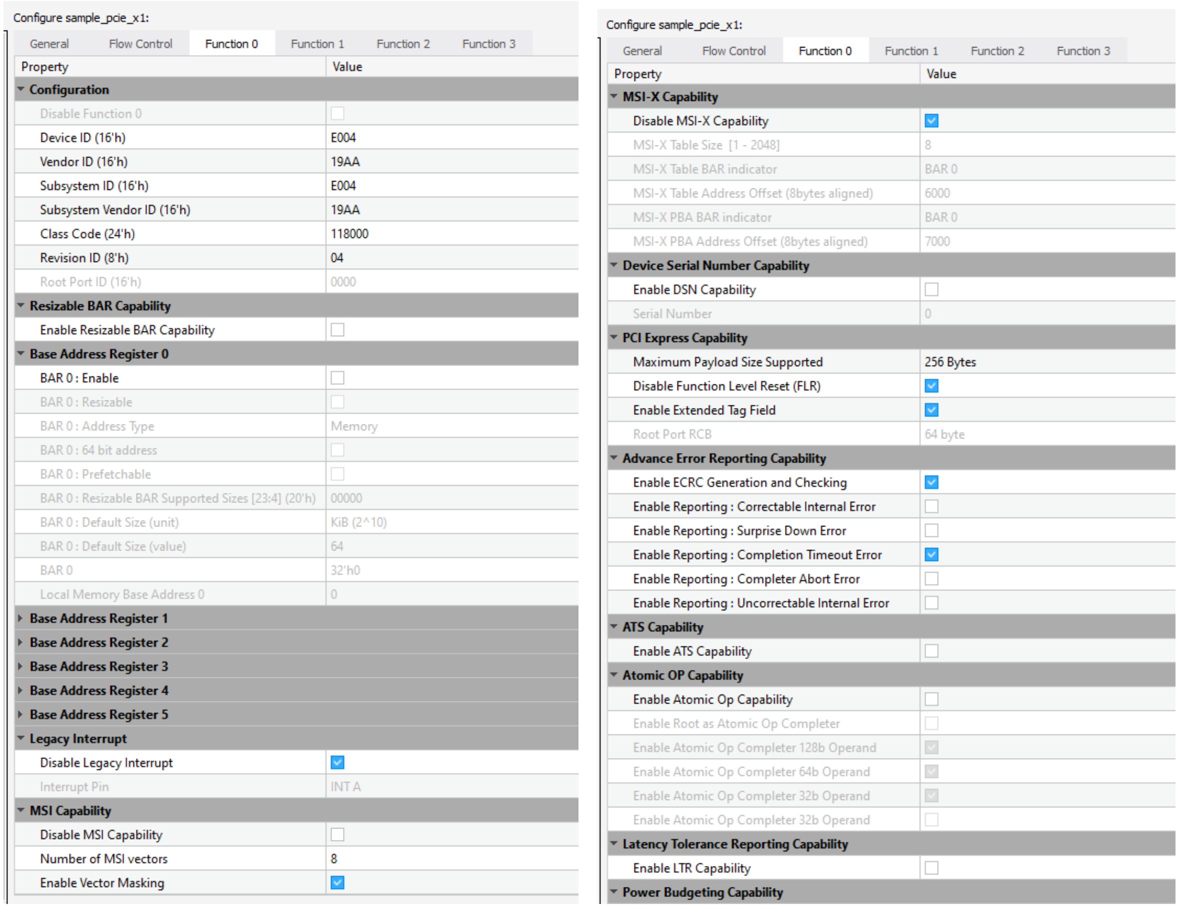
Figure 3.4 shows the configuration options in **Flow Control** tab and *Function 0* tab.



