25 PCI Express Core

25.1 Overview

The PCI Express Core (PeCORE) is a fast serial, dual simplex interconnect core with a maximum bandwidth of 40Gbits/s on each direction (transmit and receive). PeCORE is interfaced to the Talino™ EDC subsystem through buses AWB and LCB with ICBM as its DMA module.

25.1.1 References

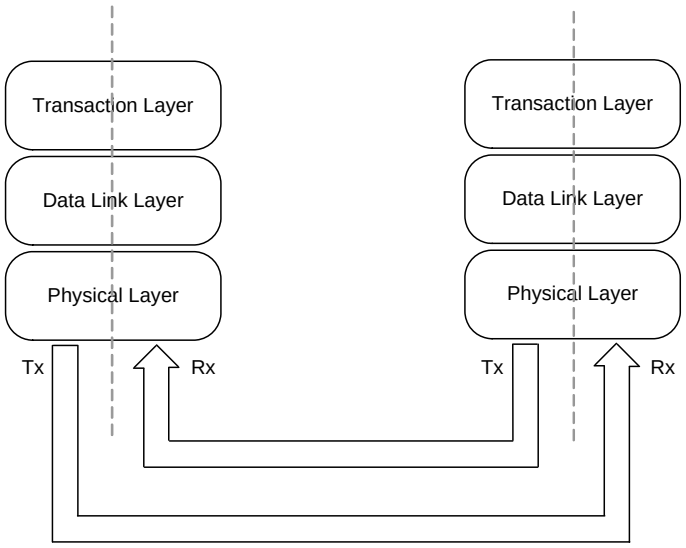
- PCI Express™ Base Specification Revision 1.1, March 28, 2005
- PCI Express™ Base Specification Revision 2.0, December 20, 2006
- PCI Express™ Base Specification Revision 2.1, March 4, 2009
- PHY Interface for the PCI Express™ Architecture Version 2.00

25.1.2 PCI Express Fundamentals

25.1.2.1 PCI Express Layer Architecture

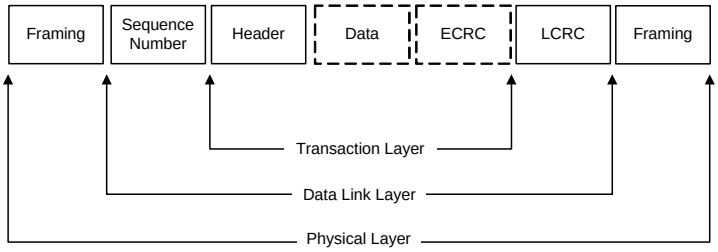
The PCI Express Specification identifies three discrete logical layers: the Transaction Layer, the Data Link Layer, and the Physical Layer. Each of these layers is divided into two sections: one that processes outbound (to be transmitted) information and one that process inbound (received) information, as shown in ***Figure 25.1***

Figure 25.1: High Level Layering Diagram



PCI Express uses packets to communicate information between components. Packets are formed in the Transaction and Data Link Layers to carry the information from the transmitting component to the receiving component. As the transmitted packets flow through the other layers, they are extended with additional information necessary to handle packets at those layers. At the receiving component the reverse process occurs. Packets get transformed from their Physical Layer representation to the Data Link Layer representation and finally to the form (TLP format) that can be processed by the Transaction Layer of the receiving device. ***Figure 25.2*** shows the conceptual flow of transaction level packet information through the layers.

Figure 25.2: Packet Flow Through the Layers



Transaction Layer (TL)

The Transaction Layer's primary responsibility is the assembly and disassembly of Transaction Layer Packets (TLPs). TLPs are used to communicate transactions, such as read and write. The TL is also responsible for managing credit-based flow control for TLPs, storage of configuration information, updates status information, ordering rules and power management services.

Data Link Layer (DLL)

The Data Link Layer primary responsibilities include Link management and data integrity, including error detection and correction. The transmission side of the DLL accepts TLPs assembled by the Transaction Layer, calculates and applies a data protection code and TLP sequence number, and submits them to the Physical Layer for transmission across the Link. The receiving DLL is responsible for checking the integrity of received TLPs and for submitting them to the TL for further processing. On detection of TLP errors, this layer is responsible for requesting retransmission of TLPs until information is correctly received, or the Link is determined to have failed. It is also responsible for power management, error reporting, and logging.

Physical Layer (PL)

The Physical Layer is responsible for converting information received from the DLL into an appropriate serialized format and transmitting it across the Link. The PL includes all circuitry for interface operation, including driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. This layer is also responsible for power management, width and lane negotiation, and reset/hot-plug control.

25.1.2.2 Transaction Layer Packet

Transaction Layer Packet (TLP) is used to send high level information between two PCI Express devices. It is through TLP that a Requester sends a Request transaction to a target device, and a Completer sends a Completion transaction to the initiator. Completion Requests are used only where required, for example, to return read data or to acknowledge Completion of IO and Configuration Write Transactions. Completion transactions are associated with their corresponding Requests by the value in the Transaction ID field of the TLP Header.

Figure 25.3: Generic TLP Format

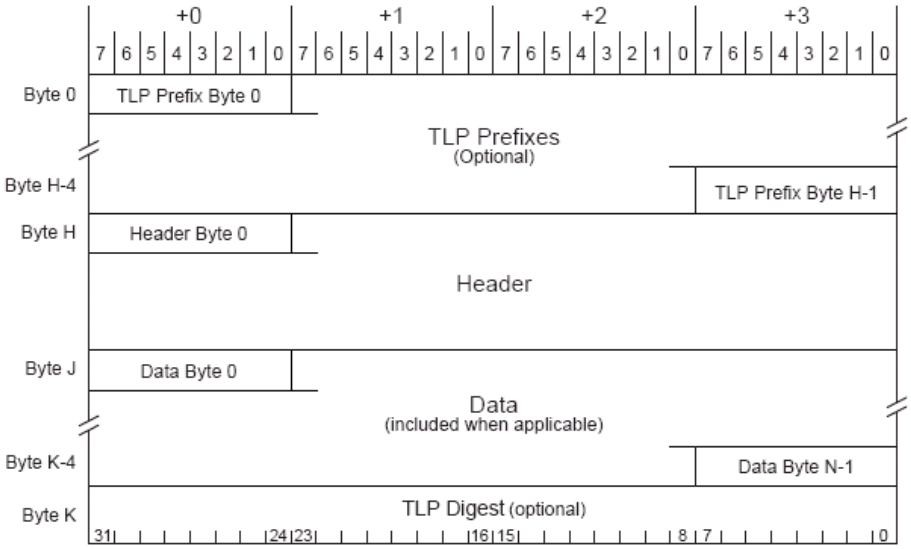


Figure 25.3 shows the generic TLP format, which is composed of the optional TLP Prefixes, the Header, the Data (for some types of TLP), and an optional TLP Digest. The TLP Header and TLP Digest are drawn with the lower numbered bytes on the left rather than on the right as has traditionally been depicted in other PCI specifications.

As a convention throughout this chapter, the header of a TLP can be termed as either TLP Header or Header, the same with TLP Payload or Payload used to describe the payload in a TLP. Other kinds of header or payload must be named explicitly.

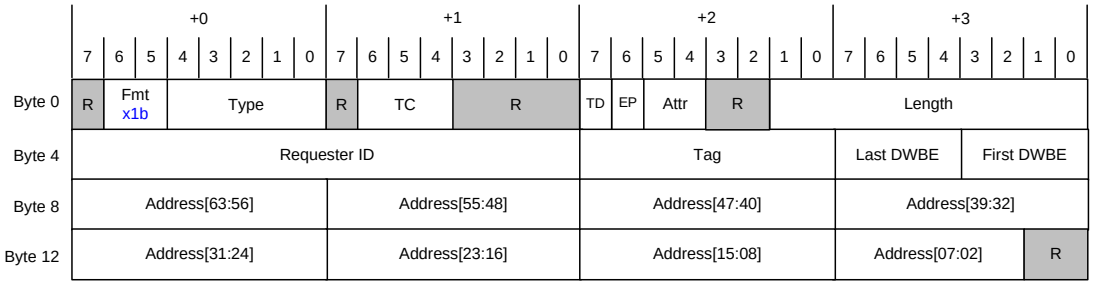
25.1.2.2.1 TLP Header Format

25.1.2.2.1.1 Memory Request, 64-bit Addressing

Figure 25.4 shows the TLP Header for a Memory Request with 64-bit address. This header format applies to any of the following Transactions:

- Memory Read Request, 64-bit Addressing
- Memory Write Requests, 64-bit Addressing

Figure 25.4: Memory Request, 64-bit addressing TLP Header

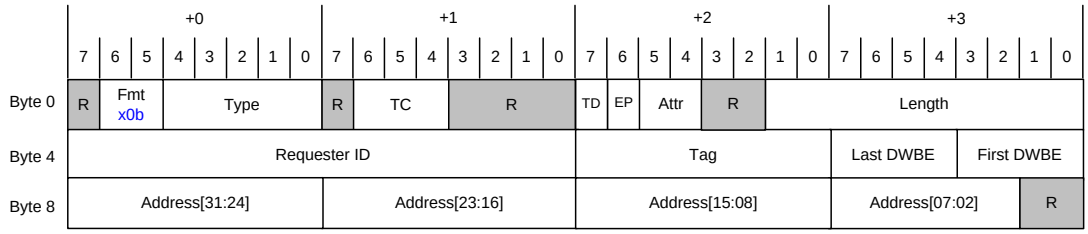


25.1.2.2.1.2 Memory Request, 32-bit Addressing

Figure 25.5 shows the TLP Header for a Memory Request with 32-bit address. This header format applies to any of the following Transactions:

- Memory Read Request, 32-bit Addressing
- Memory Write Requests, 32-bit Addressing

Figure 25.5: Memory Request, 32-bit addressing TLP Header

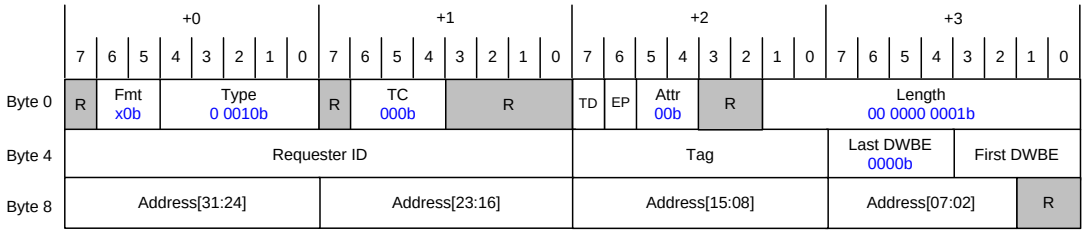


25.1.2.2.1.3 IO Request

Figure 25.6 shows the TLP Header for a IO Request. This header format applies to any of the following Transactions:

- IO Read Request
- IO Write Request

Figure 25.6: IO Request TLP Header

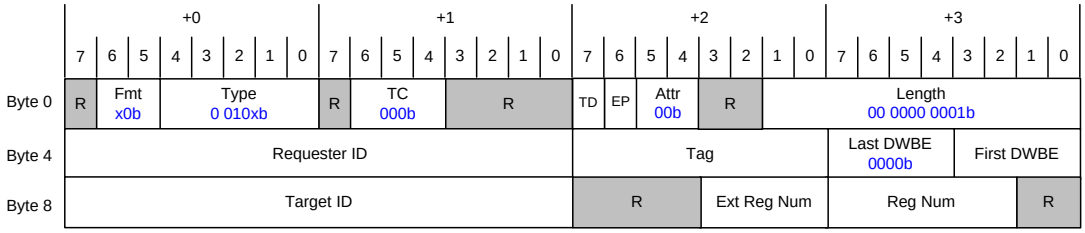


25.1.2.2.1.4 Configuration Request

Figure 25.7 shows the TLP Header for a Configuration Request. This header format applies to any of the following Transactions:

- Configuration Read Request Type 0
- Configuration Read Request Type 1
- Configuration Write Request Type 0
- Configuration Write Request Type 1

Figure 25.7: Configuration Request TLP Header

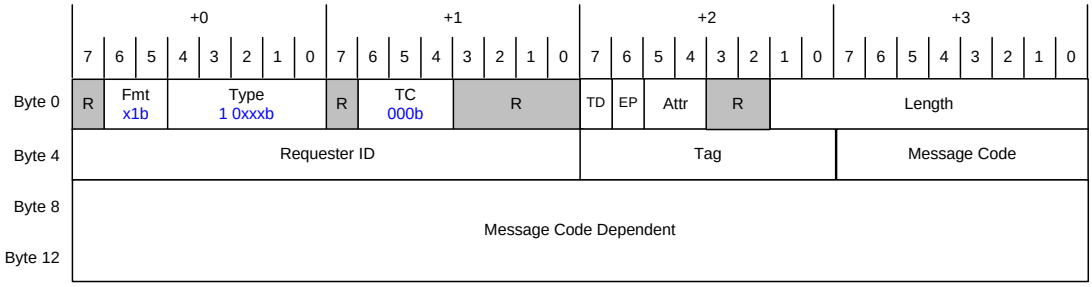


Configuration Request

25.1.2.2.1.5 Message Request

Figure 25.7 shows the TLP Header for a Message Request. This header format applies to all types of Message Request.

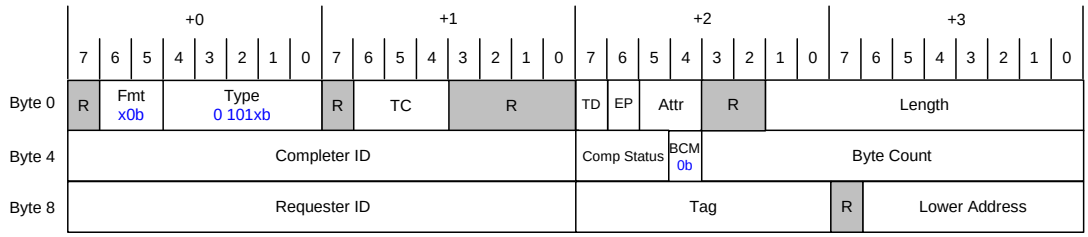
Figure 25.8: Message Request TLP Header



25.1.2.2.1.6 Completion

Figure 25.9 shows the TLP Header for all kinds of Completion.

Figure 25.9: Completion TLP Header



25.1.2.2.2 TLP Payload

Supported TLP Payload size varies depending on the configuration of PeCORE. PeDRF_PEDSCTL register (***Section 25.3.1.6.3***) defines the maximum TLP Payload size for the device/function.

25.2 Theory of Operation

25.2.1 Functional Architecture

The PCI Express Core (PeCORE) together with ICBM3 module is capable of the following PCI Express functions: (1) Root Complex, or (2) PCI Express Endpoint - depending on the application and topology where the Talino™ EDC will be used. The sections that follow illustrate the types of functional architecture PeCORE supports.

25.2.1.1 Root Complex

The Root Complex serves as the root of an I/O hierarchy that connects the CPU and Memory subsystem to the I/O. The EDC completes a Root Complex fabric with its processors, internal and external memory. PeCORE then functions as a one-port Root Complex, and its PCI Express Link can have 1, 2, 4, or 8 lanes for a maximum bandwidth of 40Gb/s (5Gb/s per lane).

Figure 25.10: Root Complex Configuration in basic topology.
The EDC is connected to a single Endpoint device, PeCORE forming a simple PCI Express fabric.

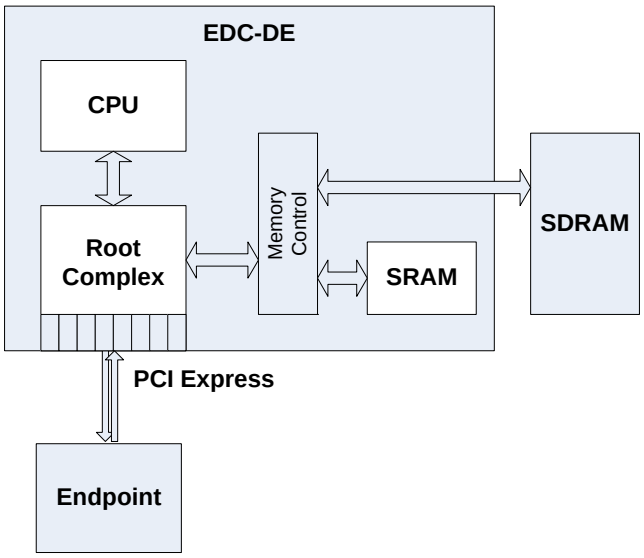
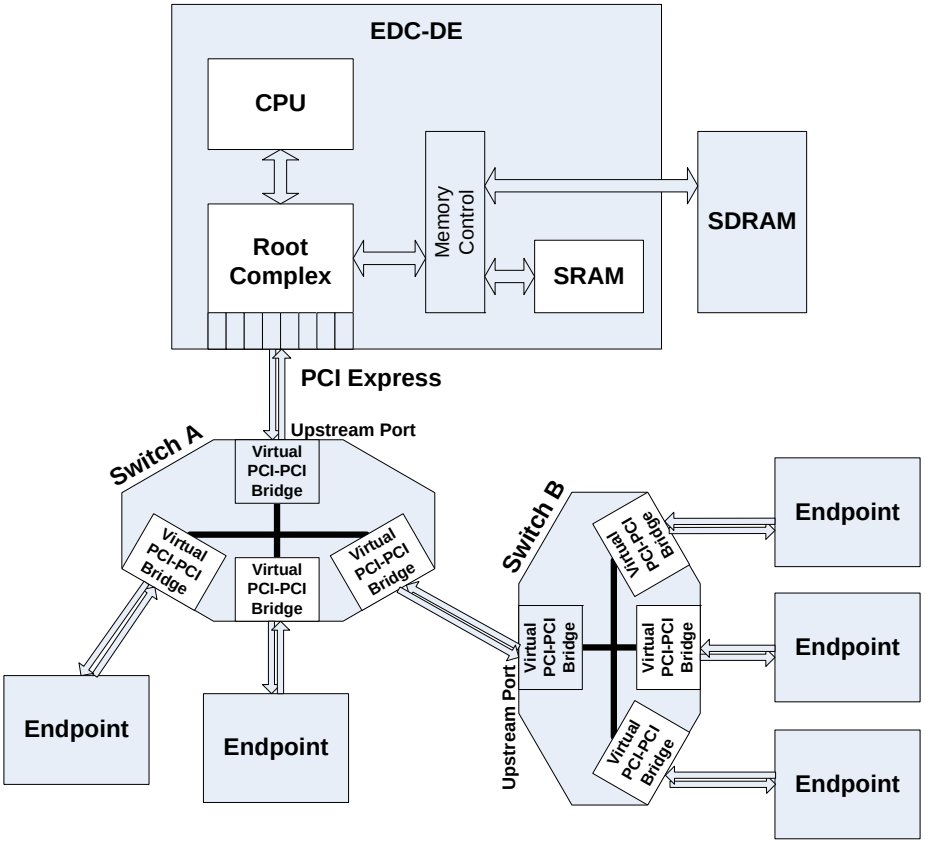


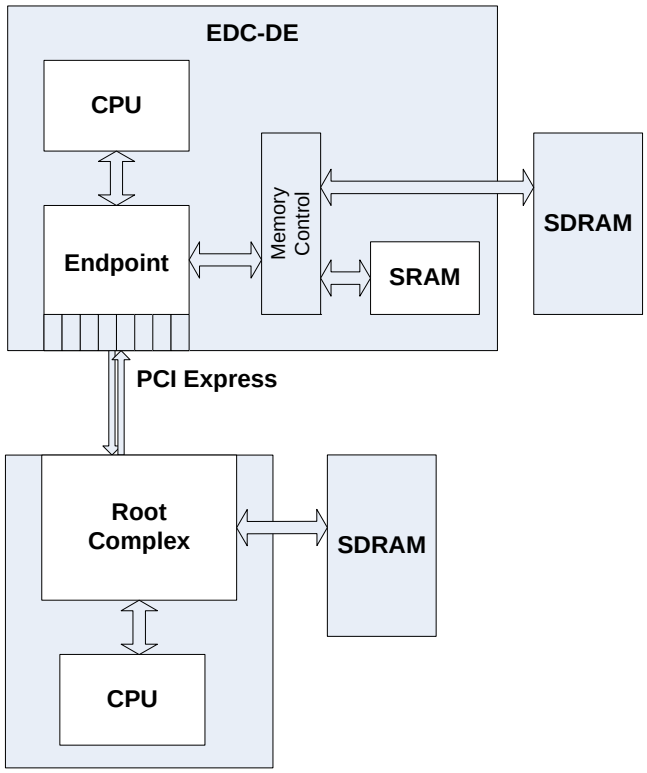
Figure 25.11: Root Complex with complex hierarchy. The EDC is connected to a Switch device, PeCORE can communicate with several Endpoints within the PCI Express fabric.



25.2.1.2 PCI Express Endpoint

The Endpoint is a device that can function only as the requester (for memory and message transactions only) or completer of a PCI Express transaction. PeCORE interfaces the EDC resources to the Root Complex fabric basically for memory transactions. Its PCI Express Link can have upto 8 lanes for a maximum bandwidth of 40Gb/s (5Gb/s per lane) in each direction.

Figure 25.12: PCI Express Endpoint in basic topology. The EDC is connected to a Root Complex device, and acts only as the requester (memory and message transactions only) or completer.



25.2.2 Modes of Operation

PeCORE supports the following modes of operation:

- PCI Express Endpoint Mode
- Root Complex Mode

PeCORE can be viewed as a bridge whether it's in PCI Express Endpoint mode or in Root Complex mode. Its bridge function provides an interface between PCIe bus and Talino™ EDC internal buses (AWB0-3). It enables PCIe requesters to access Talino™ EDC internal buses' slaves (and PeCORE's registers). It also enables Talino™ EDC internal bus masters to access PCIe targets.

25.2.2.1 PCI Express Endpoint Mode

Below is the list of the supported transactions in this mode.

Table 25.1: PCI Express Endpoint Mode Transactions

Transaction	Requester	Completer
Memory Read/Write	Yes	Yes
IO Read/Write	No	Yes
Cfg Read/Write	No	Yes
Msg	Yes	Yes

25.2.2.2 Root Complex Mode

This mode supports all transactions except for the reception of IO and Cfg Requests. Below is the list of the supported transactions.

Table 25.2: Root Complex Mode Transactions

Transaction	Requester	Completer
Memory Read/Write	Yes	Yes
IO Read/Write	Yes	No
Cfg Read/Write	Yes	No
Msg	Yes	Yes

25.2.3 Process Flows

25.2.3.1 Initialization

25.2.3.1.1 Boot from OTPROM as Root Complex

After system reset, PeCORE is initialized through the following sequence of events. Refer to “Initialization / Power-On Reset Sequence” section of CrCORE for a more detailed Power-On Reset sequence.

1. FpCORE detects the presence of bootstrap code loader from OTPROM.
2. Bootstrap loader copies Strap Configuration Code from OTPROM.
3. Bootstrap loader copies bootstrap code from OTPROM to Talino™ EDC Shared Memory.
4. Bootstrap loader jumps to entry point of Bootstrap code in Talino™ EDC Shared Memory
5. Bootstrap code configures PLLs

   After the configuration of PLLs, CrCORE enables PeCORE core clock
6. Bootstrap code copies System FW from Flash or SFPROM to Talino™ EDC Shared Memory.
7. Bootstrap code jumps to entry point of System FW.
8. System FW further initializes PeCORE.

   PeCORE registers (PCIe Configuration Registers and Internal Registers) are further initialized here.

25.2.3.1.2 Boot from OTPROM as Endpoint

After system reset, PeCORE is initialized through the following sequence of events. Refer to “Initialization / Power-On Reset Sequence” section of CrCORE for a more detailed Power-On Reset sequence.

1. FpCORE detects the presence of bootstrap code loader from OTPROM.
2. Bootstrap loader copies Strap Configuration Code from OTPROM.

   While Talino™ EDC is not yet done with its initialization, the Host should be prevented from running PCI Configuration Sequence. This is done by having the value of HCE (Host Configuration Enable of PeDRF_COPT) set to 0. If HCE is 0, PeCORE returns a CplStat of CRS (Configuration Request Retry Status) to Configuration Request from the Host. This means that the Host Configuration Request will have to be retried.
3. Bootstrap loader copies bootstrap code from OTPROM to Talino™ EDC Shared Memory.
4. Bootstrap loader jumps to entry point of Bootstrap code in Talino™ EDC Shared Memory
5. Bootstrap code configures PLLs

   After the configuration of PLLs, CrCORE enables PeCORE core clock
6. Bootstrap code copies System FW from Flash or SFPROM to Talino™ EDC Shared Memory.
7. Bootstrap code jumps to entry point of System FW.
8. System FW further initializes PeCORE.

   PeCORE registers (PCIe Configuration Registers and Internal Registers) are further initialized here.
9. Talino™ EDC allows Host to run PCI Configuration Sequence.

   System FW sets HCE of PeDRF_COPT to 1. This enables the response of Talino™ EDC to Host Configuration Requests.

25.2.3.1.3 Strap signals

Table 25.3 lists the strap signals of PeCORE.

Table 25.3: PeCORE Strap Signals

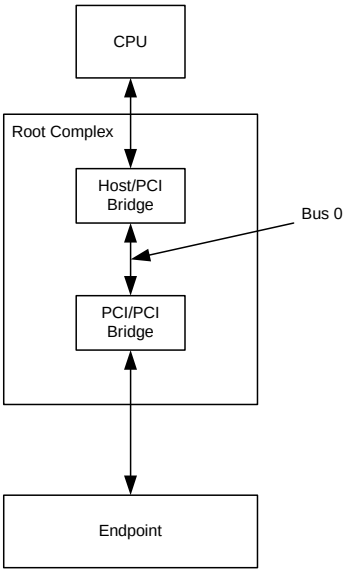
Strap	
PBM	0 - PeCORE in PCI Express Endpoint mode 1 - PeCORE in Root Complex mode
LINKWIDTH	00 - LinkWidth = 1 lane 01 - LinkWidth = 2 lanes 10 - LinkWidth = 4 lanes 11 - LinkWidth = 8 lanes
LINKSPEED	0 - Gen 1 speed (2.5Gbps) 1 - Gen 2 speed (5.0Gbps)
FC	000 - 50% (PD), 50% (CPLD), 50% (PH), 50%* (CPLH) 001 - 75% (PD), 25% (CPLD), 50% (PH), 50%* (CPLH) 010 - 25% (PD), 75% (CPLD), 50% (PH), 50%* (CPLH) 011 - 90% (PD), 10% (CPLD), 50% (PH), 50%* (CPLH) 100 - 10% (PD), 90% (CPLD), 50% (PH), 50%* (CPLH) 101 - 50% (PD), 50% (CPLD), 50% (PH), 50% (CPLH) 110 - 75% (PD), 25% (CPLD), 75% (PH), 25% (CPLH) 111 - 25% (PD), 75% (CPLD), 25% (PH), 75% (CPLH) *actual is 50%, but infinite is advertised during FC Init

25.2.3.1.4 Sample PCI Configuration Sequence

In this example, Talino™ EDC is a single-function PCIe endpoint connected to a single-port Root Complex. *Bus 0* can be assumed as an internal bus inside the Root Complex or the bus between the lone Root Port and the endpoint. For single-port Root Complex, it is advisable to use the latter assumption.

25.2.3.1.4.1 Bus 0 is internal to Root Complex

Figure 25.13: Bus 0 internal to Root Complex



Bus Enumeration

1. Root Complex CPU-reads VendorID of bus0/device0/function0. This is the Host VendorID.

   VendorID = valid

2. Root Complex CPU-reads “Header Type”.

   Header Type = 0000001b

   Multifunction bit = 0b

3. Root Complex CPU-configures “Bus Numbers”.

   Primary Bus Number = 0

   Secondary Bus Number = 1

   Subordinate Bus Number = 1

4. Root Complex CPU-reads “Capability Register”.
   Device/Port Type = 0100b (root port)

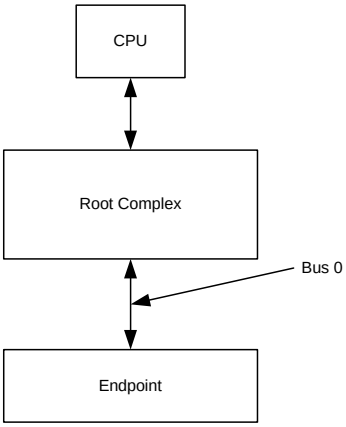
5. Root Complex reads (CfgRd0) VendorID of bus1/device0/function0.
   Vendor ID = 192Ah
6. Root Complex reads (CfgRd0) “Header Type”
   Header Type = 0000000b
   Multifunction bit = 0b
7. Root Complex reads (CfgRd0) “Capability Register”
   Device/Port Type = 0000b (endpoint)

BAR Configuration

8. Root Complex reads (CfgRd0) BAR0 of bus1/device0/function0. Saves it.
9. Root Complex writes (CfgWr0) FFFFFFFFh to BAR0.
10. Root Complex reads (CfgRd0) BAR0 again.
11. Clear bits 0-3 (Memory) or 0 (IO) of the newly read value of BAR0.
12. Invert all 32 bits then add 1. The result is the requested memory/IO size.
13. Root Complex assigns (CfgWr0) address to BAR0.
14. Repeat 8-13 until all BARs are configured.

25.2.3.1.4.2 Bus 0 is the Root Port bus

Figure 25.14: Bus 0 is the Root Port bus



Bus Enumeration

1. Root Complex reads (CfgRd0) VendorID of bus0/device0/function0.
   Vendor ID = 192Ah
2. Root Complex reads (CfgRd0) “Header Type”
   Header Type = 0000000b
   Multifunction bit = 0b
3. Root Complex reads (CfgRd0) “Capability Register”
   Device/Port Type = 0000b (endpoint)

BAR Configuration

4. Root Complex reads (CfgRd0) BAR0 of bus0/device0/function0. Saves it.
5. Root Complex writes (CfgWr0) FFFFFFFFh to BAR0.
6. Root Complex reads (CfgRd0) BAR0 again.
7. Clear bits 0-3 (Memory) or 0 (IO) of the newly read value of BAR0.
8. Invert all 32 bits then add 1. The result is the requested memory/IO size.
9. Root Complex assigns (CfgWr0) address to BAR0.
10. Repeat 4-9 until all BARs are configured.

25.2.3.1.5 PeCORE Register Initialization

This section provides the minimum register initialization PeCORE needs to operate in a particular mode of operation.

1. To generate Cfg request

FW can generate PCI-Compatible Cfg request without any prior register initialization, but to generate PCIe Enhanced Cfg request, FW must first set Configuration Request Base Address register (PeDRF_CRBAR) to the desired location. Address allocated to this register must be within the range of any of the 8 PCI Memory spaces. See Table 25.4.

Note that Cfg request can be generated only when PeCORE is in RootComplex mode.

Table 25.4: PCI Memory/IO Spaces

PCI Memory 0 - 3	**0xC000_0000 - 0xCFFF_FFFF**
PCI Memory 4 - 7	0xD000_0000 - 0xDFFF_FFFF
PCI IO 0 - 1	0xF000_0000 - 0xF7FF_FFFF

2. To generate IO request

FW must set up IO Base/Limit registers (PeDRF_HTSIOLB, PeDRF_HTIOLBU) and POM PCI register (PeDRF_POMP*n*). IO Base/Limit registers and POM PCI register must be within the range of any of the 2 PCI IO spaces. Refer to Table 25.4 for the available PCI IO spaces.

Bus Master Enable (PeDRF_PCSTCMD[2]) must also be set to enable generation of IO request.

Note that IO request can be generated only when PeCORE is in RootComplex mode.

3. To generate Memory request

For direct mode, when in RC mode, FW must set up Memory Base/Limit registers (PeDRF_HTMLMBA, PeDRF_HTPMLMB) and POM PCI register (PeDRF_POMP*n*). Memory Base/Limit registers and POM PCI register must be within the range of any of the 8 Talino™ EDC PCI Memory spaces.

For direct mode, when in EP mode, FW must set up the POM registers (PeDRF_POMP*n*, PeDRF_POML*n*, PeDRF_POMS*n*). POM address registers must be within the range of any of the 8 Talino™ EDC PCI Memory spaces. PeDRF_POMS*n*[0] must be set to enable TLP generation.

PeDRF_POMS*n*[1] must be set both in EP and RC mode to enable address translation. Refer to Table 25.15 and Table 25.16 for the address translation using the mentioned address registers above.

For any other mode of generation of memory request, no need to set up the registers above. Bus Master Enable (PeDRF_PCSTCMD[2]), however, must be enabled for all modes of memory request generation.

To use CSR-DMA Channel 0 in indirect mode 1 (or simply called indirect mode), PeDRF_MRCR0[1] must be set to 0. To enable interrupt in this mode, PeDRF_MRCR0[21] and PeDRF_CINT[17] must be set to 1. Note that only CSR-DMA Channel 0 can be used in indirect mode.

To use CSR-DMA Channels 0 - 3 in indirect mode 1 (or simply called mode1), PeDRF_MRCR*n*[1] must be set to 1. To enable interrupt in this mode, PeDRF_MRCR*n*[21] and PeDRF_CINT[17] must be set to 1.

For ICBM descriptor-based DMA, refer to ICBM section of this document for the required register initialization.

4. To receive Cfg request

FW must set Host Configuration Enable (HCE) bit (PeDRF_COPT[1]) to indicate that Talino™ EDC is ready to response to Host Configuration Request. This is normally set when Talino™ EDC is done with its self-initialization. While HCE is 0, PeCORE replies with Completion with CRS status for every Configuration request it receives.

5. To receive IO request

IO Slave Enable (PeDRF_PCSTCMD[0]) must be set.

6. To receive Memory request

Memory Slave Enable (PeDRF_PCSTCMD[1]) must be set.

25.2.3.2 Generation of Configuration Request

Configuration Request can be generated in two ways:

PCI-Compatible Mechanism - Configuration Request is generated through the use of DATA and ADDRESS registers.

PCIe Enhanced Configuration Mechanism - Configuration Request is generated by directly issuing a read/write access to Talino™ EDC PCI Memory space assigned for PCI Configuration Space.

25.2.3.2.1 PCI-Compatible Mechanism

25.2.3.2.1.1 Configuration Write

1. Write destination ID to Host Configuration Address Register (PeDRF_CRAR) with configuration enabled (CFGEN=1).

Table 25.5: PeDRF_CRAR

31	30:24	23:16	15:11	10:8	7:2	1	0
Enable	Reserved	Bus Number	Device Number	Function Number	Register Number	0	0

2. If needed, set up FBE in PeDRF_CRCR[7:4]. Default value of PeDRF_CRCR[7:4] is 4'b1111.
3. Write data to Host Configuration Data Register (PeDRF_CRDR).

   Write access to PeDRF_CRDR triggers generation and transmission of CfgWr TLP. PCI-Compatible CfgWr Request can be viewed as a normal CPU write. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending CfgWr request is received.

   If the Completion status is a CRS (Configuration Request Retry Status), the CfgWr request will be automatically retried by the hardware.

   To take advantage of CRS Software Visibility, it is recommended that the first Cfg request is a CfgRd0 to PeDRF_PCDVEID (0xFFF2_1000). To indicate that the hardware has received a CRS completion status, if CRS Software Visibility is enabled, PeCORE will return a 0xFFFF_0001 value to the CPU. The CPU, then, can attend to some other tasks instead of waiting for a successful completion. If CRS Software Visibility is not supported or disabled or the Cfg request is not a CfgRd0 to PeDRF_PCDVEID, the request will be automatically retried by the hardware.

   UR and CA completion status will be logged as errors to error registers.

25.2.3.2.1.2 Configuration Read

1. Write destination ID to Host Configuration Address Register (PeDRF_CRAR) with configuration enabled (CFGEN=1).
2. If necessary, set up FBE in PeDRF_CRCR[7:4]. Default value of PeDRF_CRCR[7:4] is 4'b1111.
3. Read Host Configuration Data Register (PeDRF_CRDR).

   Read access to PeDRF_CRDR triggers generation and transmission of CfgRd TLP. PCI-Compatible CfgRd Request can be viewed as a normal CPU read. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending CfgRd request is received.

   If the Completion status is a CRS (Configuration Request Retry Status), the CfgRd request will be automatically retried by the hardware. Refer to Section 25.2.3.2.1.1 Configuration Write regarding CRS Software Visibility.

   UR and CA Completion status are logged as errors to the corresponding error registers.

25.2.3.2.2 PCIe Enhanced Configuration Mechanism

25.2.3.2.2.1 Configuration Write

1. Set up the address as in Table 25.6, where 'n' is the Bus Number Field Width (PeDRF_COPT[7:5]).

   Sample address mapping:

   - n=8
   - A[31:28] = PeDRF_CRBAR[31:28]
   - A[27:20] = Bus Number

Table 25.6: Enhanced Configuration Addressing Mapping

Memory Address	PCIe Configuration Space
A[(20+n-1):20]	Bus Number (1 <= n <= 8)
A[19:15]	Device Number
A[14:12]	Function Number
A[11:8]	Extended Register Number
A[7:2]	Register Number
A[1:0]	Along with size of the access, used to generate Byte Enables

2. If needed, set up FBE in PeDRF_CRCR[7:4]. Default value of PeDRF_CRCR[7:4] is 4'b1111.
3. CPU-write to above address.
4. PeCORE recognizes the write access above as an access to PCIe Configuration

Space, that is, it falls within the range of one of the 8 Talino™ EDC PCI Memories and that the upper 4 bits of the address is the same as PeDRF_CRBAR[31:28]. This triggers generation and transmission of CfgWr TLP.

PCIe Enhanced CfgWr Request can be viewed as a normal CPU memory write. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending CfgWr request is received.

If the Completion status is a CRS (Configuration Request Retry Status), the CfgWr request will be automatically retried by the hardware. Refer to Section 25.2.3.2.1.1 Configuration Write regarding CRS Software Visibility.

UR and CA Completion status are logged as errors to the corresponding error registers.

25.2.3.2.2.2 Configuration Read

1. Set up the address as in Table 25.6, where ‘n’ is the Bus Number Field Width (PeDRF_COPT[7:5]).

   Sample address mapping:

   - n=8
   - A[31:28] = PeDRF_CRBAR[31:28]
   - A[27:20] = Bus Number
2. If necessary, set up FBE in PeDRF_CRCR[7:4]. Default value of CRCR[7:4] is 4’b1111.
3. CPU-read to above address.
4. PeCORE recognizes the read access above as an access to PCIe Configuration Space, that is, it falls within the range of one of the 8 Talino™ EDC PCI Memories and that the upper 4 bits of the address is the same as PeDRF_CRBAR[31:28]. This triggers generation and transmission of CfgRd TLP.

   PCIe Enhanced CfgRd Request can be viewed as a normal CPU memory read. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending CfgRd request is received.

   If the Completion status is a CRS (Configuration Request Retry Status), the CfgRd request will be automatically retried by the hardware. Refer to Section 25.2.3.2.1.1 Configuration Write regarding CRS Software Visibility.

   UR and CA Completion status are logged as errors to the corresponding error registers.

25.2.3.3 Generation of IO Request

25.2.3.3.1 IO Write

1. If needed, set up FBE in PeDRF_CRCR[7:4]. Default value of PeDRF_CRCR[7:4] is 4'b1111.
2. CPU-write to target address. This should be a direct access to Talino™ EDC PCI IO Spaces.
3. PeCORE recognizes the write access above as an access to PCI IO Space. If POM is enabled (PeDRF_POMS*n*[0] = 1) and address falls within POM Local range, PeCORE initiates the generation and transmission of IOWr TLP. If outbound mapping is enabled (PeDRF_POMS*n*[1] = 1), original address is mapped to the equivalent POM PCI address prior to IOWr TLP transmission.

   IOWr Request can be viewed as a normal CPU memory write. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending IOWr request is received.

   UR and CA Completion status are logged as errors to the corresponding error registers.

25.2.3.3.2 IO Read

1. If needed, set up FBE in PeDRF_CRCR[7:4]. Default value of PeDRF_CRCR[7:4] is 4'b1111.
2. CPU-read the target address. This should be a direct access to Talino™ EDC PCI IO Spaces.
3. PeCORE recognizes the above read access as an access to PCI IO Space. If address falls within the range of IO Base/Limit registers, PeCORE initiates the generation and transmission of IORd TLP. If outbound mapping is enabled (PeDRF_POMS*n*[1] = 1), original address is mapped to the equivalent POM PCI address prior to IORd TLP transmission.

   IORd Request can be viewed as a normal CPU memory read. No need to wait for interrupt or poll the completion status. Next instruction will be executed only after the Completion of the pending IORd request is received.

   UR and CA Completion status are logged as errors to the corresponding error registers.

25.2.3.4 Generation of Memory Request

Generation of Memory Request can be done in 4 ways.

Direct Mode - this is done by directly issuing a memory read/write access to Talino™ EDC PCI Memory space. Length is always 1 dword in this mode.

Indirect Mode - this is done by using DATA (PeDRF_MRDR0-3) and ADDRESS (PeDRF_MRALR and PeDRF_MRAHR) registers. Length can be upto 4 dwords in this mode.

CSR-based descriptor DMA - this mode is used in transferring large bulk of data. Unlike Indirect Mode where the source (if MWr) or destination (if MRd) of data is the data registers, the source (if MWr) or destination (if MRd) of data in this mode is the Talino™ EDC Shared Memory. The DMA descriptors is programmed using CSRs. Up to 4 DMA requests can be issued in parallel. DMA Chaining option is also available. This mode is also called *mode1*.

ICBM descriptor-based DMA - this mode is used in transferring large bulk of data. The source (if MWr) or destination (if MRd) of data in this mode is also the Talino™ EDC Shared Memory. The DMA descriptors is programmed in the memory as link-list descriptors. Once the DMA (ICBM) is initiated, the descriptors are automatically fetched and executed.

25.2.3.4.1 Memory Write, Direct Mode

1. if 64-bit addressing mode is enabled (PeDRF_MRCR0[0]=1), write the upper 32-bit destination address to PeDRF_MRAHR0.

   Payload Length of Memory Write in this mode is always 1. Since Length = 1, Last Byte Enable (LBE) is automatically set to 4'b000.

   First Byte Enable (FBE) is alway s set to 4'b1111.
2. CPU-write to target address. This is a direct access to Talino™ EDC PCI Memory Spaces.
3. PeCORE recognizes the write access above as an access to PCI Memory Space. If POM is enabled (PeDRF_POMS0[0] = 1) (EP mode only. No need to set this in RC mode) and address falls within the range of POM Local (if EP mode) or Mem Base/Limit registers (if RC mode), PeCORE initiates the generation and transmission of MWr TLP. If outbound mapping is enabled (PeDRF_POMS0[1] = 1), original address is mapped to the equivalent POM PCI address prior to MWr TLP transmission. Note that indirect Memory Write requests are not affected by POM registers.
4. To check if the request is already done, check if PeDRF_MRSR0[0] = 1. Checking of PeDRF_MRSR0[0] can be done through polling or through interrupt. See PeDRF_MRSR0 description (Section 25.3.2.25 PeDRF_MRSR 0/1/2/3: Memory Request Status Register 0/1/2/3) for details.

25.2.3.4.2 Memory Read, Direct Mode

1. if 64-bit addressing mode is enabled, write the upper 32-bit source address to PeDRF_MRAHR*n*.
2. CPU-read the target address. This is a direct access to Talino™ EDC PCI Memory Spaces.

Payload Length of Memory Read in this mode is always 1. Since Length = 1, Last Byte Enable (LBE) is set to 4'b000.

First Byte Enable (FBE) is always set to 4'b1111.

3. PeCORE recognizes the read access above as an access to PCI Memory Space. If POM is enabled (PeDRF_POMS0[0] = 1) (EP mode only. No need to set this in RC mode) and address falls within the range of POM Local (if EP mode) or Mem Base/Limit registers (if RC mode), PeCORE initiates the generation and transmission of MRd TLP. If outbound mapping is enabled (PeDRF_POMS0[1] = 1), original address is mapped to the equivalent POM PCI address prior to MRd TLP transmission. Note that indirect Memory Read requests are not affected by POM registers.
4. The notification of the completion of the request is implied by an acknowledgment in ALB from PeCORE. An ALB acknowledgment from PeCORE means that the ALB data returned is the data returned by the target PCIe device. Once the request is completed in ALB, it is guaranteed that PeDRF_MRSR0[0] = 1.

25.2.3.4.3 Memory Write, Indirect Mode

1. Prepare FBE, LBE and other necessary controls in PeDRF_MRCR0.
2. If 64-bit addressing is enabled (PeDRF_MRCR0[0]=1), write the upper 32-bit destination address to PeDRF_MRAHR0.
3. Write the lower 32-bit destination address to PeDRF_MRALR0. Note that PeDRF_MRALR0[1:0] are Read-Only. Addresses are always word-aligned. Address bits [1:0] are always forced to zero.
4. Write data to PeDRF_MRDR0/1/2/3 registers.

   If Length = 4 dwords, write data from PeDRF_MRDR3 to PeDRF_MRDR0

   If Length = 3 dwords, write data from PeDRF_MRDR2 to PeDRF_MRDR0

   If Length = 2 dwords, write data from PeDRF_MRDR1 to PeDRF_MRDR0

   If Length = 1 dword, write data to PeDRF_MRDR0

   Write to PeDRF_MRDR0 always triggers generation and transmission of MWr TLP. Whatever the contents of PeDRF_MRDR0 registers will be transmitted.
5. To check if the request is done, check if PeDRF_MRSR0[0] = 1. Checking of PeDRF_MRSR0[0] can be done through polling or through interrupt. See PeDRF_MRSR0 description (Section 25.3.2.25 PeDRF_MRSR 0/1/2/3: Memory Request Status Register 0/1/2/3) for details.

25.2.3.4.4 Memory Read, Indirect Mode

1. Set up FBE, LBE and other necessary controls in PeDRF_MRCR0.
2. If 64-bit addressing mode is enabled (PeDRF_MRCR0[0]=1), write the upper 32-bit source address to PeDRF_MRAHR0.
3. Write the lower 32 bits source address to PeDRF_MRALR0. Note that PeDRF_MRALR0[1:0] are Read-Only. Addresses are always word-aligned. Address

bits [1:0] are always forced to zero.

4. Read PeDRF_MRDR0/1/2/3 registers.

   If Length = 4 dwords, read data from PeDRF_MRDR3 to PeDRF_MRDR0

   Read from PeDRF_MRDR3 triggers generation and transmission of MRd TLP.

   If Length = 3 dwords, read data from PeDRF_MRDR2 to PeDRF_MRDR0

   Read from PeDRF_MRDR2 triggers generation and transmission of MRd TLP.

   If Length = 2 dwords, read data from PeDRF_MRDR1 to PeDRF_MRDR0

   Read from PeDRF_MRDR1 triggers generation and transmission of MRd TLP.

   If Length = 1 dword, read data from PeDRF_MRDR0

   Read from PeDRF_MRDR0 always triggers generation and transmission of MRd TLP.

   PeDRF_MRDR0 returns the first word, PeDRF_MRDR1 the second, PeDRF_MRDR2 the third and PeDRF_MRDR3 the fourth word.

5. The notification of the completion of the request is implied by an acknowledgment in ALB from PeCORE. An ALB acknowledgment from PeCORE means that the ALB data returned is the data returned by the target PCIe device. Once the request is completed in ALB, it is guaranteed that PeDRF_MRSR0[0] = 1.

25.2.3.4.5 Memory Write, CSR-based Descriptor DMA

1. Set up FBE, LBE, Length and other necessary controls in PeDRF_MRCR.
2. If 64-bit addressing is enabled (PeDRF_MRCR[0]=1), write the upper 32-bit destination address to PeDRF_MRAHR.
3. Write the lower 32-bit destination address to PeDRF_MRALR. Note that PeDRF_MRALR[1:0] are Read-Only. Addresses are always word-aligned. Address bits [1:0] are always forced to zero.
4. Write the source address to PeDRF_MRLAR. Write access to PeDRF_MRLAR automatically initiates the DMA request.
5. To check if the request is done, check if PeDRF_MRSR[0] = 1. Checking of PeDRF_MRSR[0] can be done through polling or through interrupt. See PeDRF_MRSR description (Section 25.3.2.25 PeDRF_MRSR 0/1/2/3: Memory Request Status Register 0/1/2/3) for details.

25.2.3.4.6 Memory Read, CSR-based Descriptor DMA

1. Set up FBE, LBE, Length and other necessary controls in PeDRF_MRCR.
2. If 64-bit addressing is enabled (PeDRF_MRCR[0]=1), write the upper 32-bit source address to PeDRF_MRAHR.
3. Write the lower 32-bit source address to PeDRF_MRALR. Note that

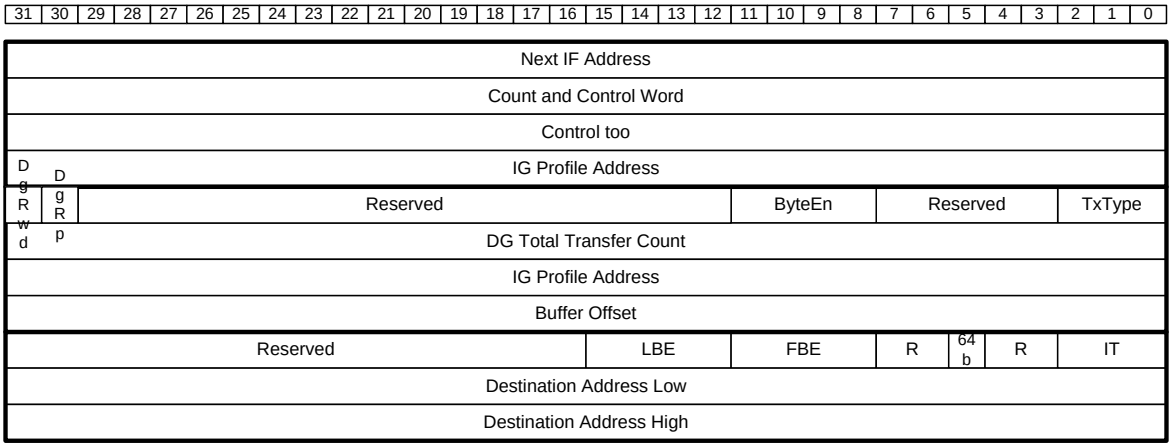
PeDRF_MRALR[1:0] are Read-Only. Addresses are always word-aligned. Address bits [1:0] are always forced to zero.

4. Write the destination address to PeDRF_MRLAR. Write access to PeDRF_MRLAR automatically initiates the DMA request.
5. To check if the request is done, check if PeDRF_MRSR[0] = 1. Checking of PeDRF_MRSR[0] can be done through polling or through interrupt. See PeDRF_MRSR description (Section 25.3.2.25 PeDRF_MRSR 0/1/2/3: Memory Request Status Register 0/1/2/3) for details.

25.2.3.4.7 Memory Write, ICBM Descriptor-based DMA

1. Assuming that some portion of the data to be transmitted is already available in the cache (SDRAM/SRAM), FW can already start creating an Instruction list to transmit the already available data. FW creates one Instruction Group (IGP) for the entire data to be transmitted. First Instruction Frames are to send the already available data in the cache. Since one MWr-Req Instruction Frame can only address one contiguous destination PCI memory space, each MWr-Req Instruction Frame is limited to one contiguous target buffer space. Figure 25.15 shows the format of the Instruction Frame that needs to be created to transfer data from Talino™ EDC Shared Memory to an outside PCIe device.