Configure sample_pcie_x1:

Property	Value
Flow Control Update	
Disable FC Update Timer	□
FC Update Timer Divider	Use PCIe Spec recommended values
Completion Credit (CH,CD) Advertisement	Advertise [Infinite for Endpoint], [Actual value…
Receive Buffer Allocation	
Posted Header Credits (20 bytes/credit) [1 - 16]	16
Posted Data Credits (16 bytes/credit) [16 - 108]	108
Non-Posted Header Credits (20 bytes/credit) [1 - 8]	8
Non-Posted Data Credits (16 bytes/credit) [2 - 6]	6
Completion Header Credits (20 bytes/credit) [1 - 32]	32
Completion Data Credits (16 bytes/credit) [16 - 96]	96
Transmit Buffer Allocation	
Posted Header Credits (20 bytes/credit) [1 - 16]	16
Posted Data Credits (16 bytes/credit) [16 - 108]	108
Non-Posted Header Credits (20 bytes/credit) [1 - 8]	8
Non-Posted Data Credits (16 bytes/credit) [2 - 6]	6
Completion Header Credits (20 bytes/credit) [1 - 32]	32
Completion Data Credits (16 bytes/credit) [16 - 96]	96

Figure 3.4. Lattice PCIe X1 Core Configuration Interface (Flow Control Tab)



Configure sample_pcie_x1:

Property	Value
Configuration	
Disable Function 0	□
Device ID (16'h)	E004
Vendor ID (16'h)	19AA
Subsystem ID (16'h)	E004
Subsystem Vendor ID (16'h)	19AA
Class Code (24'h)	118000
Revision ID (8'h)	04
Root Port ID (16'h)	0000
Resizable BAR Capability	
Enable Resizable BAR Capability	□
Base Address Register 0	
BAR 0 : Enable	□
BAR 0 : Resizable	□
BAR 0 : Address Type	Memory
BAR 0 : 64 bit address	□
BAR 0 : Prefetchable	□
BAR 0 : Resizable BAR Supported Sizes [23:4] (20'h)	00000
BAR 0 : Default Size (unit)	KiB (2^10)
BAR 0 : Default Size (value)	64
BAR 0	32'h0
Local Memory Base Address 0	0
Base Address Register 1	
Base Address Register 2	
Base Address Register 3	
Base Address Register 4	
Base Address Register 5	
Legacy Interrupt	
Disable Legacy Interrupt	☑
Interrupt Pin	INT A
MSI Capability	
Disable MSI Capability	□
Number of MSI vectors	8
Enable Vector Masking	☑

Configure sample_pcie_x1:

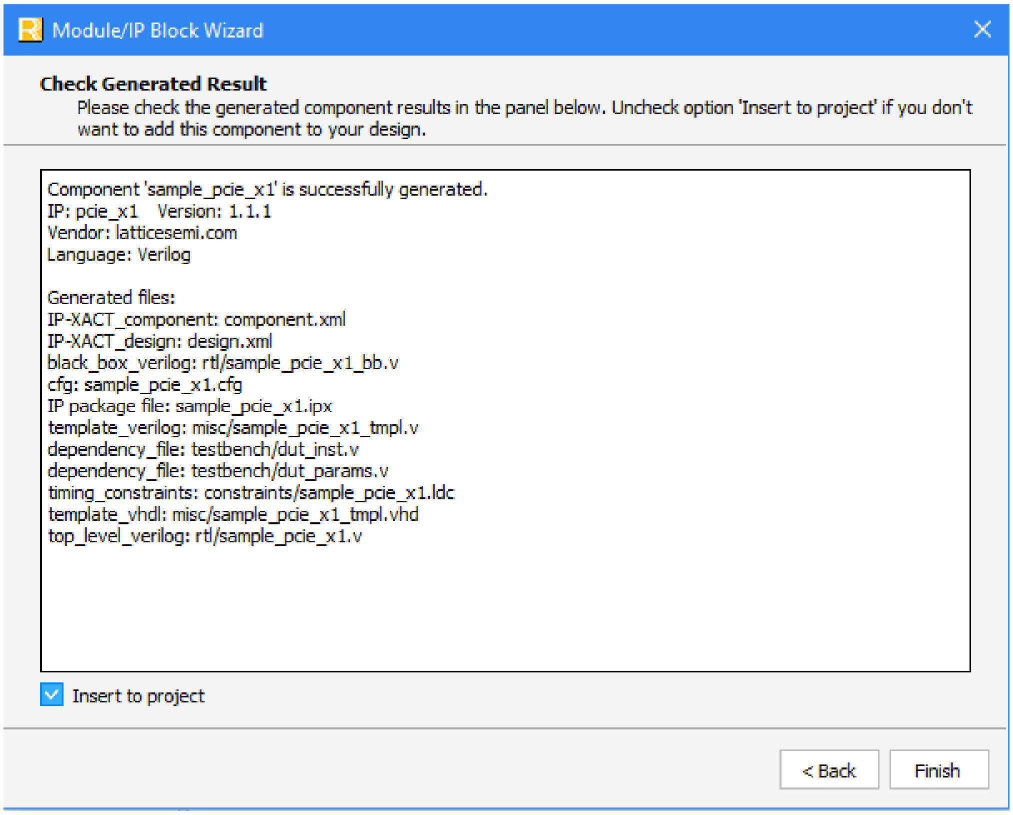
Property	Value
MSI-X Capability	
Disable MSI-X Capability	☑
MSI-X Table Size [1 - 2048]	8
MSI-X Table BAR indicator	BAR 0
MSI-X Table Address Offset (8bytes aligned)	6000
MSI-X PBA BAR indicator	BAR 0
MSI-X PBA Address Offset (8bytes aligned)	7000
Device Serial Number Capability	
Enable DSN Capability	□
Serial Number	0
PCI Express Capability	
Maximum Payload Size Supported	256 Bytes
Disable Function Level Reset (FLR)	☑
Enable Extended Tag Field	☑
Root Port RCB	64 byte
Advance Error Reporting Capability	
Enable ECRC Generation and Checking	☑
Enable Reporting : Correctable Internal Error	□
Enable Reporting : Surprise Down Error	□
Enable Reporting : Completion Timeout Error	☑
Enable Reporting : Completer Abort Error	□
Enable Reporting : Uncorrectable Internal Error	□
ATS Capability	
Enable ATS Capability	□
Atomic OP Capability	
Enable Atomic Op Capability	□
Enable Root as Atomic Op Completer	□
Enable Atomic Op Completer 128b Operand	☑
Enable Atomic Op Completer 64b Operand	☑
Enable Atomic Op Completer 32b Operand	☑
Enable Atomic Op Completer 32b Operand	□
Latency Tolerance Reporting Capability	
Enable LTR Capability	□
Power Budgeting Capability	

Figure 3.5. Lattice PCIe X1 Core Configuration User Interface (Function 0 Tab)

3.4. Instantiating the IP Core

To instantiate the IP Core:

1. After configuring the IP Core, the click the **Generate** button.
2. Click the **Finish** button.



Check Generated Result

Please check the generated component results in the panel below. Uncheck option 'Insert to project' if you don't want to add this component to your design.

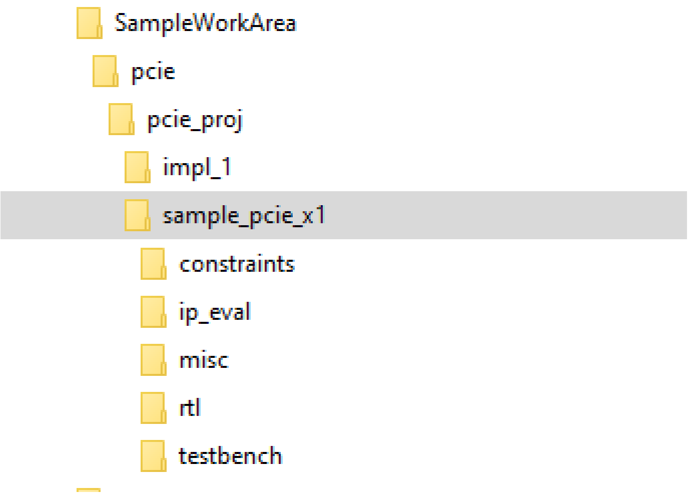
```
Component 'sample_pcie_x1' is successfully generated.
IP: pcie_x1   Version: 1.1.1
Vendor: latticesemi.com
Language: Verilog

Generated files:
IP-XACT_component: component.xml
IP-XACT_design: design.xml
black_box_verilog: rtl/sample_pcie_x1_bb.v
cfg: sample_pcie_x1.cfg
IP package file: sample_pcie_x1.ipx
template_verilog: misc/sample_pcie_x1_tmpl.v
dependency_file: testbench/dut_inst.v
dependency_file: testbench/dut_params.v
timing_constraints: constraints/sample_pcie_x1.ldc
template_vhdl: misc/sample_pcie_x1_tmpl.vhd
top_level_verilog: rtl/sample_pcie_x1.v
```