Figure 25.15: MWr-Req Instruction Frame



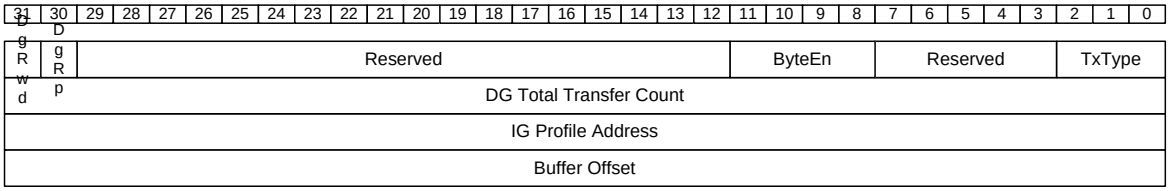
2. Enable ICBM TxDMA by writing 1 to BM_TXDCTRL[0].
3. Enable and unmask interrupts in BM_INTNORMDNE_EN, BM_INTNULLRCH_EN, BM_INTNORMDNE_MSK and BM_INTNULLRCH_MSK.
4. Load the address of the first Instruction Frame in BM_P0_INTNEXTFP_TXINST.

   ICBM TxDMA fetches the Instruction Frame from SDRAM/SRAM and saves it to one

of the command buffers of IOB.

The DCNT portion of the Instruction Frame (Figure 25.16) is posted by IOB to ICBM.TxDMA.

Figure 25.16: Tx Dcnt (MWr, TxType=0)



Through the posted DCNT, ICBM TxDMA fetches the data from SDRAM/SRAM and saves it to ICBM Tx Buffer.

To transfer data more efficiently, PeCORE immediately sends the available data from ICBM Tx Buffer once the minimum threshold for transmission is reached. This means that one Instruction Frame can have multiple equivalent Memory Write TLPs. PeCORE sends Memory Write Request TLP with payload size depending on the available data in ICBM TxBuffer. Memory Write Request TLPs are continuously sent until the accumulated data transmitted is equal to the Total Transfer Count of the Instruction Frame. PeCORE continuously sends MWr TLPs until all the succeeding Instruction Frames are exhausted.

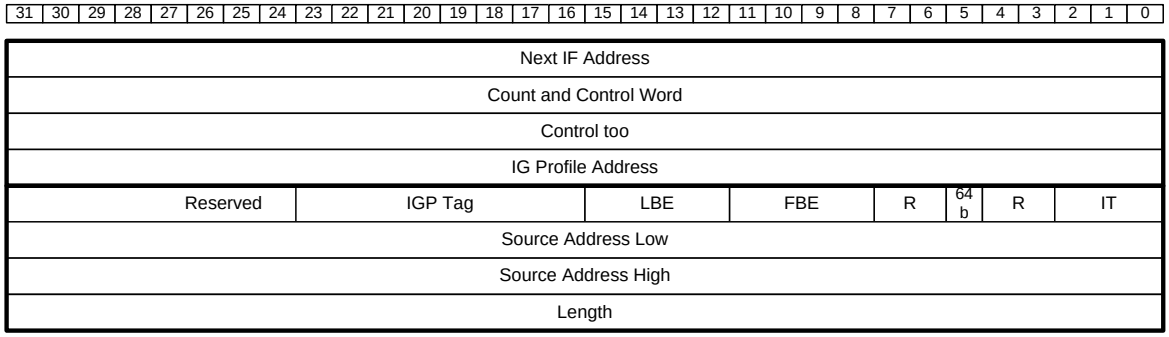
5. Wait for interrupt. An interrupt is sent to the local FW indicating the status of the data transmission. Normal done interrupt is reported if transmission is successful, otherwise, Error interrupt is reported.
6. When the interrupt is received, read BM_INTNORMMAIN_STT (if successful).
7. If BM_INTNORMMAIN_STT[0] = 1, read BM_INTNORMDNE_ADRQ. If the IG was completed successfully, BM_INTNORMDNE_ADRQ should return the address of the IG Profile.

25.2.3.4.8 Memory Read, ICBM Descriptor-based DMA

A. PeCORE sends Memory Read Requests

1. FW creates one Instruction Group (IG) for the entire data to be read from the target PCIe device. A tag is assigned to the IG. This will be used to map every Memory Read Completions that will be received by PeCORE. Assuming FW has initially allocated some data buffer space, FW can already start the transfer by creating Instruction Frames for this available buffer space. Since one MRd-Req Instruction Frame can only address one contiguous source PCI memory space, each MWr-Req Instruction Frame is limited to one contiguous source buffer space. Figure 25.17 shows the format of the Instruction Frame that needs to be created to transfer data from an outside PCIe device to Talino™ EDC Shared Memory.

Figure 25.17: MRd-Req Instruction Frame



2. Enable ICBM RxDMA by writing 1 to BM_RXDCTRL[0].
3. Enable and unmask interrupts in BM_INTNORMDNE_EN, BM_INTNULLRCH_EN, BM_INTNORMDNE_MSK and BM_INTNULLRCH_MSK.
4. Load the address of the first Instruction Frame in BM_P0_INTNEXTFP_TXINST.

   ICBM TxDMA fetches the Instruction Frame from SDRAM/SRAM and saves it to one of the command buffers of IOB.

   PeCORE fetches Instruction Frame from IOB Command Buffer. It breaks down the Instruction to multiple Memory Read Request with read request size equal to Max_Read_Request_Size (PeDRF_PEDSCTL[14:12]). Each Memory Read Request generated has unique tag number.

   PeCORE sends Memory Read Request TLPs until the accumulated read request size of all MRd TLPs sent is equal to the Instruction Frame total transfer count. MRd TLPs are continuously sent until all the succeeding Instruction Frames are exhausted.

B. PeCORE receives Memory Read Completions

1. PeCORE saves the Memory Read Completion Payload in ICBM Rx Buffer. Each Memory Read Completion is mapped to a corresponding IG. This is done by mapping the Completion tag (which is the same as the Memory Read Request tag) to its corresponding IGP tag. This IGP tag is returned to ICBM for each MRd Completion.
2. ICBM transfers the data to SDRAM/SRAM.
3. Wait for interrupt. An interrupt is sent to the local FW indicating the status of the data transmission. Normal done interrupt is reported if transmission is successful, otherwise, Error interrupt is reported.
4. When the interrupt is received, read BM_INTNORMMAIN_STT (if successful).
5. If BM_INTNORMMAIN_STT[0] = 1, read BM_INTNORMDNE_ADRQ. If the IG was completed successfully, BM_INTNORMDNE_ADRQ should return the address of the IG Profile.

25.2.3.5 Notes on Generating Request

It should be noted that, in generating request, care must be observed by FW that none of the rules like, but not limited to, the ones listed below is violated.

1. If the Length field for a request indicates a length of greater than 1 Word, then the First Word BE[3:0] field must not be 0000b.
2. If the Length field for a request indicates a length of greater than 1 Word, then the Last Word BE[3:0] field must not be 0000b.
3. If the Length field for a memory request indicates a length of 1 Word then the byte enables for the Last Word BE[3:0] field must be 0000b.
4. Non-contiguous Byte Enables (two or more enabled bytes separated by non-enabled bytes) must not be used for Memory requests of 3 Words or greater, or for Memory requests of 2 Words that are not Double-Word-aligned.
5. Requesters must ensure that Memory requests must not specify an Address/Length combination which causes a memory space access to cross a 4KB boundary.

Some rules, however, like, but not limited to, the ones listed below are taken care of by PeCORE hardware. PeCORE ensures that they are not violated.

1. Requesters must ensure that for Memory Read requests, the request Length must not exceed the value specified by Max_Read_Request_Size field.
2. The Transmitter of a TLP with a data payload as given by the TLP's length field must not allow the data payload length to exceed the length specified by the value in the Max_Payload_Size field of the Transmitter's Device Control register taken as an integral number of Word.
3. Requesters must ensure that I/O requests must have the TC0[2:0] field equal to 000b.
4. Requesters must ensure that I/O requests have the Attr[1:0] field equal to 00b.
5. Requesters must ensure that I/O requests have the Length[9:0] field equal to 00 0000 0001b (1 Word).
6. Requesters must ensure that I/O requests have the Last Word BE[3:0] field equal to 0000b.
7. Requesters must ensure that Configuration requests must have the TC0[2:0] field equal to 000b.
8. Requesters must ensure that Configuration requests have the Attr[1:0] field equal to 00b.
9. Requesters must ensure that Configuration requests have the Length[9:0] field equal to 00 0000 0001b (1 Word).
10. Requesters must ensure that Configuration requests have the Last Word BE[3:0] field equal to 0000b.

25.2.3.6 Reception Of Configuration Request

Reception and processing of Configuration requests are handled by PeCORE without FW intervention.

25.2.3.7 Reception Of IO Request

Reception and processing of IO requests are handled by PeCORE without FW intervention. Note that IO request, for PeCORE, is designed to access only resources local to PeCORE, meaning the Host or Root Complex cannot access anything beyond PeCORE through IO request.

25.2.3.8 Reception Of Memory Request

Reception and processing of Memory requests are handled by PeCORE without FW intervention.

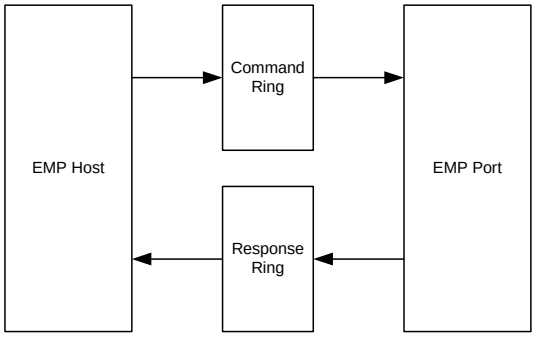
25.2.3.9 Exchange Message Protocol

PCI Express is basically a low-level IO interface. What it provides are the basic transactions needed for two PCI Express devices (each at the end of the link) to communicate. Unlike SAS or SATA, PCI Express requires some high-level messaging protocol for two PCI Express devices to communicate with some sense. PeCORE provides Exchange Message Protocol (EMP), a high-level messaging protocol, for this purpose.

In this protocol, the downstream device (like an Endpoint) is called the EMP Port, while the upstream device (like a Root Complex) is called the EMP Host.

EMP Host and EMP Port communicate and pass their messages to each other through Command and Response Rings. These Rings have programmable number of entries, and each entry has fixed size and is called an *IOFrame*. An entry in the Command Ring is called a *Command IOFrame*, while an entry in the Response Ring is called a *Response IOFrame.* EMP Host posts its messages for the EMP Port to the Command Ring, while the EMP Port posts its messages for the EMP Host to the Response Ring. EMP Port, therefore, drains the Command Ring, while the EMP Host drains the Response Ring. Figure 25.18 illustrates the flow of messages between EMP Host and EMP Port.

Figure 25.18: Flow of messages in EMP



Both Command and Response Rings have Put and Get pointers. Put pointer for posting an IOFrame, and Get pointer for draining it. Pointers for Command Ring are called Command Put Pointer and Command Get Pointer, while pointers for Response Ring are called Response Put Pointer and Response Get Pointer. All these pointers are registers located in EMP Port. Command Put Pointer and Response Get Pointer are owned by the EMP Host, while the Response Put Pointer and the Command Get Pointer are owned by the EMP Port. Refer to Table 25.7 for the list of these EMP pointers.

Table 25.7: EMP Pointers

Pointer Name	Owner	Location
Command Put Pointer (CPP)	EMP Host	EMP Port
Command Get Pointer (CGP)	EMP Port	EMP Port

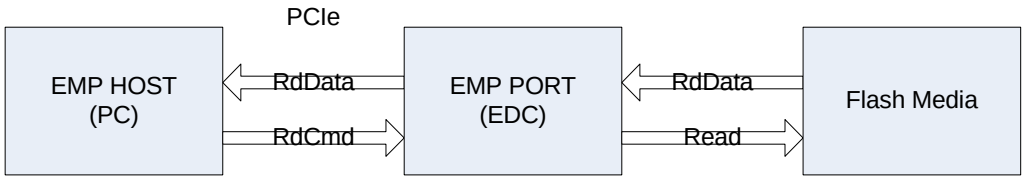
Pointer Name	Owner	Location
Response Put Pointer (RPP)	EMP Port	EMP Port
Response Get Pointer (RGP)	EMP Host	EMP Port

Note that all pointers point to a location in the EMP Host memory. EMP has upto 8 pairs of Command/Response Rings-- Ring 0 - 7.

25.2.3.9.1 EDC Receiving an IO (Read) Command

EMP is used as an efficient messaging interface between EDC, functioning as Endpoint, and the Host PC. The following is a sample process flow that happens in a PCIe-SSD application, where EDC acts as a PCIe controller and flash controller for the PCIe-SSD device. In this application EDC functions as the EMP Port (Endpoint) and the Host PC as the EMP Host (Root Complex). The scenario is, the Host PC issues an IO Read command to EDC. See Figure 25.19 for the reference diagram.

Figure 25.19: EDC Receiving an IO (Read) Command



Ring 0 is selected as the Command/Response Ring for this transaction.

Initially, `CPP0` = `CGP0` and `RPP0` = `RGP0`.

1. EMP Host creates a Command IOFrame, which contains the IO Read command, in the location pointed to by the Command Put Pointer 0 (`CPP0`) register.

   Note that all Pointer Registers do not directly point to the location of an IOFrame. The value a Pointer Register holds is just an index that can be used to locate the exact location of an IOFrame. What is known from the start is the start address of the Command/Response Ring. The size of each IOFrame entry is fixed and also known. From these available informations, the exact location of an IOFrame can be known.

2. EMP Host issues a PCIe MWr to EMP Port to increment `CPP0`. This automatically sets bit 16 of the *Port Attention Register* (PeDRF_PATR), which in turn sets PeDRF_CINT[2]. An interrupt is, then, sent to EMP Port's local FW.

3. EMP Port reads the IOFrame pointed to by the Command Get Pointer 0 (CGP0) register from EMP Host memory.
4. EMP Port increments CGP0.
5. EMP Port prepares all the requested data in its cache.
6. EMP Port builds the (ICBM) Instruction List that will DMA the requested data along with the response frame to EMP Host memory. One or more MWr InstFrames per “data buffer” or contiguous destination memory space can be used, but never more than one “data buffer” per MWr InstFrame. One MWr InstFrame can address only one contiguous memory space.
7. EMP Port DMAs the requested data along with the response frame to EMP Host memory.
8. EMP Port builds a Response IOFrame in the location pointed to by the Response Put Pointer 0 (RPP0) register.
9. EMP Port increments RPP0. This automatically sets bit 16 of the *Host Attention Register* (PeDRF_HATR), which in turn triggers the transmission of an interrupt (MSI or INTx) to EMP Host.
10. EMP Host receives the interrupt and reads the Response IOFrame pointed to by the Response Get Pointer 0 (RGP0) register from EMP Host memory.
11. EMP Host issues a PCIe MWr to increment RGP0.

25.2.3.9.2 Command Ring Full

1. EMP Host checks if CPP0, when incremented, would be equal to CGP0. If CPP0 would be equal to CGP0, this means that the Command Ring is full.
2. EMP Host aborts creation of a new Command IOFrame.
3. EMP Host issues a PCIe MWr to set bit 0 and bit 16 of the *Port Attention Register* (PeDRF_PATR) to request notification from EMP Port when, at least, one Command Ring entry has been freed up. This automatically sets PeDRF_CINT[2], and an interrupt is sent to EMP Port's local FW.
4. EMP Port sets bit 8 and bit 16 of the *Host Attention Register* (PeDRF_HATR) after it has freed up at least one Command Ring entry. This triggers transmission of an interrupt (MSI or INTx) to EMP Host.
5. EMP Host receives the interrupt and can now build a new Command IOFrame.

25.2.3.9.3 Response Ring Full

1. EMP Port checks if RPP0, when incremented, would be equal to RGP0. If RPP0 would be equal to RGP0, this means that the Response Ring is full.
2. EMP Port aborts creation of a new Response IOFrame.
3. EMP Port sets bit 0 and bit 16 of the *Host Attention Register* (PeDRF_HATR) to request notification from EMP Host when, at least, one Response Ring entry has been freed up. This triggers transmission of an interrupt (MSI or INTx) to EMP Host.

4. EMP Host receives the interrupt.
5. EMP Host, after it has freed up at least one Response Ring entry, issues a PCIe MWr to set bit 8 and bit 16 of the *Port Attention Register* (PeDRF_PATR). This automatically sets PeDRF_CINT[2], and an interrupt is sent to EMP Port's local FW.
6. EMP Port can now build a new Response IOFrame.

25.2.3.10 SRAM Test

25.2.3.10.1SRAM Test Mode - Reading and Writing

During SRAM test mode, the SRAM test address is used as the input address to the SRAM read and write port. In this mode, the SRAM test data is used to access the data lines of the SRAM read and write port.

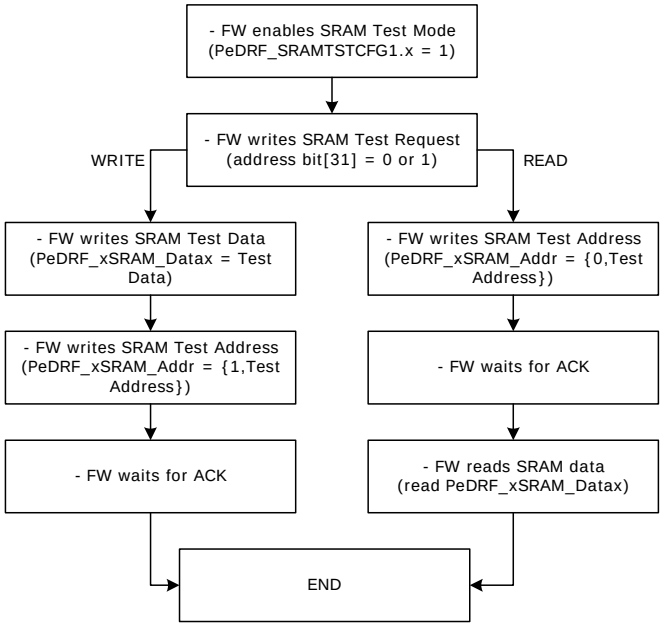
SRAM test read procedure:

- Firmware writes the SRAM address in the SRAM Test Address register (See Section 25.3.2.63 to Section 25.3.2.70), with bit[31] = 1'b0 (Wr1Rd0 = 1'b0)
  - This write action triggers the test mode logic to read from the specified SRAM location and load the read data in the corresponding SRAM test data register(s) (there are more than one test data register if the SRAM width spans more than 32 bits, for example:)
    - SRAM Test Data W0 register => SRAM data bits [31:00]
    - SRAM Test Data W1 register => SRAM data bits [63:32]
    - SRAM Test Data W2 register => SRAM data bits [95:64] ...
- Firmware reads the corresponding SRAM test data register(s)

SRAM test write procedure:

- Firmware writes the data to the corresponding SRAM test data register(s)
- Firmware writes the SRAM address in the SRAM Test Address register (See Section 25.3.2.63 to Section 25.3.2.70), with bit[31] = 1'b1 (Wr1Rd0 = 1'b1)
  - This write action triggers the test mode logic to write the data from the SRAM test data register(s) to the specified SRAM location.
  - This step can be repeated for other SRAM locations if the same needs to be written to the other SRAM locations.

Figure 25.20: Flow for SRAM Test Mode



25.2.3.10.2SRAM Test Parity Error Injection

If enabled, parity checking / interrupt generation is done during SRAM test mode. Each group of SRAM buffers may be injected with parity errors by setting to 1'b1 the corresponding bits in PeDRF_TSRAM_PE1 register. The steps for testing with parity errors are as follows:

- Enable SRAM test mode by writing 1'b1 to corresponding bits in PeDRF_SRAMTSTCFG1 (See Section 25.3.2.61)
- Enable parity error injection by writing 1'b1 to corresponding bits in PeDRF_TSRAM_PE1 (See Section 25.3.2.62)
- Do a SRAM test write procedure.
- Do a SRAM test read procedure.
- The appropriate interrupt for parity error should be asserted.

25.2.3.10.3SRAM Test Address, Generic Format

Table 25.8: Test Address Generic Format

Bit Field	Type	Default	Description
[N:00]	RW	(N+1)'h0	Test Address [N:0]
[30:N+1]	RO		Reserved
[31]	RW	1'b0	Wr1Rd0, Write1Read0

[N:00] Test Address [N:0]

Specifies the SRAM location where the read/write test is done.

[31] Wr1Rd0, Write1Read0

This bit specifies if a write or read access is to be done on the specified SRAM location. If a write access is performed, SRAM test data register(s) contents is written on the specified SRAM location. If a read access is performed, data from the specified SRAM location is loaded to the SRAM test data register(s).

Value	Function
1'b0	SRAM test read access
1'b1	SRAM test write access

25.2.3.10.4SRAM Test Data, Generic Format

Table 25.9: Test Data Generic Format

Bit Field	Type	Default	Description
[N:00]	RW	(N+1)'h0	Test Data [N:0]
[31:N+1]	RO		Reserved

[N:00] Test Data [N:0]

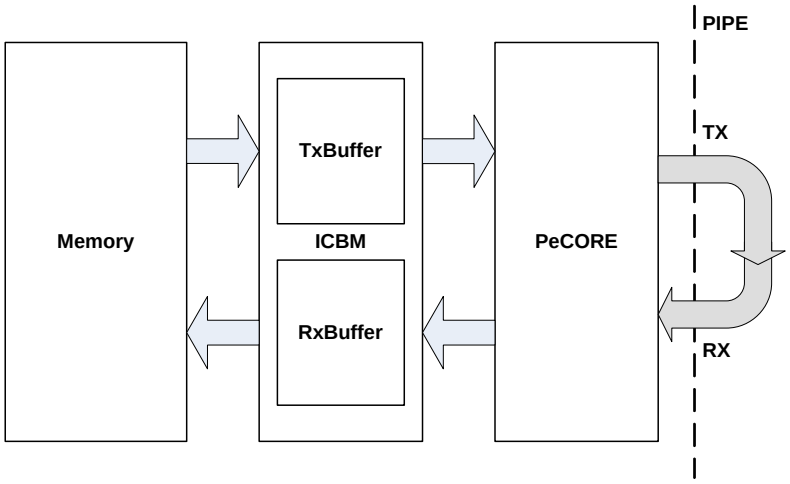
Contains write data for the SRAM test write access. Contains read data loaded from the SRAM during SRAM read test access.

25.2.3.11 Self-Loopback

This section describes the procedure in doing a self-loopback test. In this loopback test, data is transmitted from memory to ICBM.TxBuffer, then to PeCORE, then back to PeCORE when data reach PIPE, then from PeCORE to ICBM.RxBuffer, then back to memory. See Figure 25.21 for reference diagram.

In the procedure below, the mode used in generating memory request is CSR-DMA mode.

Figure 25.21: Self-Loopback Diagram



Memory Write

1. FW initiates generation of PCIe Memory Write request.
2. PeCORE receives the request, then instructs ICBM to fetch the data from the memory.
3. ICBM fetches the data from the memory.
4. PeCORE appends TLP header to the data, then issues MWr TLP (Seq# = m).
5. Loopback (for MWr TLP (Seq# = m)).
6. PeCORE receives MWr TLP (Seq# = m).
7. PeCORE stores MWr data payload to ICBM.RxBuffer. PeCORE also issues Ack DLLP (Seq# = m) for the received MWr TLP (Seq# = m).
8. ICBM DMAs MWr data payload to the memory.
9. Loopback (for Ack DLLP (Seq# = m)).
10. PeCORE receives Ack DLLP (Seq# = m) for the transmitted MWr TLP (Seq# = m).

Memory Read

1. FW initiates generation of PCIe Memory Read request.
2. PeCORE receives the request, then issues MRd TLP (Seq# = m; Tag# = n).
3. Loopback (for MRd TLP (Seq# = m; Tag# = n)).
4. PeCORE receives MRd TLP (Seq# = m; Tag# = n), then instructs ICBM to fetch the requested data from the memory. PeCORE also issues Ack DLLP (Seq# = m) for the received MRd TLP (Seq# = m; Tag# = n).
5. Loopback (for Ack DLLP (Seq# = m)).
6. PeCORE receives Ack DLLP (Seq# = m) for the transmitted MRd TLP (Seq# = m; Tag# = n).
7. PeCORE, after receiving the requested data from ICBM, issues CplD TLP (Seq# = p; Tag# = n).
8. Loopback (for CplD TLP (Seq# = p; Tag# = n)).
9. PeCORE receives CplD TLP (Seq# = p; Tag# = n).
10. PeCORE stores the Completion data payload to ICBM.RxBuffer. PeCORE also issues Ack DLLP (Seq# = p).
11. Loopback (for Ack DLLP (Seq# = p)).
12. PeCORE receives Ack DLLP (Seq# = p) for the transmitted CplD TLP (Seq# = p; Tag# = n).

25.2.4 Error-Handling

Errors in PCI Express can be divided into two categories: Correctable and Uncorrectable errors.

Recovery from correctable errors is documented in the PCI Express standard specification, and is handled by the hardware without FW intervention; while recovery from uncorrectable errors is application-specific, and outside the scope of PCI Express specification. Uncorrectable error can be considered Fatal or Non-Fatal depending on its Severity setting.

This section will discuss mainly how PeCORE handles the different kinds of uncorrectable errors. For more details on how correctable errors are handled, refer to PCI Express standard specification.

25.2.4.1 Correctable Errors

25.2.4.1.1 Receiver Error

PeCORE detects and reports the following types of receiver errors.

1. PHY Elastic Buffer Overflow
2. PHY Elastic Buffer Underflow
3. 8b10b Decode Error
4. Disparity Error
5. Loss of Symbol Lock
6. Lane-to-lane Skew Error (exceeding 6 symbols)
7. Framing Error (missing END symbol, missing START symbol, bad placement of START symbol)

All these receiver errors, when detected, are reported in PeDRF_RXERR, but only the 8b10b Decode Error can set the “Receiver Error Status” bit of PeDRF_AECESTA. Specific number of 8b10b Decode Error occurrences, however, must be detected first before this bit is asserted. This number can be set through PeDRF_DERRL.

Receiver errors are detected at the Physical Layer. They are, however, handled at the Link Layer. Packet with receiver error is passed to the Link Layer from Physical Layer with the indication that this packet has receiver error. When received at the Link Layer, this packet is discarded. NAK DLLP is scheduled for 8b10b Decode Error, Disparity Error, Loss of Symbol Lock and Framing Error. NAK DLLP, however, may or may not be scheduled for PHY Elastic Buffer Overflow/Underflow and Lane-to-lane Skew Error. These receiver errors also trigger Link-Retraining. Link-Retraining is also triggered when 8b10b Decode Error exceeds its count limit.

NAK’ed TLP is retried by the source of the TLP. Note that four consecutive NAKs on the same packet also triggers Link-Retraining.

25.2.4.1.2 Bad TLP

Bad TLPs are those TLPs that either have Link CRC (LCRC) error or invalid Sequence Number. These errors are detected and handled at the Link Layer. NAK is scheduled when a Bad TLP is detected. Upon reception of NAK, the source of Bad TLP retries the packet.

25.2.4.1.3 Bad DLLP

Bad DLLPs are those DLLPs that have CRC error. They are discarded by the receiver of that packet, without the knowledge of the sender. Note that this action sometimes may cause REPLAY_TIMER timeout if the Bad DLLP is an Ack or Nak DLLP.

25.2.4.1.4 REPLAY_NUM Rollover

This kind of error is reported when 4 consecutive NAKs are received for the same packet. Link-Retraining is triggered by this error.

25.2.4.1.5 Replay Timer Timeout

This error is reported when the maximum time limit the other device should respond with an ACK or NAK DLLP to a received TLP expires.

25.2.4.1.6 Advisory Non-Fatal Error

Some uncorrectable errors, when their Severity setting is NON-FATAL, are reported by PeCORE as Advisory Non-Fatal Error. These uncorrectable errors are the following:

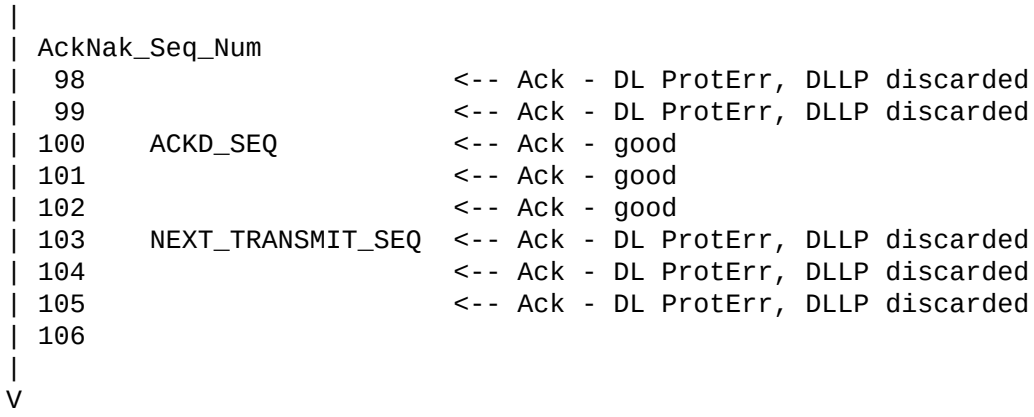
1. Unsupported Request
2. Unexpected Completion
3. Completer Abort
4. Completion Timeout
5. Poisoned TLP

25.2.4.2 Uncorrectable Errors

25.2.4.2.1 Data Link Protocol Error

This error is reported when the AckNak_Seq_Num of an ACK/NAK DLLP does not correspond to an unacknowledged TLP or to the value in ACKD_SEQ (last acknowleged Seq_Num). This is illustrated in Figure 25.22.

Figure 25.22: Data Link Protocol Error



25.2.4.2.2 Surprise Down Error

This error is reported when DLCMSM changed from DL_Active to DL_Inactive state, except when the transition is due to the following cases:

1. Secondary Bus Reset in Bridge Control Register is set to 1
2. Link Disable bit is set to 1
3. PME_Turn_Off message is sent through this port

25.2.4.2.3 Poisoned TLP

Data Poisoning is used when the data to be sent is known to have an error even before its transmission. In this case, the EP field of the TLP Header is set to 1 to indicate to the receiver of the packet that the packet has an error. Note that Data Poisoning applies only to packet with data payload.

How Data Poisoning is handled both in received and transmitted packet is described here.

25.2.4.2.3.1 RX

If Completion Data

ICBM-DMA Mode

If PeDRF_COPT[8] (Discard Poisoned TLP) is set,

If multiple completions

```
CplD A <-- good
CplD B <-- poisoned, discarded
CplD C <-- good, discarded
CplD D <-- good, discarded
```

1. PeCORE requests FW retry through "IG Update" as soon as the poisoned completion (CplD B) is detected.
2. Pending status of NP request is cleared/completed.
3. MRd request is retried starting from CplD B.
4. All completions following CplD B, even if for different request but of the same IG, will be discarded. All outstanding non-posted requests under the same IG will be removed from the queue, hence, when their corresponding completions are received, they will be discarded and considered as unexpected completions.
5. As opposed to previous scheme, PeCORE no longer waits for Completion Timeout before requesting FW retry through "IG Update". This is so, because the completer (the one that poisoned the completion) won't retry the poisoned completion. If retried, that would cause problem to the calculation of the remaining data to be returned.

If single completion

1. Pending status of NP request is cleared/completed.
2. Data is not forwarded to ICBM.
3. FW retry is requested through "IG update".

If PeDRF_COPT[8] (Discard Poisoned TLP) is not set,

Completion is treated like a normal non-poisoned completion

1. Pending status of NP request is cleared/completed
2. Data is forwarded to ICBM

CSR-DMA Mode

If PeDRF_COPT[8] (Discard Poisoned TLP) is set,

If multiple completions

```
CplD A <-- good
CplD B <-- poisoned, discarded
CplD C <-- good, discarded
CplD D <-- good, discarded
```

1. PeCORE automatically retries the request as soon as the poisoned completion (CplD B) is detected.
2. Pending status of NP request is cleared/completed.
3. MRd request is retried starting from CplD A, the very beginning of the request.
4. All completions following CplD B will be discarded.
5. As opposed to previous scheme, PeCORE no longer waits for Completion Timeout before retrying the request. This is so, because the completer (the one that poisoned the completion) won't retry the poisoned completion. If retried, that would cause problem to the calculation of the remaining data to be returned.

If single completion

1. Pending status of NP request is cleared/completed
2. PeDRF_MRSR[0] (Done) is not set
3. Data is not forwarded to ICBM
4. MRd request is automatically retried by the hardware as a new request (new TLP Tag).

If PeDRF_COPT[8] (Discard Poisoned TLP) is not set,

1. Pending status of NP request is cleared/completed
2. PeDRF_MRSR[0] (Done) is set
3. Data is forwarded to ICBM

If MWr Data

If PeDRF_COPT[8] (Discard Poisoned TLP) is set,

1. Data is not forwarded to ICBM

If PeDRF_COPT[8] (Discard Poisoned TLP) is not set,

1. Data is forwarded to ICBM

25.2.4.2.3.2 TX

If Completion Data

If multiple completions

If 1st completion is poisoned

```
CplD A <-- poisoned. won't be resent
CplD B <-- good
CplD C <-- good
CplD D <-- good
```

If poisoned completion is not first

```
CplD A <-- good
CplD D <-- good
CplD C <-- poisoned. won't be resent
CplD D <-- good
```

If single completion

```
CplD <-- poisoned. won't be resent
```

Notes:

1) PeCORE does not have recovery for this error. The burden of recovery is passed to the originator of the request. The requester of MRd has the option to retry the request or not.

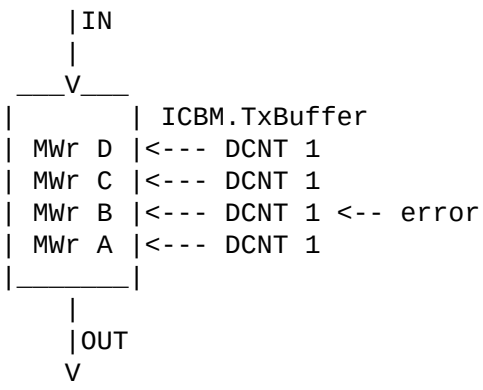
2) PeCORE cannot resend poisoned CplD right after sending the poisoned completion. That will cause a protocol error.

If MWr Data

Packet is poisoned when Memory-to-ICBM.TxBuffer parity error indication (TxDat_TagInfo[3]) is detected and the TLP header for that data with parity error is not yet transmitted.

The flow in Table 25.23 applies both to ICBM-DMA and CSR-DMA.

Figure 25.23: Retry of Poisoned MWr Data



1. MWr A already sent when error is detected in MWr B.
2. MWr B is poisoned.

   If PeDRF_COPT[9]==0, no hw retry (TxDat_Status=OK).

   If PeDRF_COPT[9]==1, hw retry is initiated (TxDat_Status=RETRY).

   During retry, ICBM.TxBuffer is automatically paused, then flushed. DCNT 1, along with other DCNTs flushed when retry is issued, then, is automatically reposted.
3. If PeDRF_COPT[9]==0, MWr C and D are sent.

   If PeDRF_COPT[9]==1, MWr C and D are not sent.
4. When retried, entire DCNT 1 is retried starting from MWr A. Retried data may not be the same MWr TLPs as the original TLPs (MWr A/B/C/D). Entire DCNT, for example, may be retried as 1 MWr TLP if the entire data is available in TxBuffer when retry is started.

Note to FW:

The following scenario happened in one of our ASIC simulations:

1. PeCORE issued one CSR-DMA MRd Request that was divided into multiple MRd TLPs.
2. 4th MRd Cpl received was poisoned. This was received while PeCORE was still gen-

erating the succeeding MRd TLPs for the request in #1.

3. Upon reception of the first poisoned Cpl, PeCORE aborted all pending MRd TLPs related to the request in #1, then issued a retry of the same request.
4. The problem was, PeCORE, since it was not yet done issuing MRd TLPs for the original request, continuosly issued MRd TLPs after the supposed aborting of the said request happened. Because of this, some CplD (completions) for the original request reached memory, because their corresponding MRd requests were not included in the abortion of request. Refer to Figure 25.24 for the flow of transactions.

Figure 25.24: ASIC Simulation: Not All MRds are Aborted

```
+-+--------+----+--+-+
|D|        |  S | T| |
|I|        |  E | A|E|
|R|COMMAND |  Q | G|P|
+-+--------+----+--+-+
 D    MRd32   91 00 0 <-- completed
 D    MRd32   92 01 0 <-- completed
 U     CplD   58 00 0 <-- good
 D    MRd32   93 02 0 <-- completed
 D    MRd32   94 03 0 <-- completed, but Cpl is poisoned
 U     CplD   59 01 0 <-- good
 D    MRd32   95 04 0 <-- aborted
 U     CplD   60 02 0 <-- good
 U     CplD   61 03 1 <-- 1st poisoned Cpl
 U     CplD   62 03 1 <-- discarded
 U     CplD   63 04 1 <-- discarded
 U     CplD   64 04 1 <-- discarded

 D    MRd32   96 00 0 <-- aborted
 D    MRd32   97 01 0 <-- aborted
 D    MRd32   98 02 0 <-- not aborted, not yet issued when abortion
happened
 U     CplD   65 00 0 <-- discarded, because MRd32 96 was aborted
 U     CplD   66 01 0 <-- discarded, because MRd32 97 was aborted
 U     CplD   67 02 0 <-- reached memory, MRd32 98 was not aborted
```

This is ok, because

First, in actual programming, it is recommended that we restrict our generation of MRd TLP to 1 IG = 1 MRd TLP (or 1 CSR-DMA Request = 1 MRd TLP). This means that with this restriction, the above scenario won't happen.

Secondly, even if we don't restrict our generation of MRd TLPs to 1 MRd TLP per CSR-DMA request (or 1 IG), FW can just ignore Cpl (completions) that reach the memory. The request will be retried from the start anyway. Retry during ICBM-DMA is also recommended to be restarted from the start of the request, since some poisoned Cpl that may reach the memory will unexpectedly update the request rewind pointer.

25.2.4.2.4 Flow Control Protocol Error

This error is reported when, at least, one of the following rules is violated:

1. Each credit type must advertise an initial credit value not less than the minimum requirement specified in the PCI Express specification.

2. An UpdateFC DLLP is not required to be sent for any credit type advertised as infinite during FC Initialization. If an UpdateFC DLLP, however, is sent for that credit type, its credit value must still be infinite.

3. A receiver must never cumulatively issue unused credits to the Transmitter more than 2047 for data payload or 127 for header.

25.2.4.2.5 Completion Timeout

This error is reported when the Completion timer for a non-posted request expires and still no Completion is received for that request.

This error is handled differently for each mode of generation of request of PeCORE. Note that Completion Timeout is configurable through PeDRF_PEDCAP2 and PeDRF_PEDSCTL2.

Direct / Indirect (Config) and IO Mode

This mode has automatic hardware retry during Completion Timeout. Completion Timeout status is reported only when 3 completion timeouts are detected for the same request. If request is read, the data returned for that request is 0xffff_ffff.

1. 1st Non-posted Request issued - Completion Timeout - 1st HW retry
2. 2nd Non-posted Request issued - Completion Timeout - 2nd HW retry
3. 3rd Non-posted Request issued - Completion Timeout - no more HW retry

   PeDRF_CRDR = 0xffff_ffff (if read)

   PeDRF_CRSR[3] = 1

   PeDRF_AEUESTA[14] = 1

Direct / Indirect (Mem) Mode

This mode has automatic hardware retry during Completion Timeout. Completion Timeout status is reported only when 3 completion timeouts are detected for the same request. If request is read, the data returned for that request is 0xffff_ffff.

1. 1st Non-posted Request issued - Completion Timeout - 1st HW retry
2. 2nd Non-posted Request issued - Completion Timeout - 2nd HW retry
3. 3rd Non-posted Request issued - Completion Timeout - no more HW retry

   PeDRF_MRDR0 = 0xffff_ffff (if read)

   PeDRF_MRSR0[0] = 1

   PeDRF_MRSR0[6] = 1

   PeDRF_AEUESTA[14] = 1

CSR-DMA Mode

This mode has automatic hardware retry during Completion Timeout. Completion Timeout status is reported only when 3 completion timeouts are detected for the same request. If request is read, the data returned for that request is 0xffff_ffff.

1. 1st Non-posted Request - Completion Timeout - 1st HW retry
2. 2nd Non-posted Request - Completion Timeout - 2nd HW retry

3. 3rd Non-posted Request - Completion Timeout - no more HW retry
   PeDRF_MRDR0-3 = 0xffff_ffff (if read)
   PeDRF_MRSRn[0] = 1
   PeDRF_MRSRn[6] = 1
   PeDRF_AEUESTA[14] = 1

ICBM-DMA Mode

Unlike the previous modes, this mode does not support automatic hardware retry during Completion Timeout. Completion Timeout status (PeDRF_AEUESTA[14]) is reported the first time it is detected.

1. 1st Non-posted Request - Completion Timeout - No retry
   PeDRF_AEUESTA[14] = 1
   Status Code field of Status Gobble of IG Profile = 0x6 (Completion Timeout)
   BM_INTGENERR1_EN[9] = 1
   Refer to ICBM section for the related details on handling error in an IG.

25.2.4.2.6 Unsupported Request / Completer Abort (UR/CA)

This section discusses how PeCORE handles Unsupported Request and Completer Abort completion status both in RX (when the issued non-posted request by PeCORE is terminated by the completer with UR/CA completion status) and in TX (when the received non-posted request by PeCORE must be terminated with UR/CA completion status).

25.2.4.2.6.1 RX

Config/IO Request

1. When completion with non-successful status is received, pending status of the (non-posted) request is cleared/completed.
2. Lc_CS is acknowledged (Lc_Ack is asserted)
3. PeDRF_CRSR[2:0] is updated.
4. If CplStat=UR/CA and request is CfgRd

   PeCORE returns a 0xffff_ffff value to the software.

   If CplStat=CRS

   if Cfg Request is a CfgRd to the VendorID, and CRS Software Visibility is enabled

   PeCORE returns a 0xffff_0001 value to the software

   else

   Cfg Request is automatically retried by the hardware

ICBM-DMA

If single completion

1. When completion is received, pending status of the (non-posted) request is cleared/completed.
2. Status Code field of Status Gobble of IG = 0x01 (CA), 0x02 (UR)

   BM_INTGENERR1_EN[9] = 1

   Refer to ICBM section for the related details on handling error in an IG.

If multiple completions and first completions are successful, then followed by an unsuccessful completion

1. Successful completion payloads are forwarded to ICBM as good data.
2. Running/rewind pointers are updated.

3. When unsuccessful completion is received, pending status of the (non-posted) request is cleared/completed.
4. Status Code field of Status Gobble of IG = 0x01 (CA), 0x02 (UR)

   BM_INTGENERR1_EN[9] = 1

   Refer to ICBM section for the related details on handling error in an IG.

If multiple completions and first completion is unsuccessful. Theoretically, when a read completion is received with non-successful status, that completion must be the last completion for the request.

1. When the first completion is received, pending status of the (non-posted) request is cleared/completed.
2. Running/rewind pointers are not updated.
3. Status Code field of Status Gobble of IG = 0x01 (CA), 0x02 (UR)

   BM_INTGENERR1_EN[9] = 1

   Refer to ICBM section for the related details on handling error in an IG.
4. Succeeding completions are discarded and treated as unexpected completions.

CSR-DMA

If single completion

1. When completion is received, pending status of the (non-posted) request is cleared/completed.
2. PeDRF_MRSRn[0] is set.
3. PeDRF_MRSRn[5:3] reflects the status of the completion.

If multiple completions, and first completions are successful, then followed by an unsuccessful completion

1. Successful completion payloads are forwarded to ICBM as good data.
2. When completion with non-successful status is received, pending status of the (non-posted) request is cleared/completed.
3. PeDRF_MRSRn[0] is set.
4. PeDRF_MRSRn[5:3] reflects the status of the completion.

If multiple completions, and first completion is unsuccessful. Theoretically, when a read completion is received with non-successful status, that completion must be the last completion for the request.

1. When completion with non-successful status is received, pending status of the (non-

posted) request is cleared/completed.

2. PeDRF_MRSRn[0] is set.
3. PeDRF_MRSRn[5:3] reflects the status of the completion.
4. Succeeding completions are discarded and treated as unexpected completions.

25.2.4.2.6.2 TX

Unsupported Request

PeCORE replies with completion with unsupported request status to the following non-posted requests listed below. Replying with UR for each type of request can be disabled through PeDRF_URCTL.

1) Locked MRd

2) CfgRd1/CfgWr1

3) Msg Request not supported

4) Request not within the range of any of the BARs

5) Request received while in D1 or D2

Completer Abort

PeCORE replies with completion with completer abort status to the following non-posted requests listed below. Replying with CA for each type of request can be disabled through PeDRF_CACTL.

1) Memory Rd/Wr to a control register, but size is more than 1

2) Request size goes out of range of the BAR

3) Memory request accesses an invalid AWB address

4) IO request accesses an AWB address

25.2.4.2.7 Unexpected Completion

This error is reported when a completion that matches no outstanding non-posted request is received by PeCORE. Unexpected completions are simply discarded by PeCORE.

25.2.4.2.8 Receiver Overflow

This error is reported when any of the six credit types overflows. This kind of error must generally not be reported. A reported receiver overflow means that PeCORE has some design error. There is, however, one exception when assertion of this kind of error has FW workaround.

The maximum number of completion header credits for MRd request PeCORE supports is 32. Infinite credit, however, is advertised for completion header during FC initialization. Since the actual credit is not infinite, to avoid overflow, PeCORE controls the generation of MRd request in such a way that no MRd request will be generated if there are no available header credits to receive the completions. This means that the required header credit is explicitly allocated for the expected completions. The question now that arises is, how many credit headers must be allocated per MRd request. If the completer is sure to reply with single completion, the answer is easy. What if the completer replies with multiple completions? For this purpose, register PeDRF_CPMR can be used to program the expected number of completions per MRd request. If the value of this register, for example, is 8, 8 is deducted from the available completion header credits per MRd request generated. This means that, for a PeDRF_CPMR value of 8, PeCORE can simultaneously issue upto 4 MRd requests at a time.

The problem arises if the completer replies with more than 8 completions per MRd request, PeCORE issues 4 MRd request, and no header credit yet is returned for the first completions. This scenario will cause completion header credits to overflow, triggering the reporting of Receiver Overflow error. FW can get around this problem by calibrating PeDRF_CPMR to a value that is equal or more than the possible maximum number of completions the completer can issue per MRd request.

25.2.4.2.9 Malformed TLP

This error is reported for the following types of error:

1. TLP Payload Size greater than the MaximumPayloadSize (PeDRF_PEDSCTL[7:5]) setting

2. Invalid FmType

3. Cfg or IO Request Error - Request is a Cfg or IO request and, at least, one of the following errors is detected:

    Length != 1

    Traffic Class (TC) != 0

    Attribute (Attr) != 0

    Last Byte Enable (LBE) != 0

4. Byte Enable Error - at least, one of the following errors is detected:

    Length != 1 and First Byte Enable (FBE) = 0

    Length = 1 and Last Byte Enable (LBE) != 0

    Length != 1 and Last Byte Enable (LBE) = 0

    Length > 2 and LBE/FBE is non-contiguous

    Length = 2, address is not quadword-aligned and LBE/FBE is non-contiguous.

5. INTx Error - PeCORE is in Endpoint mode and receives an Assert_INTx/Deassert_INTx message from downstream port.

6. 4-KB Boundary Error - Memory request address crosses the 4KB-boundary.

7. Traffic Class (TC) Error - Traffic Class is non-zero.

8. Address Type (AT) Error - Packet is not a memory request and AT is non-zero

9. TLP Length error - Length does not match the actual payload size.

Malformed TLPs are discarded by PeCORE. How they are handled in relation to other packets is discussed below.

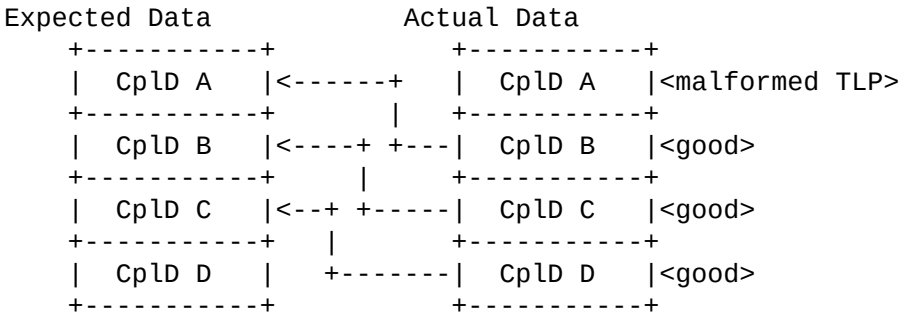
If packet is a MRd Completion

ICBM-DMA

If multiple completions and first completion is a malformed TLP

1. First completion is discarded. Payload data is not forwarded to ICBM
2. Payload data of next completions are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.25

Figure 25.25: First Completion is Malformed

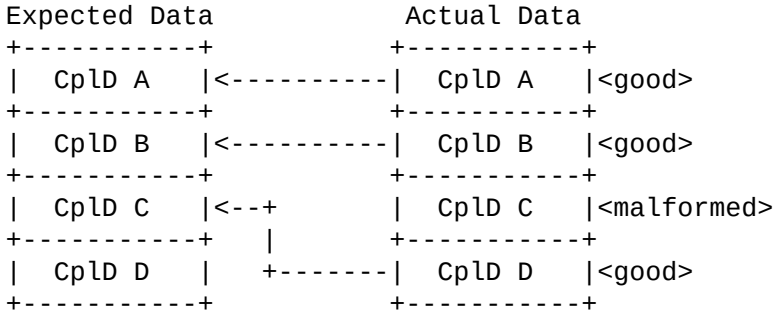


3. Since the completion for MRd request is not completed, the request will time out.
4. PeCORE reports Completion Timeout event through IG update. Since, as illustrated in figure 1, data passed to ICBM are skewed and that IG rewind pointer erroneously points to the start of CplD D, it is recommended that data is retried from the first start address of the IG.

If multiple completions and first completions are not malformed

1. Payload data of first completions are forwarded to ICBM.
2. Malformed Completion is discarded. Payload data is not forwarded to ICBM
3. Payload data of next completions following the malformed one are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.26

Figure 25.26: First Completions are not Malformed



4. Since the completion for MRd request is not completed, the request will time out.
5. PeCORE reports Completion Timeout event through IG update. Since, as illustrated in figure 2, data passed to ICBM are skewed and that IG rewind pointer erroneously points to the start of CplD D, it is recommended that data is retried from the first start address of the IG.

If single completion

1. Completion data is discarded
2. MRd request completion timer expires
3. PeCORE reports Completion Timeout event through IG update

CSR-DMA

If multiple completions and first completion is malformed

1. First completion is discarded. Payload data is not forwarded to ICBM
2. Payload data of next completions are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.25
3. Since the completion for MRd request is not completed, the request will time out.
4. MRd request is automatically retried by the hardware. Request is retried from the start.

If multiple completions and first completions are not malformed

1. Payload data of first completions are forwarded to ICBM.
2. Malformed Completion is discarded. Payload data is not forwarded to ICBM
3. Payload data of next completions following the malformed one are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.26
4. Since the completion for MRd request is not completed, the request will time out.
5. MRd request is automatically retried by the hardware. Request is retried from the start.

If single completion

1. Completion is discarded
2. MRd request completion timer expires
3. MRd request is automatically retried by the hardware

If packet is an IO/Cfg Completion

1. Completion is discarded
2. IO/Cfg request completion timer expires
3. IO/Cfg request is automatically retried by the hardware

If packet is a Request

1. Request is discarded

25.2.4.2.10ECRC Error

Since ECRC error could mean that the TLP header is corrupted, there is no certainty that the packet with ECRC error can be processed correctly. To avoid incorrectly treating, for example, a MRd TLP as MWr TLP or something else, or mapping a completion to the wrong request, packet with ECRC error is discarded by PeCORE.