☑ Insert to project

Figure 3.6. Check Generated IP

The IP Core and other supporting files ARE generated in the specified directory. Figure 3.7 shows the directory structure of the generated IP Core.



```
SampleWorkArea
 pcie
  pcie_proj
   impl_1
   sample_pcie_x1
    constraints
    ip_eval
    misc
    rtl
    testbench
```

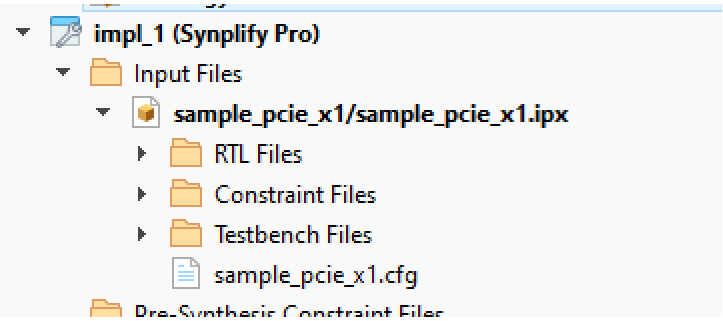
Figure 3.7. Generated IP Core Directory Structure

The testbench directory contains bus functional models for simulation. It also includes a generated file *dut_inst.v* that can be used to instantiate the IP Core either by including the file or by copying the module instance. A simple reference design for IP evaluation is provided in the *ip_eval* directory that is used when running synthesis flow and simulation.

3.5. Synthesizing and Implementing the IP Core

To run the synthesis flow using the IP evaluation reference design:

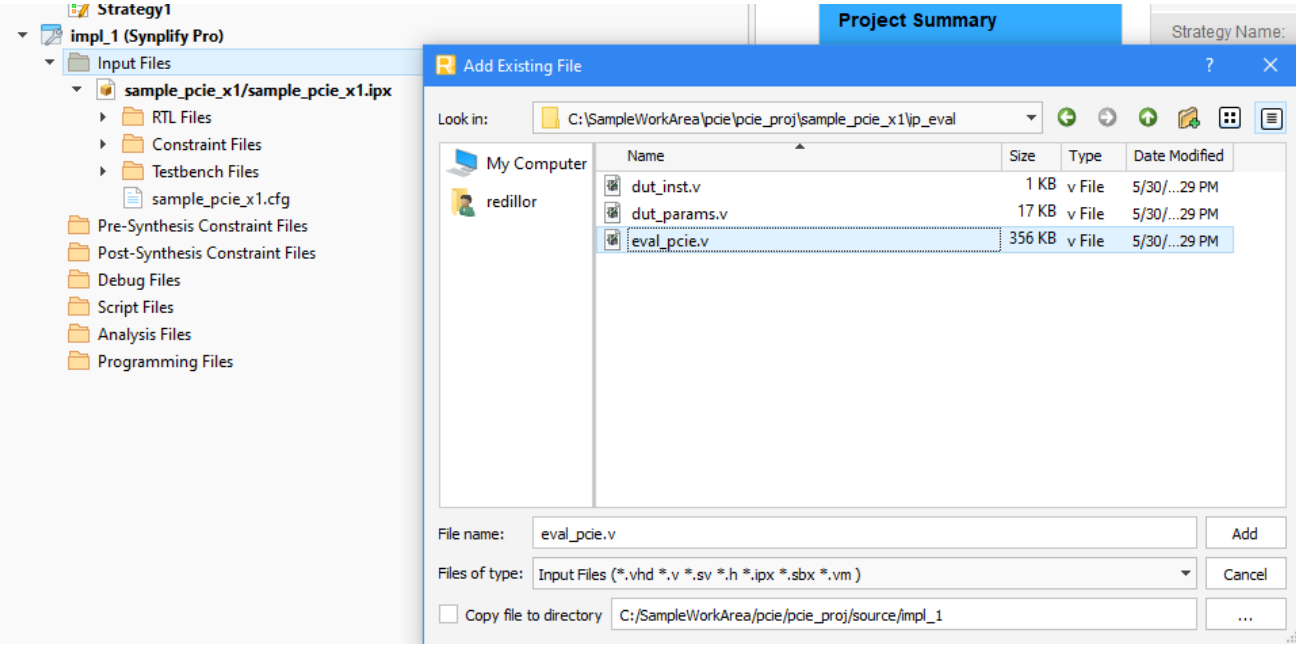
1. Check the **File List** tab. An ipx file should be in the **Input Files** after the IP Core is generated.