If a MRd with multiple completions, for example, has a completion with ECRC error sandwiched between good completions, PeCORE cannot discard completions after the completion with ECRC error, because PeCORE, in the first place, cannot ascertain that the detected completion with ecrc error belongs to a certain request.

How a packet received with ECRC error is handled in relation to other packets is discussed in this section.

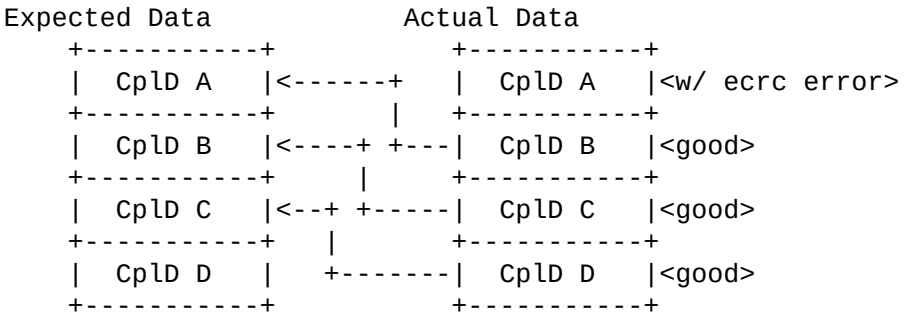
If packet is a MRd Completion

ICBM-DMA

If multiple completions and first completion has ECRC error

1. First completion is discarded. Payload data is not forwarded to ICBM
2. Payload data of next completions are forwarded to ICBM. Data are passed to ICBM.Rx-Buffer as illustrated in Figure 25.27

Figure 25.27: First Completion has ECRC Error

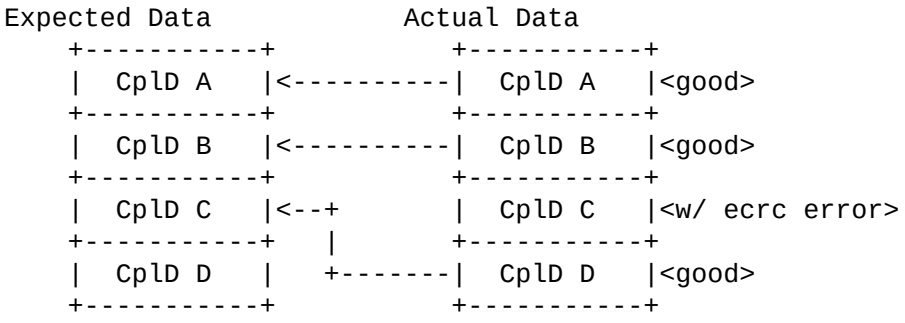


3. Since the completion for MRd request is not completed, the request will time out.
4. PeCORE reports Completion Timeout event through IG update. Since, as illustrated in Figure 25.27, data passed to ICBM are skewed and that IG rewind pointer erroneously points to the start of CplD D, it is recommended that data is retried from the first start address of the IG.

If multiple completions and first completions don't have ECRC error

1. Payload data of first completions are forwarded to ICBM.
2. Completion with ECRC error is discarded. Payload data is not forwarded to ICBM
3. Payload data of next completions that have no ecrc error are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.28
4. Since the completion for MRd request is not completed, the request will time out.
5. PeCORE reports Completion Timeout event through IG update. Since, as illustrated in figure 2, data passed to ICBM are skewed and that IG rewind pointer erroneously points to the start of CplD D, it is recommended that data is retried from the first start address of the IG.

Figure 25.28: First Completions don't have ECRC error



If single completion

1. Completion data is discarded
2. MRd request completion timer expires
3. PeCORE reports Completion Timeout event through IG update

CSR-DMA

If multiple completions and first completion has ECRC error

1. First completion is discarded. Payload data is not forwarded to ICBM
2. Payload data of next completions are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.27
3. Since the completion for MRd request is not completed, the request will time out
4. MRd request is automatically retried by the hardware. Request is retried from the start.

If multiple completions and first completions don't have ecrc error

1. Payload data of first completions are forwarded to ICBM.
2. Completion with ecrc error is discarded. Payload data is not forwarded to ICBM
3. Payload data of next completions that have no ecrc error are forwarded to ICBM. Data are passed to data buffer as illustrated in Figure 25.28
4. Since the completion for MRd request is not completed, the request will time out.
5. MRd request is automatically retried by the hardware. Request is retried from the start.

If single completion

1. Completion is discarded
2. MRd request completion timer expires
3. MRd request is automatically retried by the hardware

If packet is an IO/Cfg Completion

1. Completion is discarded
2. IO/Cfg request completion timer expires
3. IO/Cfg request is automatically retried by the hardware

If packet is a Request

1. Request is discarded

25.2.4.2.11TLP Nullification

25.2.4.2.11.1 TX

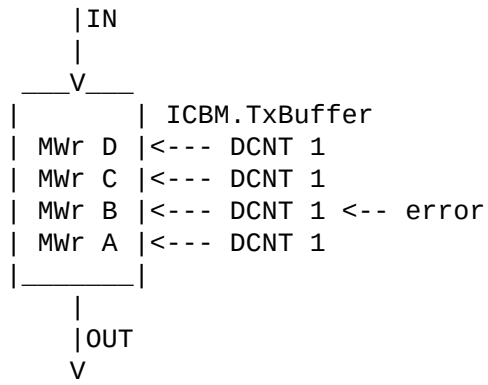
Packet is nullified when one of the following happens:

1) Parity error is detected during ICBM.TxBuffer-to-IOC data transmission. This is indicated by the signal TxDat_ParErr.

2) Parity error is detected during Memory-to-ICBM.TxBuffer data transmission (indicated by the signal TxDat_TagInfo[3]), but the TLP header for the data with parity error is already transmitted.

MWr Data

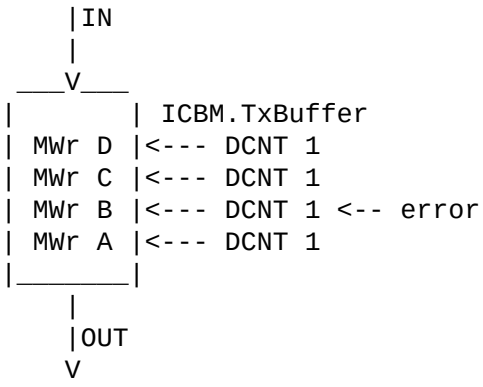
Figure 25.29: Nullification of MWr Data



1. MWr A already sent when error is detected in MWr B
2. MWr B is nullified. Automatic HW retry is initiated (TxDat_Status=RETRY)
3. MWr C and MWr D are not sent
4. When retried, entire DCNT 1 is retried starting from MWr A. Retried data may not be the same MWr TLPs as the original TLPs (MWr A/B/C/D). Entire DCNT, for example, may be retried as 1 MWr TLP if the entire data is available in TxBuffer when retry is started.

Completion Data

Figure 25.30: Nullification of Completion Data



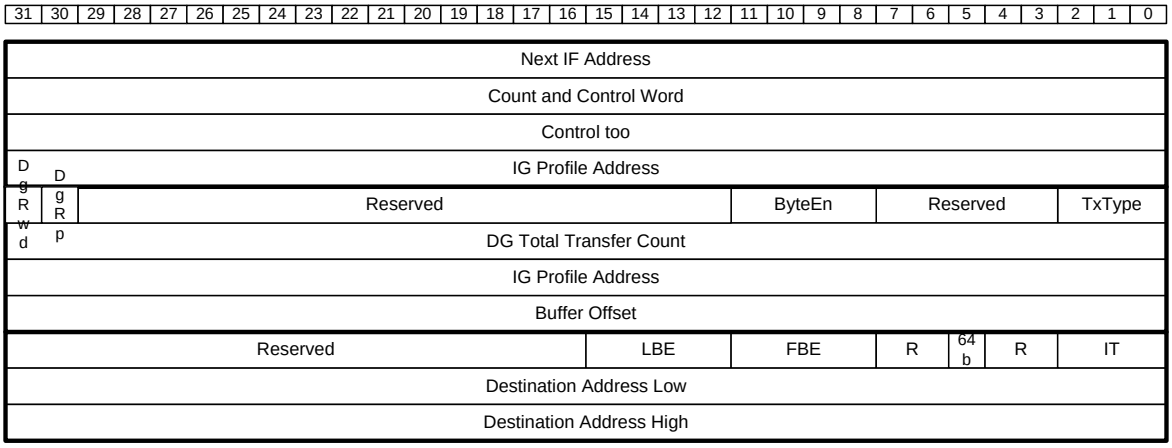
1. MWr A already sent when error is detected in MWr B
2. MWr B is nullified. No HW retry is initiated. Because the completion returned by PeCORE is not complete, the non-posted request sent by the requester will time out. The requester has the option to retry the request or not.
3. MWr C and MWr D are not sent

25.2.4.2.11.2 RX

Nullified TLPs received by PeCORE are handled automatically by the hardware at the Link Layer without FW intervention.

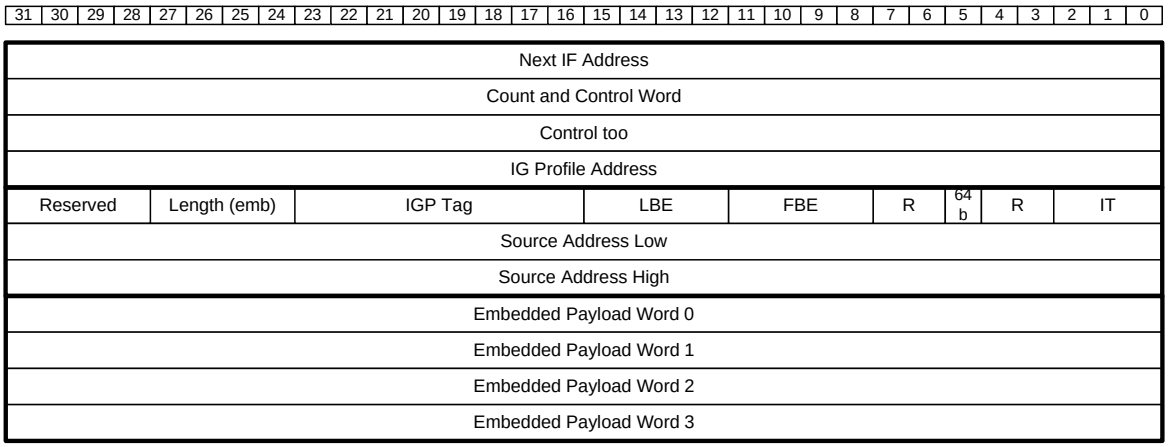
25.2.5 *Instruction Frames*

Figure 25.31: MWr-Req Instruction Frame



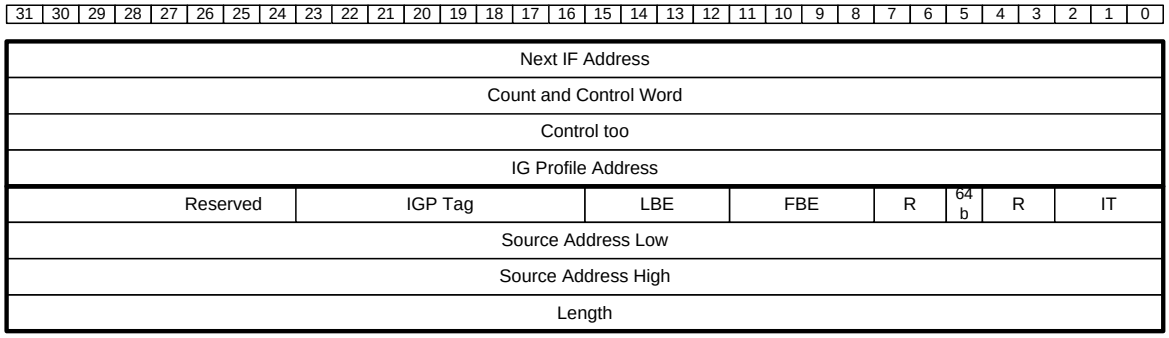
MWr-Req instruction frame is used when Talino™ EDC wants to initiate MWr request. The first 4 words is the Instruction Frame Header, while the next 4 words is the DCNT. Instruction Frame Header and DCNT are discussed in detail in ICBM section of this document. The last 3 words are for PeCORE use.

Figure 25.32: MWr-Req With Embedded Payload Instruction Frame



MWr-Req with embedded payload instruction frame is used when Talino™ EDC wants to initiate MWr request and the data to be transmitted are embedded in the Instruction Frame, not transmitted from ICBM.Tx Buffer. The instruction frame can contain up to 4 words of data payload. The first 4 words of the instruction frame is the Instruction Frame Header. The next 3 words are for PeCORE use, and the remaining words are the embedded data payload.

Figure 25.33: MRd-Req Instruction Frame



MRd-Req instruction frame is used when Talino™ EDC wants to initiate MRd request. The first 4 words is the Instruction Frame Header, while the last 4 words are for PeCORE use.

Table 25.10: IT (Instruction Type)

Value	IT	
3'b000	MWr-Req	Used when Talino™ EDC wants to initiate MWr Request.
3'b001	MRd-Req	Used when Talino™ EDC wants to initiate MRd Request.
3'b100	MWr-Req (emb)	Used when Talino™ EDC wants to initiate MWr Request with embedded payload.
others	reserved	

Table 25.11: 64b Addressing Mode

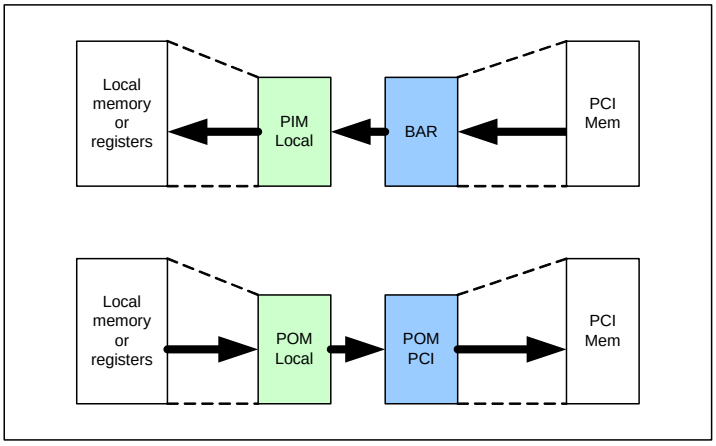
Value	64b
1'b0	32bit TLP addressing mode
1'b1	64bit TLP addressing mode

Table 25.12: Other Fields

Field	Comment
FBE	First Word Byte Enables
LBE	Last Word Byte Enables
IGP Tag	Tag assigned to IG
Length (emb)	Number of words embedded in the Instruction Frame
Address Low	The lower 32bits of destination (during MWr) or source (during MRd) address
Address High	The upper 32bits of destination (during MWr) or source (during MRd) address

25.2.6 Address Translation

Figure 25.34: Address Translation



25.2.6.1 Inbound Transaction

In Endpoint mode, before PeCORE owns an incoming Memory or IO Request transaction, that transaction should be within the range of any of the enabled BARs. Once owned by PeCORE, the address of that transaction is translated to the equivalent PIM Local address. While BAR defines PCI (system) Memory assigned by Root Complex to Talino™ EDC, PIM Local can point either to a local RAM or register. Reception, decoding and address translation of incoming Memory/IO Request is enabled through *PIM Enable*. Table 25.13 shows the mapping of BARs to corresponding PIM Local registers in Endpoint mode.

Table 25.13: BAR-to-PIMLocal Mapping in Endpoint mode

Transaction Address	Translated to
BAR 0	PIM Local 0
BAR 1 (upper 32 bits of BAR 0 if 64-bit mode)	PIM Local 1 (upper 32 bits of PIM Local 0 if 64-bit mode)
BAR 2	PIM Local 2
BAR 3 (upper 32 bits of BAR 2 if 64-bit mode)	PIM Local 3 (upper 32 bits of PIM Local 2 if 64-bit mode)
BAR 4	PIM Local 4
BAR 5 (upper 32 bits of BAR 4 if 64-bit mode)	PIM Local 5 (upper 32 bits of PIM Local 4 if 64-bit mode)

Sample inbound 32-bit address translation in Endpoint mode:

*Transaction address = **0x01234567***

BAR0 = 0x01230000

PIM Local 0 = 0x89ab0000

PIM Size 0 = 0x00100001 (since bit[20]=1, size=1MB)

Base Address = [31:20]

Transaction Address[31:20] = BAR0[31:20]?

0x012 = 0x012 (address match)

Translated Address = {PIM Local[31:20], Transaction Address[19:0]}

= {0x89a,0x34567} = ***0x89a34567***

Sample inbound 64-bit address translation in Endpoint mode:

Transaction address = ***0xaaaaaaaa_01234567***

BAR 0 = 0x01230000

BAR 1= 0xaaaaaaaa

PIM Local 0 = 0x89ab0000

PIM Local 1 = 0xbbbbbbbb

PIM Size 0 = 0x00100001 (since bit[20]=1, size=1MB)

Base Address = [63:20]

Transaction Address[63:20] = {BAR1[31:0],BAR0[31:20]}?

0xaaaaaaaa_012 = 0xaaaaaaaa_012 (address match)

Translated Address = {PIM Local 1[31:0], PIM Local 0[31:20], Transaction Address[19:0]}

= {0xbbbbbbbb_89a,0x34567} = ***0xbbbbbbbb_89a34567***

In Root Complex mode, all incoming Memory and IO request transactions, whether or not within the range of the enabled BARs, are owned by PeCORE. If a transaction falls within the range of an enabled BAR, it is interpreted that the destination of the request is the internal CSRs of PeCORE. PIM Local address, however, still has to match the local address of PeCORE's internal CSR. All transactions outside BARs are forwarded to AWB buses without address translation. A safeguard, however, is implemented such that no unwanted transactions reaches AWB. Table 25.14 shows the mapping of incoming transactions to corresponding BARs.

Table 25.14: Inbound Transactions Mapping in Root Complex Mode

Transaction Address	Transaction Destination
BAR 0	Internal CSR

BAR 1	Internal CSR
Outside BARs	forwarded to AWB

Sample inbound 32-bit address translation in Root Complex mode:

*Transaction address = **0x01234567***

BAR0 = 0x01230000

PIM Local 0 = 0xfff20000 (PeCORE's internal CSR base address)

PIM Size 0 = 0x00010001 (since bit[16]=1, size=64KB)

Base Address = [31:16]

Transaction Address[31:16] = BAR0[31:16]?

0x0123 = 0x0123 (address match)

Translated Address = {PIM Local[31:16], Transaction Address[15:0]}

*= {0xfff2,0x4567} = **0xfff24567***

25.2.6.2 Outbound Transaction

While inbound Memory/IO transactions always have to go through BAR-to-PIM Local address translation, outbound transactions may skip POMLocal-to-POM PCI address translation. In Endpoint mode, any local access to local address that falls within any of the POM Local registers triggers generation and transmission of Memory/IO Request TLP. If *POM Address Translation Enable* is 1, local address is translated to equivalent PCI address. If *POM Address Translation Enable* is 0, Memory/IO Request transactions are transmitted without address translation.

Note that only direct memory/IO requests are affected by POM registers. Indirect and Descriptor-based requests are not affected by POM registers. TLP address in these modes are transmitted unchanged. Transaction in these modes can be initiated virtually to any address. It does not need to be within the range of POM Local registers. Table 25.15 shows the POM Local-to-POM PCI mapping in Endpoint mode.

Address ranges to be assigned to POM Local registers should be from Talino™ EDC's PCI Memory 0-7 or PCI IO 0-1.

Table 25.15: POMLocal-to-POMPCI Mapping in Endpoint mode

Transaction Address	Translated to
POM Local 0	POM PCI 0
POM Local 1 (upper 32 bits of POM Local 0 if 64-bit mode)	POM PCI 1 (upper 32 bits of POM PCI 0 if 64-bit mode)
POM Local 2	POM PCI 2
POM Local 3 (upper 32 bits of POM Local 2 if 64-bit mode)	POM PCI 3 (upper 32 bits of POM PCI 2 if 64-bit mode)

POM Local 4	POM PCI 4
POM Local 5 (upper 32 bits of POM Local 4 if 64-bit mode)	POM PCI 5 (upper 32 bits of POM PCI 4 if 64-bit mode)

Sample outbound 32-bit address translation in Endpoint mode:

*Transaction Local Address = **0xc0034567***

POM Local 0 = 0xc0000000

POM PCI 0 = 0x89a00000

POM Size 0 = 0x00100003 (since bit[20]=1, size=1MB)

Base Address = [31:20]

Transaction Local Address[31:20] = POM Local 0[31:20]?

0xc00 = 0xc00 (address match)

Translated Address = {POM PCI 0[31:20], Transaction Local Address[19:0]}

*= {0x89a,0x34567} = **0x89a34567***

In Root Complex mode, IO and Memory transaction each has its own dedicated (local) Base registers. Outbound IO transaction local address must fall within the range of IO Base/Limit registers before an IO transaction is initiated. If *POM Address Translation Enable* is 1, local address is translated to equivalent POM PCI address. Outbound Memory transaction local address, on the other hand, must fall within the range of either the Memory Base/Limit registers or Prefetchable Base/Limit registers. If *POM Address Translation Enable* is 1, local address is translated to equivalent POM PCI address. Table 25.16 shows the mapping of outbound transactions in Root Complex mode.

Address ranges to be assigned to IO Base/Limit registers should be from Talino™ EDC's PCI IO 0-1, while address to be assigned to Memory Base/Limit registers and Prefetchable (Upper) Base/Limit registers must be from Talino™ EDC's PCI Memory 0-7.

Table 25.16: Outbound Transactions mapping in Root Complex mode

Transaction Address	Translated to
32-bit IO Base Register	POM PCI 0
Memory Base Register	POM PCI 1
Prefetchable Base Register	POM PCI 2
Prefetchable Upper Base Register	POM PCI 3
-	POM PCI 4 (unused)
-	POM PCI 5 (unused)

Sample Outbound 32-bit IO address translation in Root Complex mode:

*Transaction Local Address = **0xd0004567***

IO Base = 0xd0000000

IO Limit = 0xd0010000

0xd0004567 within the IO Base and Limit range? Yes

Base Address = [31:16]

POM PCI 0 = 0x89a00000

Translated Address = {POM PCI 0[31:16], Transaction Local Address[15:0]}

*= {0x89a0,0x4567} = **0x89a00567***

Sample Outbound 32-bit Memory address translation in Root Complex mode:

*Transaction Local Address = **0xc0034567***

Memory Base = 0xc0000000

Memory Limit = 0xc0100000

0xc0004567 within the IO Base and Limit range? Yes

Base Address = [31:20]

POM PCI 0 = 0x89a00000

Translated Address = {POM PCI 0[31:20], Transaction Local Address[19:0]}

*= {0x89a,0x34567} = **0x89a34567***

25.2.7 Link Training and Status State Machine

25.2.7.1 Overview

Link initialization and training is a Physical Layer control process that configures and initializes each Link before a normal packet can be transferred through the Lanes. Aside from configuring and initializing the Link and supporting normal packet transfer, this process also supports state transitions when recovering from Link errors (Link Retraining) and restarting a port from low power states (Wake-Up). After reset, this process is initiated automatically without any intervention from software/firmware. The Link Training and Status State Machine (LTSSM) is the block responsible for this control process.

Basically, LTSSM will determine the Link Width, Link Data Rate, Lane Reversal, and Polarity Inversion.

Link Width refers to the negotiated number of Lanes that can be supported by the two devices connected to the Link. A device may support multiple Links but for PeCORE, it can only support single Link. This can be determined through Link Width Negotiation.

Link Data Rate, on the other hand, is the negotiated speed of the Link. For PeCORE, PCIE Gen2 or a maximum of 5.0 GT/s data rate is supported. This means that it can support both 2.5 GT/s and 5.0 GT/s transfer rates. Similar to Link Width, a process called Link Speed Negotiation determines the Link's data rate of operation.

Ideally, for a multi-lane device, Lane 0 should be connected to Lane 0 of the other device, Lane 1 to Lane 1, and so on. During the Lane ordering process, if Lanes are connected in reverse, there is a possibility that the wires will introduce interference into the Link. Hence, Lane Reversal may be supported by the device. However, in PeCORE, Lane Reversal is not supported. Hardware implementation must ensure that there will be no crisscrossing of wires.

In PeCORE, Polarity Inversion is supported. This feature logically reverses the differential pair terminals of the device if it was found out that they were inversely connected. After the receiver detection process, LTSSM will inform the PHY interface through the RxPolarity port that it must invert the polarity of the receiver lanes.

Aside from those mentioned above, LTSSM also performs Lane to Lane de-skew within a multi-lane link.

Due to Link wire length variations and the different driver/receiver charactesitics on a multi-lane link, each of the parallel bit streams that represent a packet are transmitted simultaneously, but they do not arrive at the receiver on each lane at the same time. The LTSSM block compensate for this skew by adding delays on each lane so that the receiver can receive and align the serial bit streams of the packet. For PeCORE, a maximum of 6-symbol skew can be de-skewed. Above that, invalid ordered sets and malformed TLP's may result.

25.2.7.2 Link Speed Negotiation

All devices are required to start Link initialization using 2.5 GT/s data rate on each lane. This means that for devices that supports multiple data rates, the Link will train initially in 2.5 GT/s data rate after which data rate change occurs by going through the Recovery State. A normal interrupt (CINT_LNKL0INT = 1) will be issued upon transitioning to L0 (LTSR_LTS[3:0] = 4'b1100). Initially

during that state (L0), it is understood that only the Root Complex may initiate data rate change by writing 1 to PELKSCO_PERETLK. After the initial speed change, PeCORE, whether set as a Root Complex or as an Endpoint, it may initiate another speed change by writing 1 to PELKSCO_PERETLK (Root Complex) or PELTCTL_GOREC (Endpoint).

Before initiating speed change again, during L0, the FW must set the desired speed in PELKSCO2_TRGLNKSP. Consequently, the FW must ensure that the desired speed is within the limit of maximum link speed setting in PELKCAP_MXLKS. The FW may also check in LASCC_EDSLS whether the other device connected to the link supports the desired speed setting.

Successful negotiation will then follow this flow L0 -> Recovery -> L0. The negotiated speed will be reflected in PELKSCO_PELKSPE. The desired speed may not be achieved but it is still considered a successful negotiation. One case will be the failure of achieving Symbol Lock. In any case, both devices will revert back to the speed they operated prior to entering Recovery State.

25.2.7.3 Link Width Negotiation

Similar to Link Speed Negotiation, all devices are required to initially negotiate Link Width during link initialization. Initially, after reset, PeCORE will try to negotiate the link width specified in PELKCAP_PEMXLWD(Max Link Width). PeCORE will follow this state transition: Detect -> Polling -> Configuration -> L0. At L0 state, the negotiated Link Width will be reflected in PeLKSCO_PENLKWI. Only at this state, PeCORE may opt to downsize or upconfigure the negotiated link width. The steps on how to downsize/upconfigure are as follows:

1. Before initiating the change, FW would make sure that the Hardware Autonomous Width Disable bit was set to 0 and the Link Upconfigure Capability bit was set to 1 before the entry to Configuration State. If the latter bit was not set to 1, FW must set that bit to 1 and PeCORE must be retrain again before it can initiate to change link width by writing 1 to PELKSCO_PERETLK (Root Complex) or PELTCTL_GOREC (Endpoint). This will inform the other device connected to the link that PeCORE supports link width change. If one of the device does not supports link width change, both devices cannot initiate to downsize/upconfigure the link width.
2. FW must set the desired link width by setting LASCC_UCLW. This will be the the target link width for the change. For upconfiguration, the desired link width must not be greater than the link width after the initial link training.Furthermore, FW must also set LASCC_UCLWI to 1. This will signify that PeCORE is the initiator of the change.
3. At L0, we can start initiating link width change by directing PeCORE to Recovery. If Root Complex, set retrain link bit (PELKSCO_PERETLK) to 1. If Endpoint, set LTCTL_GOREC to 1.
4. Succesful negotiation will follow this flow L0 -> Recovery -> Configuration - L0. PeCORE will then use LASCC_UCLWI = 1 to transition from Recovery to Configuration. At L0, the negotiated link width as a result of the change will be reflected in PeLKSCO_PENLKWI.

25.2.8 Power Management

PeCORE is compliant with the PCI Express Power Management (PCI Express-PM) capabilities and protocols specified in the PCI Express™ Base Specification Revision 2.1. However, do take note that a comprehensive discussion on PCI Express Power Management is not included in this section. Hence, it is highly advisable for the reader to refer to the Power Management chapter (Chapter 5) of the PCI Express™ Base Specification. The discussions included in this section are the power management details referred to by the PCI Express™ Base Specification as implementation specific and PeCORE's role in the overall EDC power scheme.

25.2.8.1 Link and Device Power States

For link power states, PeCORE, in both Root Complex and Endpoint modes, supports the Active State Power Management States - L0s ASPM and L1 ASPM. This functionality is enabled by setting the PeDRF_PEASPMC register field.

When L0s functionality is enabled, the transmit side of the link transitions to L0s if there is no detected TLP/DLLP transmission attempt for some time indicated in PeDRF_LNKIDLTML0S. When both L0s and L1 functionality are enabled and both directions are already in L0s, the link transitions to L1 if there is no detected TLP/DLLP transmission attempt for some time indicated in PeDRF_LNKIDLTML1. However if L1 functionality is enabled but L0s functionality is disabled, the link would transition to L1 if there is no detected TLP/DLLP transmission attempt during L0 for some time indicated in PeDRF_LNKIDLTML1.

For device power states, PeCORE supports the optional D1 and D2 Power Management States by default. However, these can be disabled by deasserting PeDRF_PMCPHDR_PMD1SUP and PeDRF_PMCPHDR_PMD2SUP for D1 and D2 support respectively.

25.2.8.2 EDC Power Management

Aside from what was described in the Power Management chapter of the PCI Express™ Base Specification Revision 2.1, PeCORE Power Management is also integrated in the EDC's overall power management strategy. To improve power usage efficiency, EDC is grouped into different power domains that could be powered down when not in use. A power domain (or Clock and Power Management Unit) is defined as a logical grouping of EDC clock, reset, and power supply controls for power management.

PeCORE is assigned within one of these Clock and Power Management Units (CPMU) - CPMU3 (refer to Figure 25.35). Included in CPMU3 are the ff cores: PePHY, PeCORE, IOB_PE, ICBM_PE, IdCORE(0-3)_3aPe, CcCORE_PE. A Power Management Master (PMMstr) module is included in PeCORE in order to effectively manage CPMU3's power consumption. The PMMstr checks the activity, Lc chip select, and interrupt lines of all the cores within CPMU3 and decides which clocks within CPMU3 would be turned off. Note that clock stopping via PMMstr is hardware initiated.

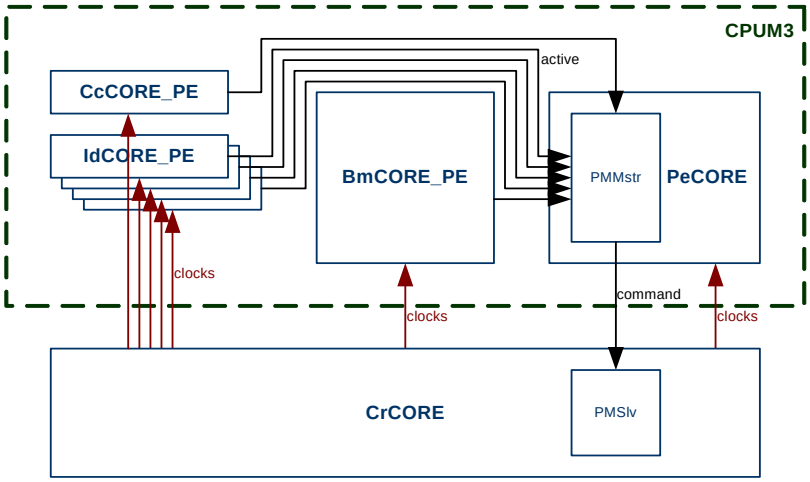
In CPMU3, there are two types of clock stopping via PMMstr:

1. Full clock stopping - If all cores within CPMU3 are considered IDLE for a set amount of time (defined in PeDRF_INACTCTR) while the link is in L1/L2, then the PMMstr will request for clock stoppage to all stoppable clocks in CPMU3.

2. Partial clock stopping - If all cores within CPMU3 except PeCORE are considered IDLE for a set amount of time (defined in PeDRF_INACTCTR) while the link is in L0/L0s, then the PMMstr will request for clock stoppage to all stoppable clocks not used in PeCORE. Note that PeDRF_PRTLCLKSTPEN register bit must be asserted in order for this type of clock stopping to be enabled. Also note that usage of partial clock stopping feature is advisable only if FW is sure that no MWr would be invoked by the other device.

The PMMstr's clock stop request would be sent to CrCORE. Then CrCORE would turn off the required clocks upon detection of PMMstr's request via its Power Management Slave (PMSlv) module. This process leads to the improved power usage of the EDC during IDLE states.

Figure 25.35: CPMU3



PeCORE is considered IDLE when there are no ongoing and pending transactions on PeCORE's Tx path, Rx path, and CSR control logic. As a whole, CPMU3 would be considered IDLE when no transactions are ongoing amongst its internal cores and to cores and devices adjacent to it (eg. AWBs, PCIe device at the other side of the link). Otherwise, it would be considered ACTIVE.

Note that PeCORE's power management module has a separate clock source which is non-stop-pable (via hardware). This would allow PeCORE to inform (via ACTIVE status signal assertion) the PmBUS in case it needs to power up. For instance, PeCORE needs to power up when the device at the other end of the lane tries to communicate with it.

For a more comprehensive discussion on CPMUs; see Clock, Reset, and Power Management Chapter (Chapter 15).

25.2.8.3 Link Power Management Processes

Link Power Down Process

When PeCORE is in Root Complex mode, FW can choose to turn off the Endpoint device(s) in the hierarchy. The process proceeds as follows:

1. Set PeDRF_HPO to '1'. This initiates the link power down handshake between the Root Complex and Endpoint.
2. PME_Turn_Off Message would be sent to all Endpoint devices in the hierarchy.
3. The Endpoint devices will reply back with PME_TO_Ack, and this would be received by

PeCORE.

4. PM_Enter_L23 would then be sent by the Endpoint devices when power can already be removed.
5. PM_Request_Ack would then be replied back by PeCORE.
6. After which, PeDRF_HPO register would be then also be reset to '0'. This indicates that the powerdown handshake is done.

Link Wake Up Process

When PeCORE is in Endpoint mode and the link is powered down, FW can choose to power up the link. The process proceeds as follows:

1. Set PeDRF_LNKWAKEUP register to '1'. This initiates the link wake up process between the Root Complex and Endpoint.
2. PeCORE will request for link wake up by sending both beacon and WAKE# signals to the Root Complex device. (For a more in depth discussion of beacon and WAKE# signals, do check the latest PCI Express™ Base Specification.)
3. The Root Complex would then detect the presence of the wake up request. It would then acknowledge the wake up request by deasserting electrical idle.
4. Both devices would then go to LTSSM Detect state and after some initializations and configurations would eventually transition to L0 state.

25.2.8.4 PeCORE Clocks and Resets

This section discusses the clocks and resets used within PeCORE and its Physical Layer.

The PeCORE operates mainly from the clock which is generated in the Physical Layer and is fed to CrCORE: 125MHz for gen1, 250MHz for gen2. Switching between the two frequencies is supervised by the LTSSM. A PCIe device will always start at the lower frequency. Once a Link is successfully established, the device may attempt to operate at the higher frequency. The clock from the Physical Layer may be stopped if the device is in sleep mode. Hence, a 333MHz non-stoppable clock, not originating from the Physical Layer, is needed in order to maintain operation of some critical registers and functions that must be functioning even when the main PeCORE clock is down.

The PCI Express™ Base Specification defines a number of Conventional (Fundamental) Reset mechanisms for setting or returning all Port states to the initial conditions:

- Cold Reset is a Fundamental Reset following the application of power to the device.
- Warm Reset is a Fundamental Reset triggered by hardware without removal or reapplication of power to the device.
- Hot Reset is a Conventional Reset propagated across a Link (in-band reset signal; refer to Section 4.2.4.7 of the PCI Express™ Base Specification).

Rules applied for PCIe device reset:

- On reset, all Port registers and state machines are set to their initial values, except the sticky registers (refer to sections 7.4 and 7.6 of the PCIe specification).
- The PCIe component must enter LTSSM Detect state within 20ms after Fundamental Reset

- System Software must wait for at least 100ms after a Conventional Reset for component internal initialization. All components intended to be visible to Software at boot time must be ready to receive Configuration Requests within 100ms.
- Root Complex and System Software must allow 1.0s minimum time after Conventional Reset before it determines that a device is broken because the device fails to return a Successful Completion status for a valid Configuration Request.

25.2.8.4.1 Clock and Reset Sequence

From the block diagram at Figure 25.36, the following tables (Table 25.17 and Table 25.18) list summaries of the clock and reset signals to and from CrCORE:

Figure 25.36: Clock and Reset Block Diagram

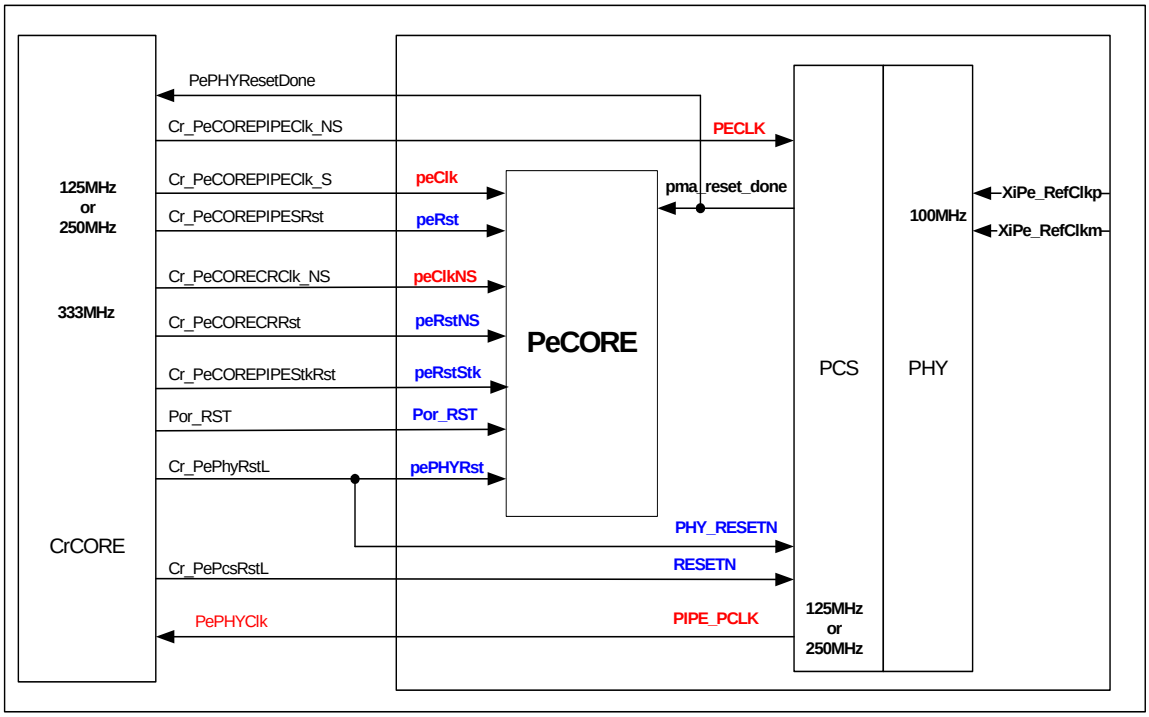


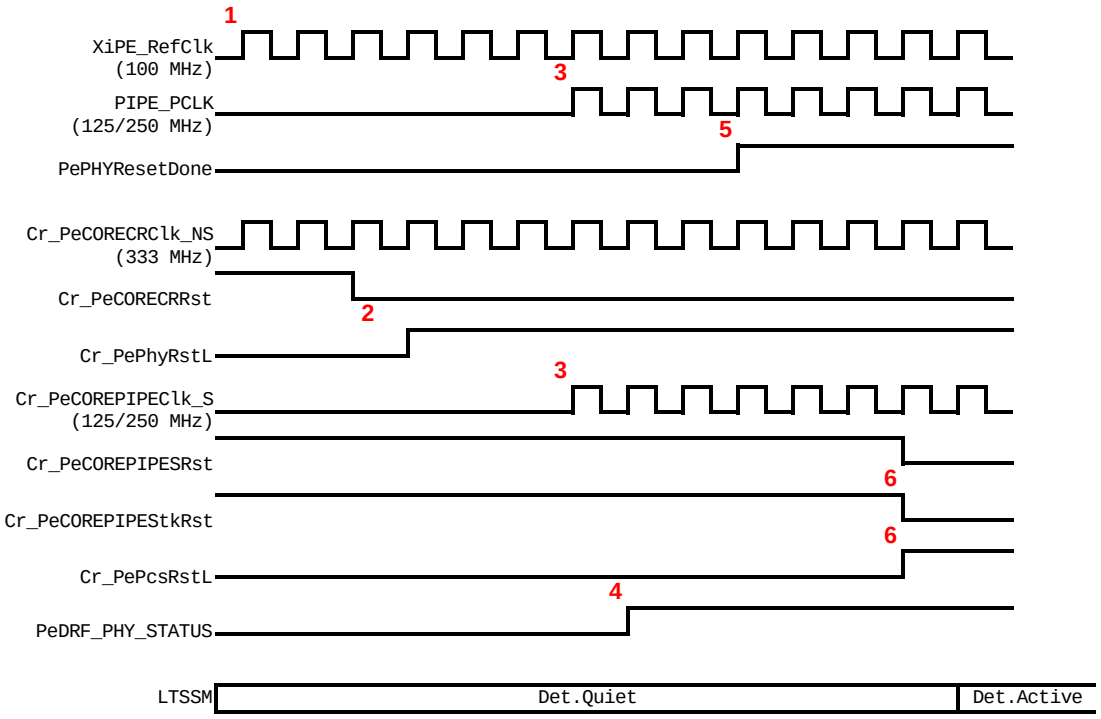
Table 25.17: Clock Signals

Clock Name	Frequency	Description
Cr_PeCORECRClk_NS	333 MHz	Non-stoppable clock; continuously running even if the Physical Layer stops the PLL clock during sleep mode
Cr_PeCOREPIPEClk_S	125/250 MHz	Stoppable clock derived from the Physical Layer PLL clock. (This clock may be stopped by shutting down the Physical Layer itself or its reference clock)
PePHYClk	125/250 MHz	Physical Layer PLL clock from PHY.

Table 25.18: Reset Signals

Reset Name	Sync Clock	Description
Cr_PeCOREPIPESRst	Cr_PeCOREPIPEClk_S (125/250 MHz)	Active High. This is the main PeCORE reset and should be de-asserted at least 16 clock cycles after PePHYResetDone goes high which indicates that the IP PHY has completed its reset sequence. This reset is controlled by bit[5] in PMU3[1]_PE register of CrCORE.
Cr_PeCORECRRst	Cr_PeCORECRClk_NS (333 MHz)	Active High. This resets the all the registers which runs at 333 MHz non-stoppable clock. Early deassertion of this reset allows software access to specific control registers even before the IP PHY has started generating the main PeCORE clock. This reset is controlled by bit[2] in PMU3[1]_PE register of CrCORE.
Cr_PeCOREPIPEStkRst	Cr_PeCOREPIPEClk_S (125/250 MHz)	Active High. This resets the sticky registers. During Cold and Warm Reset, all sticky registers except those supported by Vaux will be reset to default values. During Hot Reset, all sticky registers will not be modified. This reset is controlled by bit[7] in PMU3[1]_PE register of CrCORE.
Cr_RawPORst		Active High. Power on Reset
Cr_PePhyRstL	Cr_PeCORECRClk_NS (333 MHz)	Active Low. This is required by the IP PHY and should be deasserted first in order to activate the Physical Layer logic circuits and PLL that would eventually provide the main PeCORE clock. This reset is controlled by bit[4] in PMU3[1]_PE register of CrCORE.
Cr_PePcsRstL	Cr_PeCOREPIPEClk_S (125/250 MHz)	Active Low. This is required by the IP PHY and should be de-asserted at least 16 clock cycles after PePHYResetDone goes high which indicates that the IP PHY has completed its reset sequence. This reset is controlled by bit[3] in PMU3[1]_PE register of CrCORE.

Figure 25.37: Power-On Reset Sequence Timing Diagram



To reset PeCORE during power-on all its reset signals must be asserted. The power-on reset sequence shown in Figure 25.28 is described in more detail in the following statements:

1. The reference clock input to the IP PHY (EDC PCIe reference clock input pins, **XiPE_RefClkm** and **XiPE_RefClkp**) must become stable first. The 333-MHz non-stoppable clock **Cr_PeCORECRClk_NS** is also active and stable.
2. FW should deassert **Cr_PeCORECRRst** (controlled by bit[2] in PMU3[1]_PE register of CrCORE). PeCORE control and status registers with address offset 0x8xxx are now accessible. FW will also activate the IP PHY by deasserting **Cr_PePhyRstL** (controlled by bit[4] in PMU3[1]_PE register of CrCORE). FW may opt to set first the registers that need to be configured before activating the IP PHY.
3. The IP PHY PLL becomes active and generates the 125/250-MHz PeCORE clock. This will be used by CrCORE to create **Cr_PeCOREPIPEClk_S**.
4. After detecting a stable clock **Cr_PeCOREPIPEClk_S** and **Cr_PePhyRstL** was deasserted**,** PeCORE then updates the PHY status register in PeDRF_PHY_STATUS.
5. The IP PHY asserts **PePHYResetDone** to inform the CrCORE that PHY initialization is done.
6. CrCORE should then automatically deassert **Cr_PePcsRstL**, **Cr_PeCOREPIPESRst**, and **Cr_PeCOREPIPEStkRst** (controlled respectively by bit[3], bit[5], and bit[7] of CrCORE register PMU3[1]_PE).

Firmware can check whether the IP PHY has problems in generating a stable pipe clock by doing the following steps:

1. Read PHY_READY field of PHY_STATUS register.
   - if PHY_READY = 1, pipe clock is already stable
   - if PHY_READY = 0, pipe clock is not yet stable. FW can opt to repeat checking this field for a preferred number of times.
2. If PHY_READY is still 0 after checking for the preferred number of times, FW can opt to manually deassert **Cr_PePcsRstL**, **Cr_PeCOREPIPESRst**, and **Cr_PeCOREPIPEStkRst**. FW should then again read PHY_READY.
   - if PHY_READY = 1, this means that **PePHYResetDone** was not deasserted by IP PHY but the pipe clock is already present.
   - if PHY_READY = 0, no pipe clock is present.
3. If no pipe clock is present, FW can opt to repeat the process for a preferred number of times before deciding that the IP PHY is defective. FW should ensure that it reasserts **Cr_PePcsRstL**, **Cr_PeCOREPIPESRst**, and **Cr_PeCOREPIPEStkRst** before repeating the process.

25.2.8.4.2 PERST# and Cold and Warm/Soft Reset

The EDC reset pins (for POR, etc) will generally be used for PCIe defined Cold and Warm/Soft Fundamental Resets.

In PCI Express Card Electromechanical (CEM) Specification, an auxiliary signal PERST# is described as follows:

> The PERST# signal is used to indicate when the power supply is within its specified voltage tolerance and is stable. It also initializes a component's state machines and other logic once power supplies stabilize. On power up, the deassertion of PERST# is delayed 100 ms (TPVPERL) from the power rails achieving specified operating limits. Also, within this time, the reference clocks (REFCLK+, REFCLK-) also become stable, at least TPERST-CLK before PERST# is deasserted. PERST# is asserted in advance of the power being switched off in a power-managed state like S3. PERST# is asserted when the power supply is powered down, but without the advanced warning of the transition.

On some applications, the PERST# signal from the card edge may also serve as POR.

In EDC Root Complex (RC) applications, the pin XbPe_PERST will be an output that can be used to control the PERST# signal in a card slot. Programming the output state is done via the PeDRF_PERSTNOUT field of PeDRF_PCIE_AUX control register.

In EDC Endpoint (EP) applications, the PERST# from the slot may be made to serve as POR or general reset by connecting to EDC reset pins. The pin XbPe_PERST will be an input that can be used to reset the device through software (Soft Reset):

1. XbPe_PERST input is pulled low to signify a reset request. If it is pulled low for at least 100us, a Critical Interrupt (Interrupt# 60) is generated in IcCORE. Otherwise, the

PERST# event is deemed as false; and no interrupt is generated.

2. Upon detecting the critical interrupt, FW reads PeCORE Interrupt Register 2, PeDRF_CINT2 and sees that PeDRF_CINT2[2] is asserted.
3. FW would then clear PeDRF_CINT2[2].
4. Core reset could now be asserted.

Also, a type of Soft Reset occurs during Endpoint transition from D3hot to D0 when the No_Soft_Reset (PeDRF_PMNSRES) bit of PeDRF_PMCSDAT register is deasserted. The process begins with the Endpoint (PeCORE) currently at D3hot, while link is currently at L1.

1. Endpoint triggers exit to L1.
2. LTSSM transitions from L1 to L0.
3. When in L0, Endpoint sends PME message to Root Complex when PME generation is enabled (peDRF_PMCSDAT_PmeEn is asserted). The PME message is used to request the Root Complex to change the Endpoint's device state from D3hot to D0.
4. Then, Root Complex does a Config Write to Endpoint in order to change its device state to D0.
5. After the Endpoint's device state has transitioned to D0, it then asserts PeDRF_D3HD0RSTINT bit of PeDRF_CINT2 register when its mask bit (PeDRF_D3HD0RSTMSK) is deasserted. This triggers a critical interrupt.
6. FW detects the interrupt and would then assert **Cr_PeCOREPIPESRst**, **Cr_PeCORECRRst**, and **Cr_PeCOREPIPEStkRst**.

25.2.8.4.3 Hot Reset

When the LTSSM detects an in-band Hot Reset request, a critical interrupt is generated in PeDRF_CINT2. The firmware may then decide when the device can reset itself.

1. When LTSSM receives the Hot Reset request, it asserts PeDRF_CINT2[1]. A Critical Interrupt (Interrupt# 60) is also generated in IcCORE.
2. Upon detecting the critical interrupt, FW reads PeCORE Interrupt Register 2, PeDRF_CINT2, and sees that PeDRF_CINT2[1] is asserted.
3. FW would then clear PeDRF_CINT2[1].
4. FW checks the validity of the reset request by reading PeDRF_HOTRST[16]. Note that the other device could opt to cancel the reset request.
   - If PeDRF_HOTRST[16] = 1, this signifies that the reset request from the other device is still valid. When FW is ready to reset the device, FW writes 1 to Hot Reset Ready (PeDRF_HOTRST[2]). FW waits 100us (this is for PeCORE LTSSM to acknowledge back Hot Reset), then FW asserts both **Cr_PeCOREPIPESRst** and **Cr_PeCORECRRst**.
   - If PeDRF_HOTRST[16] = 0, this signifies that the reset request from the other device was cancelled.

Refer to discussion on LTSSM for further details.

25.2.8.5 WAKE#

In PCI Express Card Electromechanical (CEM) Specification:

> The WAKE# signal is an open drain, active low signal that is driven low by a PCI Express component to reactivate the PCI Express slot's main power rails and reference clocks. Only add-in cards that support the wake process connect to this pin. If the add-in card has wakeup capabilities, it must support the WAKE# function. Likewise, only systems that support the wakeup function need to connect to this pin, but if they do, they must fully support the WAKE# function. Such systems are not required to support Beacon as a wakeup mechanism, but are encouraged to support it. If the wakeup process is used, the +3.3Vaux supply must be present and used for this function. The assertion and de-assertion of WAKE# are asynchronous to any system clock.