- impl_1 (Synplify Pro)
  - Input Files
    - sample_pcie_x1/sample_pcie_x1.ipx
      - RTL Files
      - Constraint Files
      - Testbench Files
      - sample_pcie_x1.cfg
  - Pre-Synthesis Constraint Files

Figure 3.8. Generated IP Core Directory Structure

2. Right click on the **Input Files**, select **Add** then select **Existing File**.
3. Browse through the generated IP Core directory and select **eval_pcie.v under ip_eval**.



- Strategy1
- impl_1 (Synplify Pro)
  - Input Files
    - sample_pcie_x1/sample_pcie_x1.ipx
      - RTL Files
      - Constraint Files
      - Testbench Files
      - sample_pcie_x1.cfg
  - Pre-Synthesis Constraint Files
  - Post-Synthesis Constraint Files
  - Debug Files
  - Script Files
  - Analysis Files
  - Programming Files

Project Summary

Strategy Name:

Add Existing File

Look in: C:\SampleWorkArea\pcie\pcie_proj\sample_pcie_x1\ip_eval

My Computer
redillor

Name	Size	Type	Date Modified
dut_inst.v	1 KB	v File	5/30/...29 PM
dut_params.v	17 KB	v File	5/30/...29 PM
eval_pcie.v	356 KB	v File	5/30/...29 PM

File name: eval_pcie.v

Files of type: Input Files (*.vhd *.v *.sv *.h *.ipx *.sbx *.vm)

☐ Copy file to directory C:/SampleWorkArea/pcie/pcie_proj/source/impl_1

Figure 3.9. Include eval_pcie.v top Design Module

4. Add the pdc constraint file eval_pcie.pdc (follow steps 2 and 3).
5. Run the synthesis flow, map and place and route by clicking the **Toolbar** button.

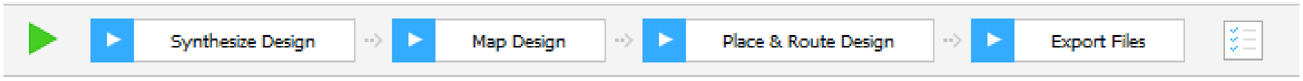


Figure 3.10. Run Synthesis Flow

The corresponding button turns green with a check mark once the flow is done.