This is implemented in EDC pin XbPe_WAKE. In the absence of the in-band Beacon signal, this pin can be asserted low to wake the device from sleep. FW can assert WAKE# by setting PeDRF_LNKWAKEUP to 1.

25.2.9 PHY Interface

EDC uses GUC IP PHY for PCIe which implements the Physical Coding Layer (PCS) and Physical Media Attachment Layer (PMA) described in Intel's document PHY Interface for the PCI Express Architecture (PIPE). Control registers for the GUC PHY are defined in Section 25.3.2.

25.3 Core Registers

PeCORE registers are accessible to both the local processor and an external PCI host. These registers are divided into two major parts: (a) the PCI Express Configuration Space and the (b) PeCORE's Internal Registers. Both PCIe Configuration Space and Internal Registers are accessible to local processor through fix local addresses, while they are accessible to the Host on addresses assigned during initialization. PCIe Configuration Space is accessible to the Host through Configuration Request while the Internal Registers are accessible through BAR. ***Figure 25.38*** shows PeCORE's Register address mapping in Talino™ EDC's system memory.

Figure 25.38: PeCORE's Register Local Mapping

Internal Registers (125/250 MHz)	0xFFF2_0FFF <– 0xFFF2_0000
PCIe Configuration Space (125/250 MHz)	0xFFF2_1FFF <– 0xFFF2_1000
Internal Registers (125/250 MHz)	0xFFF2_7FFF <– 0xFFF2_2000
Internal Registers (333 MHz)	0xFFF2_FFFF <– 0xFFF2_8000

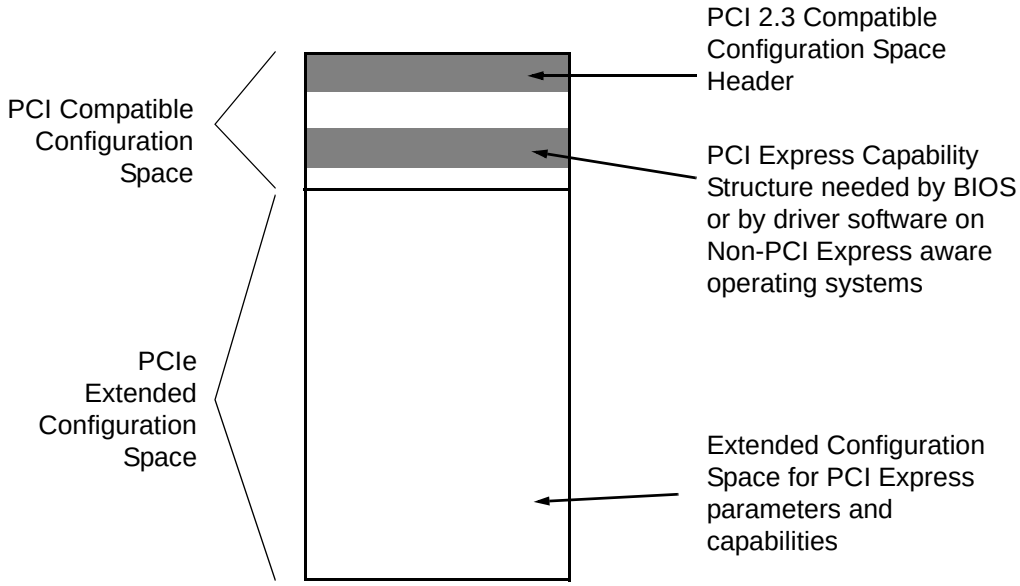
25.3.1 PCI Express Configuration Registers

Talino™ EDC's local processor configures PeCORE via CSR access through the Local CSR Bus (LCB), while Host accesses these registers though Configuration Requests.

These registers are accessed with single-beat (1 to 4 bytes) transfers. Software must take the necessary actions when accessing these registers (i.e. use appropriate masks when extracting only the desired bits during reads and preserving the values of other bits during writes).

The configuration space is divided into two sections (see Figure 25.39), the PCI Configuration space and the PCI Express Extend Configuration Space. The PCI configuration space contains the registers common (compatible) to PCI while the extended space is for the optional PCI Express Extended Capability Registers.

Figure 25.39: PCI Express Configuration Space divisions



The PCI Compatible Configuration Registers are then further divided into: PCI Common Header, PCI Header Type Specific Registers, and the PCI Capabilities Registers.

The following are the complete groups of PeCORE's internal and configuration registers:

- PCI Compatible Configuration Registers
  - PCI Common Header Configuration Registers
  - PCI Header Type 0 Specific Configuration Registers
  - PCI Capabilities Registers
    - MSI Capability Registers
    - PCI Power Management Capability Registers
    - PCI Express Capability Registers
- PCIe Extended Capability Registers
  - Advanced Error Reporting Capability Registers

- Device Serial Number Capability Registers
- Power Budgeting Capability Registers

- PeCORE Internal Registers
  - PeCORE Options Registers
  - PeCORE Interrupt Registers
  - Exchange Message Protocol Registers
  - Host Configuration Request Registers
  - Memory Request Registers
  - Credit Allocation Registers
  - PHY-related Registers

Details on these register categories and their members are discussed in the succeeding sub-sections. Table 25.19 to Table 25.24 tabulates each register, showing the address offset, type, and codename.

Table 25.19: PCI Common Header Configuration Registers (RC & EP)

Address Offset	Code	Register Name	Section
0x1000	PeDRF_PCDVEID	Device ID and Vendor ID Register	**Section 25.3.1.1.1**
0x1004	PeDRF_PCSTCMD	Status and Command Register	**Section 25.3.1.1.2**
0x1008	PeDRF_PCCREID	Class and Revision ID Register	**Section 25.3.1.1.3**
0x100C	PeDRF_PCBHLCS	Built-In Self Test, Header Type, Latency Timer and Cache Line Size Register	**Section 25.3.1.1.4**

Table 25.20: Header Type 0 Specific Configuration Registers (EP)

Address Offset	Code	Register Name	Section
0x1010	PeDRF_HTBAR0	BAR0	**Section 25.3.1.2.1**
0x1014	PeDRF_HTBAR1	BAR1	**Section 25.3.1.2.1**
0x1018	PeDRF_HTBAR2	BAR2	**Section 25.3.1.2.1**
0x101C	PeDRF_HTBAR3	BAR3	**Section 25.3.1.2.1**
0x1020	PeDRF_HTBAR4	BAR4	**Section 25.3.1.2.1**
0x1024	PeDRF_HTBAR5	BAR5	**Section 25.3.1.2.1**
0x1028	RSVD	Reserved	
0x102C	PeDRF_HTSVSID	Subsystem Vendor ID and Subsystem ID Register	**Section 25.3.1.2.2**
0x1030	PeDRF_HTERBAR	Expansion ROM Base Address Register	**Section 25.3.1.2.3**
0x1034	PeDRF_HTNCAPP	Next Capabilities Pointer Register	**Section 25.3.1.2.4**
0x1038	RSVD	Reserved	
0x103C	PeDRF_HTMMIIL	Interrupt Pin, and Interrupt Line Register	**Section 25.3.1.2.5**

Table 25.21: Header Type 1 Specific Configuration Registers (RC)

Address Offset	Code	Register Name	Section
0x1010	PeDRF_HTBAR0	BAR0	**Section 25.3.1.3.1**
0x1014	PeDRF_HTBAR1	BAR1	**Section 25.3.1.3.1**
0x1018	PeDRF_HTSSSPB	Secondary Latency Timer, Subordinate Bus Number, Secondary Bus Number, Primary Bus Number	**Section 25.3.1.3.2**
0x101C	PeDRF_HTSIOLB	Secondary Status, I/O Limit, I/O Base	**Section 25.3.1.3.3**
0x1020	PeDRF_HTMLMBA	Memory Limit, Memory Base	**Section 25.3.1.3.4**
0x1024	PeDRF_HTPMLMB	Prefetchable Memory Limit, Prefetchable Memory Base	**Section 25.3.1.3.5**
0x1028	PeDRF_HTPBU32	Prefetchable Base Upper 32 Bits	**Section 25.3.1.3.6**
0x102C	PeDRF_HTPLU32	Subsystem Vendor ID and Subsystem ID Register	**Section 25.3.1.3.7**
0x1030	PeDRF_HTIOLBU	I/O Limit Upper 16 Bits, I/O Base Upper 16 Bits	**Section 25.3.1.3.8**
0x1034	PeDRF_HTNCAPP	Next Capabilities Pointer Register	**Section 25.3.1.3.9**
0x1038	PeDRF_HTERBAR	Expansion ROM Base Address	**Section 25.3.1.3.10**
0x103C	PeDRF_HTBCIIL	Bridge Control, Interrupt Pin, and Interrupt Line Register	**Section 25.3.1.3.11**

Table 25.22: PCI Capabilities Registers

Address Offset	Code	Register Name	Section
0x1040	PeDRF_PMCPHDR	PM Capability Header Register	**Section 25.3.1.4.1**
0x1044	PeDRF_PMCSDAT	PM Control Status and Data Register	**Section 25.3.1.4.2**
0x1048	PeDRF_MSCPHDR	MSI Message Capability HeaderRegister	**Section 25.3.1.5.1**
0x104C	PeDRF_MSIMADD	MSI Message Address Register	**Section 25.3.1.5.2**
0x1050	PeDRF_MSMUADD	MSI Message Upper Address	**Section 25.3.1.5.3**
0x1054	PeDRF_MSMEIMD	MSI Message End Of Interrupt and Message Data Register	**Section 25.3.1.5.4**
0x1058	PeDRF_MSMASKX	MSI Mask Register	**Section 25.3.1.5.5**
0x105C	PeDRF_MSPENDI	MSI Pending Register	**Section 25.3.1.5.6**
0x1060	PeDRF_PECPHDR	PCIe Capability Header Register	**Section 25.3.1.6.1**
0x1064	PeDRF_PEDCAPA	PCIe Capability Device Capabilities Register	**Section 25.3.1.6.2**
0x1068	PeDRF_PEDSCTL	PCIe Capability Device Status and Control Register	**Section 25.3.1.6.3**
0x106C	PeDRF_PELKCAP	PCIe Capability Link Capabilities Register	**Section 25.3.1.6.4**
0x1070	PeDRF_PELKSCO	PCIe Capability Link Status and Control Register	**Section 25.3.1.6.5**
0x9074	PeDRF_PESCAPA	PCIe Capability Slot Capabilities Register	**Section 25.3.1.6.6**
0x9078	PeDRF_PESSCTL	PCIe Capability Slot Status and Control Register	**Section 25.3.1.6.7**
0x107C	PeDRF_PERCCTL	PCIe Capability Root Capabilities and Control Register	**Section 25.3.1.6.8**
0x1080	PeDRF_PERSTAT	PCIe Capability Root Status Register	**Section 25.3.1.6.9**
0x1084	PeDRF_PEDCAP2	PCIe Capability Device Capabilities 2 Register	**Section 25.3.1.6.10**
0x1088	PeDRF_PEDSCTL2	PCIe Capability Device Status 2 and Control 2 Register	**Section 25.3.1.6.11**
0x108C	Reserved	*PCIe Capability Link Capabilities 2 Register*	

0x1090	PeDRF_PELKSCO2	PCIe Capability Link Status 2 and Control 2 Register	**Section 25.3.1.6.12**
0x1094	Reserved	*PCIe Capability Slot Capabilities 2 Register*	
0x1098	Reserved	*PCIe Capability Slot Status 2 and Control 2 Register*	
0x109C	Reserved	*Additional Capability Register*	
0x10A0	PeDRF_PIML0	PCIe Inbound Map Local 0 Register	**Section 25.3.1.6.13**
0x10A4	PeDRF_PIML1	PCIe Inbound Map Local 1 Register	**Section 25.3.1.6.13**
0x10A8	PeDRF_PIML2	PCIe Inbound Map Local 2 Register	**Section 25.3.1.6.13**
0x10AC	PeDRF_PIML3	PCIe Inbound Map Local 3 Register	**Section 25.3.1.6.13**
0x1200*	PeDRF_PIML4	PCIe Inbound Map Local 4 Register	**Section 25.3.1.6.13**
0x1204*	PeDRF_PIML5	PCIe Inbound Map Local 5 Register	**Section 25.3.1.6.13**
0x10B0	PeDRF_PIMS0	PCIe Inbound Map Size 0 Register	**Section 25.3.1.6.14**
0x10B4	PeDRF_PIMS1	PCIe Inbound Map Size 1 Register	**Section 25.3.1.6.14**
0x10B8	PeDRF_PIMS2	PCIe Inbound Map Size 2 Register	**Section 25.3.1.6.14**
0x10BC	PeDRF_PIMS3	PCIe Inbound Map Size 3 Register	**Section 25.3.1.6.14**
0x1210*	PeDRF_PIMS4	PCIe Inbound Map Size 4 Register	**Section 25.3.1.6.14**
0x1214*	PeDRF_PIMS5	PCIe Inbound Map Size 5 Register	**Section 25.3.1.6.14**
0x10C0	PeDRF_POMP0	PCIe Outbound Map PCIe 0 Register	**Section 25.3.1.6.15**
0x10C4	PeDRF_POMP1	PCIe Outbound Map PCIe 1 Register	**Section 25.3.1.6.16**
0x10C8	PeDRF_POMP2	PCIe Outbound Map PCIe 2 Register	**Section 25.3.1.6.16**
0x10CC	PeDRF_POMP3	PCIe Outbound Map PCIe 3 Register	**Section 25.3.1.6.16**
0x1220*	PeDRF_POMP4	PCIe Outbound Map PCIe 4 Register	**Section 25.3.1.6.16**
0x1224*	PeDRF_POMP5	PCIe Outbound Map PCIe 5 Register	**Section 25.3.1.6.16**
0x10D0	PeDRF_POML0	PCIe Outbound Map Local 0 Register	**Section 25.3.1.6.17**
0x10D4	PeDRF_POML1	PCIe Outbound Map Local 1 Register	**Section 25.3.1.6.17**
0x10D8	PeDRF_POML2	PCIe Outbound Map Local 2 Register	**Section 25.3.1.6.17**
0x10DC	PeDRF_POML3	PCIe Outbound Map Local 3 Register	**Section 25.3.1.6.17**
0x1230*	PeDRF_POML4	PCIe Outbound Map Local 4 Register	**Section 25.3.1.6.17**
0x1234*	PeDRF_POML5	PCIe Outbound Map Local 5 Register	**Section 25.3.1.6.17**
0x10E0	PeDRF_POMS0	PCIe Outbound Map Size 0 Register	**Section 25.3.1.6.18**
0x10E4	PeDRF_POMS1	PCIe Outbound Map Size 1 Register	**Section 25.3.1.6.18**
0x10E8	PeDRF_POMS2	PCIe Outbound Map Size 2 Register	**Section 25.3.1.6.18**
0x10EC	PeDRF_POMS3	PCIe Outbound Map Size 3 Register	**Section 25.3.1.6.18**
0x1240*	PeDRF_POMS4	PCIe Outbound Map Size 4 Register	**Section 25.3.1.6.18**
0x1244*	PeDRF_POMS5	PCIe Outbound Map Size 5 Register	**Section 25.3.1.6.18**
0x1250	PeDRF_EXPIML	PCIe Expansion ROM Local Register	**Section 25.3.1.6.19**
0x1254	PeDRF_EXPIMS	PCIe Expansion ROM Size Register	**Section 25.3.1.6.20**

Table 25.23: PCIe Extended Capabilities Registers

Address Offset	Code	Register Name	Section
0x1100	PeDRF_AECPHDR	Advanced Error Reporting Capability Header Register	**Section 25.3.1.7.1**
0x1104	PeDRF_AEUESTA	Uncorrectable Error Status Register	**Section 25.3.1.7.2**
0x1108	PeDRF_AEUEMAS	Uncorrectable Error Mask Register	**Section 25.3.1.7.3**
0x110C	PeDRF_AEUESVT	Uncorrectable Error Severity Register	**Section 25.3.1.7.4**
0x1110	PeDRF_AECESTA	Correctable Error Status Register	**Section 25.3.1.7.5**
0x1114	PeDRF_AECEMAS	Correctable Error Mask Register	**Section 25.3.1.7.6**
0x1118	PeDRF_AECCTRL	Advanced Error Capabilities and Control Register	**Section 25.3.1.7.7**

0x111C	PeDRF_AEHW0LO	Header Log 0 Register	**Section 25.3.1.7.8**
0x1120	PeDRF_AEHW1LO	Header Log 1 Register	**Section 25.3.1.7.8**
0x1124	PeDRF_AEHW2LO	Header Log 2 Register	**Section 25.3.1.7.8**
0x1128	PeDRF_AEHW3LO	Header Log 3 Register	**Section 25.3.1.7.8**
0x112C	PeDRF_AERECOM	Root Error Command	**Section 25.3.1.7.9**
0x1130	PeDRF_AERESTA	Root Error Status	**Section 25.3.1.7.10**
0x1134	PeDRF_AEESOID	Error Source ID Register	**Section 25.3.1.7.11**
0x1138	PeDRF_SNCPHDR	Device Serial Number Capability Header Register	**Section 25.3.1.8.1**
0x113C	PeDRF_SNSN1DW	Serial Number First DW Register	**Section 25.3.1.8.2**
0x1140	PeDRF_SNSN2DW	Serial Number Second DW Register	**Section 25.3.1.8.2**
0x1144	PeDRF_PBCPHDR	Power Budgeting Capability Header Register	**Section 25.3.1.9.1**
0x1148	PeDRF_PBDSELE	Power Budgeting Data Select Register	**Section 25.3.1.9.2**
0x114C	PeDRF_PBDATAX	Power Budgeting Data Register	**Section 25.3.1.9.3**
0x1150	PeDRF_PBCAPAB	Power Budgeting Capability Register	**Section 25.3.1.9.4**

Table 25.24: PeCORE Internal Registers

Address Offset	Code	Register Name	Section
0x0000	PeDRF_CIR	Core Information Register	**Section 25.3.2.1**
0x0004	PeDRF_COPT	Core Options Register	**Section 25.3.2.2**
0x0008	PeDRF_CINT	Core Interrupt Status Register	**Section 25.3.2.3**
0x000C	PeDRF_SYSINT	System Interrupt Register	**Section 25.3.2.4**
0x0010	PeDRF_RXERR	Receiver Error Register	**Section 25.3.2.5**
0x0014	PeDRF_CACTL	Completer Abort Control Register	**Section 25.3.2.6**
0x0018	PeDRF_URCTL	Unsupported Request Control Register	**Section 25.3.2.7**
0x001C	PeDRF_MALFCTL	Malformed Detection Control Register	**Section 25.3.2.8**
0x0020	PeDRF_PDCR	Pause DMA Control Register	**Section 25.3.2.9**
0x0024	PeDRF_PDSR	Pause DMA Status Register	**Section 25.3.2.10**
0x0028	PeDRF_DERRL	8b18b Decode Error Limit Register	**Section 25.3.2.11**
0x002c	PeDRF_PHYERCTL	Physical Layer Error Control Register	**Section 25.3.2.12**
0x0030	PeDRF_UPFC	UpdateFC Latency Control Register	**Section 25.3.2.13**
0x0034	PeDRF_CPMR	Completion Per MRd Register	**Section 25.3.2.14**
0x0038	PeDRF_AKLAT	AckNak Latency Control Register	**Section 25.3.2.15**
0x003c	reserved		
0x0040 / 0x0CF8	PeDRF_CRAR	Configuration Request Address Register	**Section 25.3.2.16**
0x0044 / 0x0CFC	PeDRF_CRDR	Configuration Request Data Register	**Section 25.3.2.17**
0x0048	PeDRF_CRCR	Configuration Request Control Register	**Section 25.3.2.18**
0x004C	PeDRF_CRSR	Configuration Request Status Register	**Section 25.3.2.19**
0x0050	PeDRF_CRBAR	Configuration Request Base Address Register	**Section 25.3.2.20**
0x0100	PeDRF_MRLAR0	Memory Request Local Address Register 0	**Section 25.3.2.21**
0x0104	PeDRF_MRLAR1	Memory Request Local Address Register 1	**Section 25.3.2.21**
0x0108	PeDRF_MRLAR2	Memory Request Local Address Register 2	**Section 25.3.2.21**
0x010C	PeDRF_MRLAR3	Memory Request Local Address Register 3	**Section 25.3.2.21**
0x0110	PeDRF_MRALR0	Memory Request Low Address Register 0	**Section 25.3.2.22**
0x0114	PeDRF_MRALR1	Memory Request Low Address Register 1	**Section 25.3.2.22**
0x0118	PeDRF_MRALR2	Memory Request Low Address Register 2	**Section 25.3.2.22**
0x011C	PeDRF_MRALR3	Memory Request Low Address Register 3	**Section 25.3.2.22**

Address Offset	Code	Register Name	Section
0x0120	PeDRF_MRAHR0	Memory Request High Address Register 0	**Section 25.3.2.23**
0x0124	PeDRF_MRAHR1	Memory Request High Address Register 1	**Section 25.3.2.23**
0x0128	PeDRF_MRAHR2	Memory Request High Address Register 2	**Section 25.3.2.23**
0x012C	PeDRF_MRAHR3	Memory Request High Address Register 3	**Section 25.3.2.23**
0x0130	PeDRF_MRCR0	Memory Request Control Register 0	**Section 25.3.2.24**
0x0134	PeDRF_MRCR1	Memory Request Control Register 1	**Section 25.3.2.24**
0x0138	PeDRF_MRCR2	Memory Request Control Register 2	**Section 25.3.2.24**
0x013C	PeDRF_MRCR3	Memory Request Control Register 3	**Section 25.3.2.24**
0x0140	PeDRF_MRSR0	Memory Request Status Register 0	**Section 25.3.2.25**
0x0144	PeDRF_MRSR1	Memory Request Status Register 1	**Section 25.3.2.25**
0x0148	PeDRF_MRSR2	Memory Request Status Register 2	**Section 25.3.2.25**
0x014C	PeDRF_MRSR3	Memory Request Status Register 3	**Section 25.3.2.25**
0x0150	PeDRF_MRLR0	Memory Request Length Register 0	**Section 25.3.2.26**
0x0154	PeDRF_MRLR1	Memory Request Length Register 1	**Section 25.3.2.26**
0x0158	PeDRF_MRLR2	Memory Request Length Register 2	**Section 25.3.2.26**
0x015c	PeDRF_MRLR3	Memory Request Length Register 3	**Section 25.3.2.26**
0x0160	PeDRF_MRDR0	Memory Request Data Register 0	**Section 25.3.2.27**
0x0164	PeDRF_MRDR1	Memory Request Data Register 1	**Section 25.3.2.27**
0x0168	PeDRF_MRDR2	Memory Request Data Register 2	**Section 25.3.2.27**
0x016C	PeDRF_MRDR3	Memory Request Data Register 3	**Section 25.3.2.27**
0x0300	PeDRF_TDMT	Tx Data Minimum Threshold	**Section 25.3.2.28**
0x0304	PeDRF_MORP	Maximum Outstanding Request Partition	**Section 25.3.2.29**
0x0400	PeDRF_CPP0	EMP Command Put Pointer 0	**Section 25.3.2.30**
0x0404	PeDRF_CPP1	EMP Command Put Pointer 1	**Section 25.3.2.30**
0x0408	PeDRF_CPP2	EMP Command Put Pointer 2	**Section 25.3.2.30**
0x040C	PeDRF_CPP3	EMP Command Put Pointer 3	**Section 25.3.2.30**
0x0410	PeDRF_CPP4	EMP Command Put Pointer 4	**Section 25.3.2.30**
0x0414	PeDRF_CPP5	EMP Command Put Pointer 5	**Section 25.3.2.30**
0x0418	PeDRF_CPP6	EMP Command Put Pointer 6	**Section 25.3.2.30**
0x041C	PeDRF_CPP7	EMP Command Put Pointer 7	**Section 25.3.2.30**
0x0420	PeDRF_CGP0	EMP Command Get Pointer 0	**Section 25.3.2.31**
0x0424	PeDRF_CGP1	EMP Command Get Pointer 1	**Section 25.3.2.31**
0x0428	PeDRF_CGP2	EMP Command Get Pointer 2	**Section 25.3.2.31**
0x042C	PeDRF_CGP3	EMP Command Get Pointer 3	**Section 25.3.2.31**
0x0430	PeDRF_CGP4	EMP Command Get Pointer 4	**Section 25.3.2.31**
0x0434	PeDRF_CGP5	EMP Command Get Pointer 5	**Section 25.3.2.31**
0x0438	PeDRF_CGP6	EMP Command Get Pointer 6	**Section 25.3.2.31**
0x043C	PeDRF_CGP7	EMP Command Get Pointer 7	**Section 25.3.2.31**
0x0440	PeDRF_RPP0	EMP Response Put Pointer 0	**Section 25.3.2.32**
0x0444	PeDRF_RPP1	EMP Response Put Pointer 1	**Section 25.3.2.32**
0x0448	PeDRF_RPP2	EMP Response Put Pointer 2	**Section 25.3.2.32**
0x044C	PeDRF_RPP3	EMP Response Put Pointer 3	**Section 25.3.2.32**
0x0450	PeDRF_RPP4	EMP Response Put Pointer 4	**Section 25.3.2.32**
0x0454	PeDRF_RPP5	EMP Response Put Pointer 5	**Section 25.3.2.32**
0x0458	PeDRF_RPP6	EMP Response Put Pointer 6	**Section 25.3.2.32**
0x045C	PeDRF_RPP7	EMP Response Put Pointer 7	**Section 25.3.2.32**
0x0460	PeDRF_RGP0	EMP Response Get Pointer 0	**Section 25.3.2.33**
0x0464	PeDRF_RGP1	EMP Response Get Pointer 1	**Section 25.3.2.33**
0x0468	PeDRF_RGP2	EMP Response Get Pointer 2	**Section 25.3.2.33**