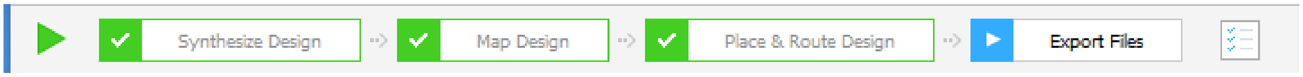


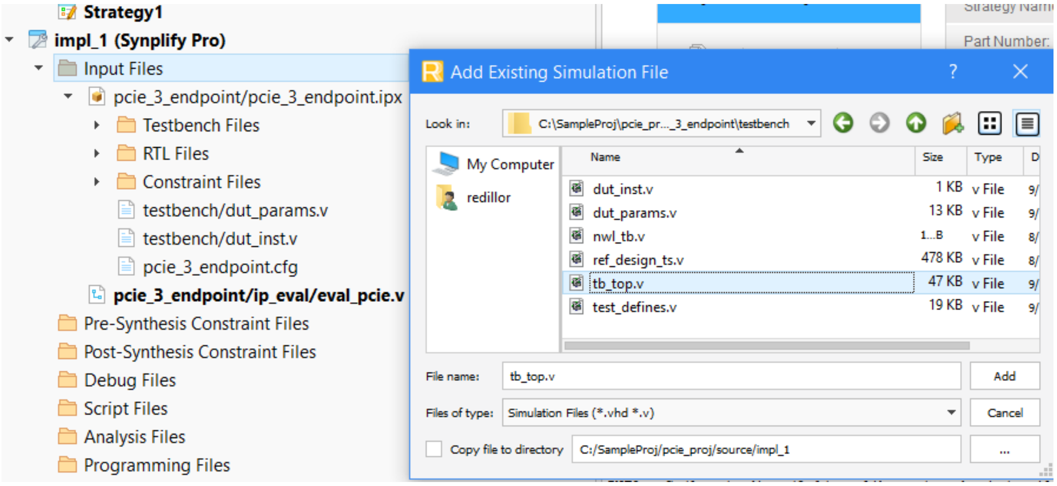
Figure 3.11. Synthesis Flow Status

3.6. Running Functional Simulation

Running functional simulation can be performed after the IP is generated.

To run functional simulation:

1. Right click on the Input Files, select **Add** then select **Existing Simulation File**.
2. Browse through the generated IP Core directory and select *tb_top.v* under testbench.



- Strategy1
- impl_1 (Synplify Pro)
  - Input Files
    - pcie_3_endpoint/pcie_3_endpoint.ipx
      - Testbench Files
      - RTL Files
      - Constraint Files
      - testbench/dut_params.v
      - testbench/dut_inst.v
      - pcie_3_endpoint.cfg
    - **pcie_3_endpoint/ip_eval/eval_pcie.v**
  - Pre-Synthesis Constraint Files
  - Post-Synthesis Constraint Files
  - Debug Files
  - Script Files
  - Analysis Files
  - Programming Files

Add Existing Simulation File

Look in: C:\SampleProj\pcie_pr..._3_endpoint\testbench

My Computer
redillor

Name	Size	Type	D
dut_inst.v	1 KB	v File	9/
dut_params.v	13 KB	v File	9/
nwl_tb.v	1...B	v File	8/
ref_design_ts.v	478 KB	v File	8/
tb_top.v	47 KB	v File	9/
test_defines.v	19 KB	v File	9/

File name: tb_top.v

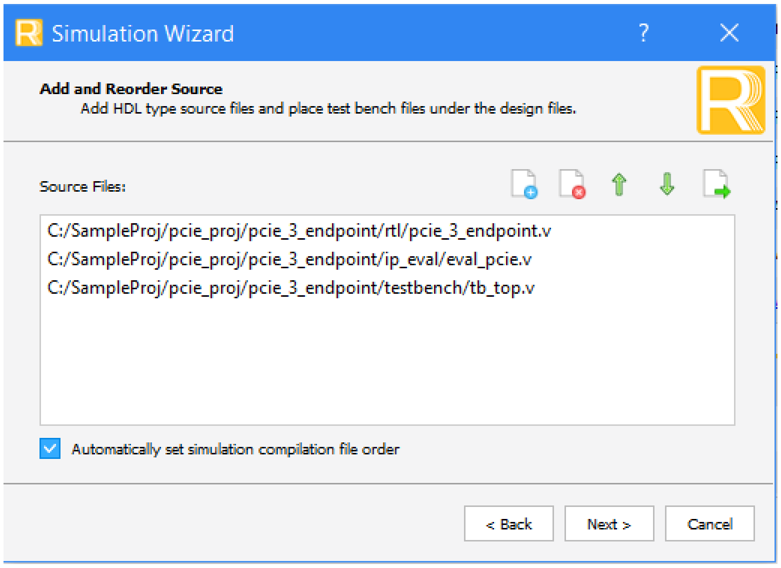
Files of type: Simulation Files (*.vhd *.v)

□ Copy file to directory C:/SampleProj/pcie_proj/source/impl_1

Figure 3.12. Include tb_top.v as Testbench File

3. Add *eval_pcie.v* (if not yet added)

3. Click the button located on the **Toolbar** to initiate the **Simulation Wizard**.
4. Click **Next** to open the **Add and Reorder Source** window.



Simulation Wizard

Add and Reorder Source

Add HDL type source files and place test bench files under the design files.

Source Files:

C:/SampleProj/pcie_proj/pcie_3_endpoint/rtl/pcie_3_endpoint.v
C:/SampleProj/pcie_proj/pcie_3_endpoint/ip_eval/eval_pcie.v
C:/SampleProj/pcie_proj/pcie_3_endpoint/testbench/tb_top.v

☑ Automatically set simulation compilation file order

Figure 3.13. Simulation Wizard

5. Click **Next**. The **Summary** window is shown.
6. Click **Finish** to run the simulation.

3.7. Hardware Evaluation

The Lattice PCIe X1 Core supports Lattice's IP hardware evaluation capability. This makes it possible to generate the IP core that operate in hardware for a limited period of time (approximately four hours) without requiring the purchase of an IP license. It may also be used to evaluate the core in hardware in user-defined designs. The hardware evaluation capability may be enabled/disabled in the Strategy dialog box. Hardware evaluation is enabled by default. To change this setting, go to Project > Active Strategy > LSE/Synplify Pro Settings.

4. Ordering Part Number

The Ordering Part Number (OPN) for this IP Core are the following:

- PCI-EXP1-CNX-U – PCI Express x1 Endpoint for CrossLink-NX - Single Design License
- PCI-EXP1-CNX-UT – PCI Express x1 Endpoint for CrossLink-NX - Site License
- PCI-EXP1-CTNX-U – PCI Express x1 Endpoint for Certus-NX - Single Design License
- PCI-EXP1-CTNX-UT – PCI Express x1 Endpoint for Certus-NX - Site License

Appendix A. Resource Utilization

Table 4.1 shows a sample resource utilization of the Lattice PCIe X1 Core on LIFCL-40-9BG400I.

Table 4.1. Resource Utilization

IP Configuration	Slices	LUTs	Registers	EBR	PCIE
AHB-Lite Data Interface, APB Register Interface, DMA Support Enabled	3172	5217	3172	29	1
Default Hard IP Interface	0	0	0	0	1

References

- For more info on the FPGA device, visit http://www.latticesemi.com/en/Products/FPGAandCPLD.
- For complete information on Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the Lattice Radiant Software User Guide.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.3, December 2020

Section	Change Summary
Introduction	Changed PCI Express Base Specification Revision to 3.0 in the Features section.
Functional Description	• Revised perst_n_i description in Table 2.8. Clock and Reset Port Descriptions. • Added information in the AHB-Lite Data Interface and the AHB-Lite Configuration Interface sections. • Adjusted level of pcie_ll (0x0F000) heading.
All	Updated reference to Lattice Radiant Software User Guide.

Revision 1.2, June 2020

Section	Change Summary
All	• Changed IP name to PCIE X1. • Changed document title to PCIe X1 IP Core - Lattice Radiant Software.
Disclaimers	Added this section.
Introduction	Updated Table 1.1: • Added Certus-NX support. • Added LFD2NX-40 as targeted device. • Updated Synopsis Synplify Pro version. • Updated Lattice Implementation to Lattice Radiant 2.1.
Functional Description	• Added Root Port mode support. • Updated diagrams to match the current IP. • Corrected IP Register names and offset mapping.
IP Generation	Removed reference to Lattice Radiant 2.0 Tutorial.
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	Updated device to LIFCL-40-9BG400I.

Revision 1.1, March 2020

Section	Change Summary
All	• Removed details for unsupported features. • Minor adjustments in formatting/styles.

Revision 1.0, December 2019

Section	Change Summary
All	Initial release.

LATTICE
SEMICONDUCTOR®

www.latticesemi.com