Address Offset	Code	Register Name	Section
0x046C	PeDRF_RGP3	EMP Response Get Pointer 3	**Section 25.3.2.33**
0x0470	PeDRF_RGP4	EMP Response Get Pointer 4	**Section 25.3.2.33**
0x0474	PeDRF_RGP5	EMP Response Get Pointer 5	**Section 25.3.2.33**
0x0478	PeDRF_RGP6	EMP Response Get Pointer 6	**Section 25.3.2.33**
0x047C	PeDRF_RGP7	EMP Response Get Pointer 7	**Section 25.3.2.33**
0x0480	PeDRF_HATR	EMP Host Attention Register	**Section 25.3.2.34**
0x0484	PeDRF_PATR	EMP Port Attention Register	**Section 25.3.2.35**
0x0488	PeDRF_PSTR	EMP Port Status Register	**Section 25.3.2.36**
0x048C	PeDRF_PCLR	EMP Port Control Register	**Section 25.3.2.37**
0x0600	PeDRF_CAPD	Credit Allocation - Posted Data Register	**Section 25.3.2.38**
0x0604	PeDRF_CAPH	Credit Allocation - Posted Header Register	**Section 25.3.2.39**
0x0608	PeDRF_CAND	Credit Allocation - Posted Data Register	**Section 25.3.2.40**
0x060C	PeDRF_CANH	Credit Allocation - Posted Header Register	**Section 25.3.2.41**
0x0610	PeDRF_CACD	Credit Allocation - Posted Data Register	**Section 25.3.2.42**
0x0614	PeDRF_CACH	Credit Allocation - Posted Header Register	**Section 25.3.2.43**
0x0700	PeDRF_RXMSIBAR	MSI Base Address Register	**Section 25.3.2.44**
0x0704	PeDRF_RXMSIDATA	MSI Data Register	**Section 25.3.2.45**
0x0708	PeDRF_MSICFG0	MSI Configuration Register 0	**Section 25.3.2.46**
0x070C	PeDRF_MSICFG1	MSI Configuration Register 1	**Section 25.3.2.47**
0x0710	PeDRF_MSICFG2	MSI Configuration Register 2	**Section 25.3.2.48**
0x0714	PeDRF_MSICFG3	MSI Configuration Register 3	**Section 25.3.2.49**
0x0718	PeDRF_MSICFG4	MSI Configuration Register 4	**Section 25.3.2.50**
0x071C	PeDRF_MSICFG5	MSI Configuration Register 5	**Section 25.3.2.51**
0x0720	PeDRF_MSICFG6	MSI Configuration Register 6	**Section 25.3.2.52**
0x0724	PeDRF_MSICFG7	MSI Configuration Register 7	**Section 25.3.2.53**
0x0728	PeDRF_HAACLR	Host Attention AutoClear Register	**Section 25.3.2.54**
0x0780	PeDRF_PMDATA0	Power Management Data 0	**Section 25.3.2.55**
0x0784	PeDRF_PMDATA1	Power Management Data 1	**Section 25.3.2.55**
0x0788	PeDRF_PMDATA2	Power Management Data 2	**Section 25.3.2.55**
0x078C	PeDRF_PMDATA3	Power Management Data 3	**Section 25.3.2.55**
0x0790	PeDRF_PMDATA4	Power Management Data 4	**Section 25.3.2.55**
0x0794	PeDRF_PMDATA5	Power Management Data 5	**Section 25.3.2.55**
0x0798	PeDRF_PMDATA6	Power Management Data 6	**Section 25.3.2.55**
0x079C	PeDRF_PBDATA7	Power Management Data 7	**Section 25.3.2.55**
0x07A0	PeDRF_PBDATA0	Power Budgeting Data 0	**Section 25.3.2.56**
0x07A4	PeDRF_PBDATA1	Power Budgeting Data 1	**Section 25.3.2.56**
0x07A8	PeDRF_PBDATA2	Power Budgeting Data 2	**Section 25.3.2.56**
0x07AC	PeDRF_PBDATA3	Power Budgeting Data 3	**Section 25.3.2.56**
0x07B0	PeDRF_PBDATA4	Power Budgeting Data 4	**Section 25.3.2.56**
0x07B4	PeDRF_PBDATA5	Power Budgeting Data 5	**Section 25.3.2.56**
0x8800	PeDRF_LTSR	Link Training Status Register	**Section 25.3.2.57**
0x0804	PeDRF_LTCTL	Link Training Controller Register	**Section 25.3.2.58**
0x0808	PeDRF_LASCC	LTSSM Auxilliary Status, Control, and Capability	**Section 25.3.2.59**
0x080C	PeDRF_SPMC	System Power Management Control Register	**Section 25.3.2.60**
0x0900	PeDRF_SRAMTSTCFG1	SRAM Test Configuration Register	**Section 25.3.2.61**
0x0904	PeDRF_TSRAM_PE1	SRAM Test Parity Error Injection	**Section 25.3.2.62**
0x0908	PeDRF_HDRBUF_Addr	Rx Header Buffer SRAM Test Address	**Section 25.3.2.63**
0x090C	PeDRF_HDRBUF_Data0	Rx Header Buffer SRAM Test Data W0	**Section 25.3.2.64**

Address Offset	Code	Register Name	Section
0x0910	PeDRF_HDRBUF_Data1	Rx Header Buffer SRAM Test Data W1	**Section 25.3.2.65**
0x0914	PeDRF_RTRYBUF_Addr	Tx Retry Buffer SRAM Test Address	**Section 25.3.2.66**
0x0918	PeDRF_RTRYBUF_Data0	Tx Retry Buffer SRAM Test Data W0	**Section 25.3.2.67**
0x091C	PeDRF_RTRYBUF_Data1	Tx Retry Buffer SRAM Test Data W1	**Section 25.3.2.68**
0x0920	PeDRF_IOBUF0_Addr	IO Buffer SRAM Test Address	**Section 25.3.2.69**
0x0924	PeDRF_IOBUF0_Data0	IO Buffer SRAM Test Data W0	**Section 25.3.2.70**
0x0928	PeDRF_IOBUF0_Data1	IO Buffer SRAM Test Data W1	**Section 25.3.2.71**
0x092C	PeDRF_BISTCTL	BIST Control Register	**Section 25.3.2.72**
0x0930	PeDRF_BISTPTRN1	BIST Pattern Register 1	**Section 25.3.2.73**
0x0934	PeDRF_BISTPTRN2	BIST Pattern Register 2	**Section 25.3.2.74**
0x0938	PeDRF_BISTSTAT	BIST Status Register	**Section 25.3.2.75**
0x0940	PeDRF_PCS0DBG	GUC PCS 0 Debug Register	**Section 25.3.2.76**
0x0944	PeDRF_PCS1DBG	GUC PCS 1 Debug Register	**Section 25.3.2.77**
0x0948	PeDRF_PCS2DBG	GUC PCS 2 Debug Register	**Section 25.3.2.78**
0x094C	PeDRF_PCS3DBG	GUC PCS 3 Debug Register	**Section 25.3.2.79**
0x0950	PeDRF_PCS4DBG	GUC PCS 4 Debug Register	**Section 25.3.2.80**
0x0954	PeDRF_PCS5DBG	GUC PCS 5 Debug Register	**Section 25.3.2.81**
0x0958	PeDRF_PCS6DBG	GUC PCS 6 Debug Register	**Section 25.3.2.82**
0x095C	PeDRF_PCS7DBG	GUC PCS 7 Debug Register	**Section 25.3.2.83**
0x0960	PeDRF_PMADDBG	GUC PMAD Debug Register	**Section 25.3.2.84**
0x893C	PeDRF_GUCPMABO	GUC PHY PMA BIST Observation	**Section 25.3.2.85**
0x8940	PeDRF_GUCPMABC1	GUC PHY PMA BIST Controller 1	**Section 25.3.2.86**
0x8944	PeDRF_GUCPMABC2	GUC PHY PMA BIST Controller 2	**Section 25.3.2.87**
0x8948	PeDRF_GUCPMARXC	GUC PHY PMA Rx Controller	**Section 25.3.2.88**
0x894C	PeDRF_GUCPMAPLLC	GUC PHY PMA PLL Controller	**Section 25.3.2.89**
0x8950	PeDRF_GUCPMATXC0	GUC PHY PMA Tx Controller Lane 0	**Section 25.3.2.90**
0x8954	PeDRF_GUCPMATXC1	GUC PHY PMA Tx Controller Lane 1	**Section 25.3.2.90**
0x8958	PeDRF_GUCPMATXC2	GUC PHY PMA Tx Controller Lane 2	**Section 25.3.2.90**
0x895C	PeDRF_GUCPMATXC3	GUC PHY PMA Tx Controller Lane 3	**Section 25.3.2.90**
0x8960	PeDRF_GUCPMATXC4	GUC PHY PMA Tx Controller Lane 4	**Section 25.3.2.90**
0x8964	PeDRF_GUCPMATXC5	GUC PHY PMA Tx Controller Lane 5	**Section 25.3.2.90**
0x8968	PeDRF_GUCPMATXC6	GUC PHY PMA Tx Controller Lane 6	**Section 25.3.2.90**
0x896C	PeDRF_GUCPMATXC7	GUC PHY PMA Tx Controller Lane 7	**Section 25.3.2.90**
0x8970	PeDRF_GUCPMARXC0	GUC PHY PMA Rx Controller Lane 0	**Section 25.3.2.91**
0x8974	PeDRF_GUCPMARXC1	GUC PHY PMA Rx Controller Lane 1	**Section 25.3.2.91**
0x8978	PeDRF_GUCPMARXC2	GUC PHY PMA Rx Controller Lane 2	**Section 25.3.2.91**
0x897C	PeDRF_GUCPMARXC3	GUC PHY PMA Rx Controller Lane 3	**Section 25.3.2.91**
0x8980	PeDRF_GUCPMARXC4	GUC PHY PMA Rx Controller Lane 4	**Section 25.3.2.91**
0x8984	PeDRF_GUCPMARXC5	GUC PHY PMA Rx Controller Lane 5	**Section 25.3.2.91**
0x8988	PeDRF_GUCPMARXC6	GUC PHY PMA Rx Controller Lane 6	**Section 25.3.2.91**
0x898C	PeDRF_GUCPMARXC7	GUC PHY PMA Rx Controller Lane 7	**Section 25.3.2.91**
0x8990	PeDRF_GUCPMARXDC0	GUC PHY PMA Rx DFE Controller Lane 0	**Section 25.3.2.92**
0x8994	PeDRF_GUCPMARXDC1	GUC PHY PMA Rx DFE Controller Lane 1	**Section 25.3.2.92**
0x8998	PeDRF_GUCPMARXDC2	GUC PHY PMA Rx DFE Controller Lane 2	**Section 25.3.2.92**
0x899C	PeDRF_GUCPMARXDC3	GUC PHY PMA Rx DFE Controller Lane 3	**Section 25.3.2.92**
0x89A0	PeDRF_GUCPMARXDC4	GUC PHY PMA Rx DFE Controller Lane 4	**Section 25.3.2.92**
0x89A4	PeDRF_GUCPMARXDC5	GUC PHY PMA Rx DFE Controller Lane 5	**Section 25.3.2.92**
0x89A8	PeDRF_GUCPMARXDC6	GUC PHY PMA Rx DFE Controller Lane 6	**Section 25.3.2.92**
0x89AC	PeDRF_GUCPMARXDC7	GUC PHY PMA Rx DFE Controller Lane 7	**Section 25.3.2.92**

Address Offset	Code	Register Name	Section
0x8000	PeDRF_PHY_STATUS	Physical Layer Status	**Section 25.3.2.93**
0x8004	PeDRF_PCIE_AUX	PCIe Auxiliary Signals	**Section 25.3.2.94**
0x8008	PeDRF_LOOPBACK	PCIe Loopback Control	**Section 25.3.2.95**
0x9000	PeDRF_CINT2	PeCORE Interrupt Register 2	**Section 25.3.2.96**
0x9004	PeDRF_HOTRST	Hot Reset Status and Control	**Section 25.3.2.97**
0x907C	PeDRF_PERST	PERST Timeout Control	**Section 25.3.2.98**
0x7FFC	PeDRF_DBG0	Debug CSR 0	**Section 25.3.2.99**
0x7FF8	PeDRF_DBG1	Debug CSR 1	**Section 25.3.2.100**
0x7FF4	PeDRF_DBG2	Debug CSR 2	**Section 25.3.2.101**
0x7FF0	PeDRF_DBG3	Debug CSR 3	**Section 25.3.2.102**
0x7FEC	PeDRF_DBG4	Debug CSR 4	**Section 25.3.2.103**
0x7FE8	PeDRF_DBG5	Debug CSR 5	**Section 25.3.2.104**
0x7FE4	PeDRF_DBG6	Debug CSR 6	**Section 25.3.2.105**
0x7FE0	PeDRF_DBG7	Debug CSR 7	**Section 25.3.2.106**
0x7FDC	PeDRF_DBG8	Debug CSR 8	**Section 25.3.2.107**
0x7FD8	PeDRF_DBG9	Debug CSR 9	**Section 25.3.2.108**

PCI Configuration Space is accessed through ID-based routing, so the actual address of PCI standard configuration registers, in Root Complex point of view, is determined during initialization, that is, when Bus#, Device#, Function# are assigned to the device. Note that Internal Registers are not visible to Root Complex as part of PCI Configuration Space. These registers can be accessed, though, through one of the BARs, that is, through Memory or IO Request transaction. Not all the internal registers of PeCORE, however, are accessible through memory or IO request. Only the Exchange Message Protocol registers are exposed to the Root Complex.

25.3.1.1 PCI Common Header Configuration Registers (RC & EP)

The following subsections detail the registers common to PCI 3.0 Type 0 and Type 1 Configuration Space Header devices. These registers are instantiated per function implemented by the PCI Express device.

25.3.1.1.1 PeDRF_PCDVEID: Device and Vendor ID Register (RC & EP)

PCIe Offset: **0x1000**

Table 25.25: PeDRF_PCDVEID

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RW	RO	16'h192A	PeDRF_PCVENID	Vendor ID
[31:16]	RW	RO	16'h0007	PeDRF_PCDEVID	Device ID

[15:00] PeDRF_PCVENID - Vendor ID

This 16-bit register is used to identify the manufacturer of the PCI device. The value in this register is assigned by the PCI-SIG to PCI Vendor companies. The Vendor ID assigned by the PCI-SIG to BiTMICRO is 192Ah.

[31:16] PeDRF_PCDEVID - Device ID

This 16-bit register is used to identify the PCI device. The local processor has read-write access to this field. PeCORE is assigned with 0007h value as the Device ID.

25.3.1.1.2 PeDRF_PCSTCMD: Status and Command Register (RC & EP)

PCIe Offset: **0x1004**

Table 25.26: PeDRF_PCSTCMD

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_PCIOSCT	IO Space Control
[01]	RW	RW	1'b0	PeDRF_PCMSCTL	Memory Space Control
[02]	RW	RW	1'b0	PeDRF_PCBMENA	Bus Master Enable
[05:03]	RO	RO	3'b000	RSVD	Reserved
[06]	RW	RW	1'b0	PeDRF_PCPERES	Parity Error Response
[07]	RO	RO	1'b0	RSVD	Reserved
[08]	RW	RW	1'b0	PeDRF_PCSECTL	System Error Control
[09]	RO	RO	1'b0	RSVD	Reserved
[10]	RW	RW	1'b0	PeDRF_PCINTDI	Interrupt Disable
[18:11]	RO	RO	8'h00	RSVD	Reserved
[19]	RO	RO	1'b0	PeDRF_PCINTST	Interrupt Status
[20]	RO	RO	1'b1	PeDRF_PCCLIST	Capabilities List
[23:21]	RO	RO	3'b000	RSVD	Reserved
[24]	RW1C	RW1C	1'b0	PeDRF_PCMDPER	Status Master Data Parity Error
[26:25]	RO	RO	2'b00	RSVD	Reserved
[27]	RW1C	RW1C	1'b0	PeDRF_PCSTABR	Status Signaled Target Abort
[28]	RW1C	RW1C	1'b0	PeDRF_PCRTABR	Status Received Target Abort
[29]	RW1C	RW1C	1'b0	PeDRF_PCRMABR	Status Received Master Abort
[30]	RW1C	RW1C	1'b0	PeDRF_PCSSERR	Status Signaled System Error
[31]	RW1C	RW1C	1'b0	PeDRF_PCDPERR	Status Detected Parity Error

[00] PeDRF_PCIOSCT - IO Space Control

Table 25.27: PeDRF_PCIOSCT Field Encoding

Value	Function
1'b0	Disabled the function from responding to I/O accesses
1'b1	Enable the function to respond to I/O accesses

[01] PeDRF_PCMSCTL - Memory Space Control

Table 25.28: PeDRF_PCMSCTL Field Encoding

Value	Function
1'b0	Disabled the function from responding to memory accesses

Table 25.28: PeDRF_PCMSCTL Field Encoding

Value	Function
1'b1	Enable the function to respond to memory accesses

[02] PeDRF_PCBMENA - Bus Master Enable

This bit enables PeCORE to generate Memory and I/O Read/Write Requests as an Endpoint, and forward Memory and I/O Read/Write Requests in the upstream direction as a Root or Switch Port. When 0, PeCORE is prevented to generate any Memory or I/O Requests, including MSI/MSI-X interrupt Messages as an Endpoint, or forwards any Memory or I/O Requests received from the downstream port to the upstream port as a Root or Switch Port. All Memory and I/O Requests received from Root port are treated as Unsupported Requests (UR), and to all Non-Posted Requests received from the Switch downstream port, Completion with UR completion status must be returned, if this bit is disabled.

Table 25.29: CBME Field Encoding

Value	Function
1'b0	Disable the function from generating reads or writes to memory or I/O spaces
1'b1	Enable the function from generating reads or writes to memory or I/O spaces

[05:03] RSVD - Reserved

This bit does not apply to PCIe. Hardwired to 0.

[06] PeDRF_PCPERES - Parity Error Response

Enables the device of Parity Error reporting

Table 25.30: PeDRF_PCPERES Field Encoding

Value	Function
1'b0	Disable Parity Error reporting
1'b1	Enable Parity Error reporting

[07] RSVD - Reserved

This bit does not apply to PCIe. Hardwired to 0.

[08] **PeDRF_PCSECTL - System Error Control**

This bit enables the reporting of Non-fatal and Fatal errors detected by the PCIe device to the Root Complex. For devices with Type 1 configuration space headers, this bit enables the transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error messages forwarded from the secondary interface. Note that errors can also be reported through specific bits in the Device Control register.

[09] **RSVD - Reserved**

This bit does not apply to PCIe.

[10] **PeDRF_PCINTDI - Interrupt Disable**

This bit enables PeCORE to generate INTx interrupt Messages. Note that INTx emulation interrupts already asserted by the device, except those that are forwarded by Root and Switch Ports from devices downstream of the Root or Switch Port, must be deasserted when this bit is set.

[18:11] **RSVD - Reserved**

[19] **PeDRF_PCINTST - Interrupt Status**

This bit indicates if an INTx interrupt Message is pending in the device internally. Note that INTx emulation interrupts already asserted by the device, except those that are forwarded by Root and Switch Ports from devices downstream of the Root or Switch Port, must be deasserted when this bit is set.

[20] **PeDRF_PCCLIST - Capabilities List**

This bit indicates whether PeCORE implements the pointer for a new capabilities linked list at offset 34h or not. Since all PCIe devices are required to implement the PCIe capability structure, this bit must be set to 1.

[23:21] **RSVD - Reserved**

This bit does not apply to PCIe. Hardwired to 0.

[24] **PeDRF_PCMDPER - Master Data Parity Error**

This bit is set by Requestor, Primary Side for Type 1 Configuration Space header device, if the Command Parity Error Response bit (CPER) in the Command register is set and either the Requestor receives a poisoned Completion or the Requestor poisons a write Request.

[26:25] **RSVD - Reserved**

This bit does not apply to PCIe. Hardwired to 0.

[27] PeDRF_PCSTABR - Signaled Target Abort

This bit is set when a device, Primary Side for Type 1 Configuration Space header device, completes a Request using Completer Abort Completion Status.

[28] PeDRF_PCRTABR - Received Target Abort

This bit is set when a Requestor, Primary Side for Type 1 header Configuration Space header device for requests initiated by the Type 1 header device, receives a Completion with Completer Abort (CA) Completion Status.

[29] PeDRF_PCRMABR - Received Master Abort

This bit is set when a Requestor, Primary Side for Type 1 header Configuration Space header device for requests initiated by the Type 1 header device, receives a Completion with Unsupported Request (UR) Completion Status.

[30] PeDRF_PCSSERR - Signaled System Error

This bit is set when a device sends an ERR_FATAL or ERR_NONFATAL Message, and the Command SERR# Enable bit (CSE) of the Command register is set.

[31] PeDRF_PCDPERR - Detected Parity Error

This bit is set by a device, Primary Side for Type 1 Configuration Space header device, whenever it receives a Poisoned TLP. Note that setting of this bit is not affected by the current value of the Command Parity Error Response bit (CPER) of the Command register.

25.3.1.1.3 PeDRF_PCCREID: Class Code and Revision ID Register *(RC & EP)*

PCIe Offset: **0x1008**

Table 25.31: PeDRF_PCCREID

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RO	8'h01	PeDRF_PCREVID	Revision ID
[31:08]	RW	RO	24'h060000	PeDRF_PCCCODE	Class Code

[07:00] PeDRF_PCREVID - Revision ID

This 8-bit register is used to indicate the current incremental revision number of the device.

[31:08] PeDRF_PCCCODE - Class Code

This field identifies the generic class of devices to which this function belongs and (in some cases) its register level programming interface.

Note: Encoding for Class Code is provided in the Appendix D of the *PCI Local Bus Specification*, Revision 3.0

25.3.1.1.4 PeDRF_PCBHLCS: Header Type/Cache Line Size Register *(RC & EP)*

PCIe Offset: **0x100C**

Table 25.32: PeDRF_PCBHLCS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RW	8'h00	PeDRF_PCCLSIZ	Cache Line Size
[15:08]	RO	RO	8'h0	RSVD	Reserved
[23:16]	RW	RO	8'h0	PeDRF_PCHDRTY	Header Type
[31:24]	RO	RO	8'h0	RSVD	Reserved

[07:00] PeDRF_PCCLSIZ - Cache Line Size

This 8-bit register is set by the firmware and the operating system to system cache line size. This field is implemented as a read/write by PCIe devices for legacy compatibility purposes but has not impact on any PCIe device functionality.

[15:08] RSVD - Reserved

This field does not apply to PCIe. This register must be hardwired to 0.

[23:16] PeDRF_PCHDRTY - Header Type

This 8-bit register is used to identify the second part of the PCI header (beginning at offset 10h) and determines whether a PCI device is a single or multiple functions device. Bit 7 of this register is used to identify a multi-function device. If bit 7 is 0, then the device is a single function. If bit 7 is 1, then the device has multi-functions.

Table 25.33: PeDRF_PCHDRTY encoding

Value	Function
8'h00	core has one function, using header type 0
8'h01	core has one function, using header type 1
8'h80	core has multiple function, using header type 0
8'h81	core has multiple function, using header type 1
	All other encoding are reserved

[31:24] RSVD - Reserved

This field does not apply to PeCORE. This register must hardwired to 0.

25.3.1.2 PCI Header Type 0 Specific Configuration Space Registers *(EP)*

The following subsections detail the registers of PCI 3.0 Type 0 Configuration Space header for PCIe devices.

25.3.1.2.1 PeDRF_HTBAR0/1/2/3/4/5: BAR0/1/2/3/4/5 Register *(EP)*

PCIe Offset: **0x1010**, **0x1014**, **0x1018**, **0x101c**, **0x1020**, **0x1024**

Table 25.34: PeDRF_HTBAR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_HTMI	Space Indicator
[02:01]	RW	RO	2'b00	PeDRF_HTTYP	Memory Type
[03]	RW	RO	1'b1	PeDRF_HTPREF	Prefetchable
[07:04]	RO	RO	4'h00	RSVD	Reserved
[31:08]	RO	RW	24'hfff000	PeDRF_HTBA	Base Address
[07:01]	RO	RO	7'h00	RSVD	Reserved
[31:08]	RO	RW	24'hfff000	PeDRF_HTBA	Base Address

BAR register defines a region of System PCI Memory space owned by Talino™ EDC. This 32-bit register is enabled by the Enable bit of PeDRF_PIMS register.

The mapping of address from PCI address to local address is controlled by PIM-BAR register pair.

[00] PeDRF_HTMI - Space Indicator

This indicates what space is defined by PeDRF_HTBAR.

Table 25.35: PeDRF_HTMI

	Space Indicator
1'b0	Memory Space
1'b1	IO Space

If PeDRF_HTBAR[0] is 0,

[02:01] PeDRF_HTTYP - Memory Type

This field defines the memory type.

Table 25.36: PeDRF_HTTYP encoding

Value	Function
2'b00	Base address is 32-bits, and may be set anywhere within the 32-bit range
2'b01	reserved
2'b10	Base address is 64-bits, and may be set anywhere within the 64-bit range. Pairing is limited to this: BAR0-BAR1 BAR2-BAR3 BAR4-BAR5
2'b11	reserved

[03] PeDRF_HTPREF - Prefetchable

This bit indicates if the region is prefetchable.

Table 25.37: PeDRF_HTPREF Encoding

Value	Function
1'b0	Memory is not prefetchable
1'b1	Memory is prefetchable

[07:04] RSVD - Reserved

[31:08] PeDRF_HTBA - Base Address

These bits initially contain the size of the requested PCI Memory space. The Root Complex reads this value, then assigns an address range by writing to this register. The writable bits of this field depends on the requested size.

Table 25.38: PeDRF_HTBA

PeDRF_HTBAR[31:08]	Size	Writable bits
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xxx1	256B	[31:08]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xx10	512B	[31:09]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_x100	1KB	[31:10]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_1000	2KB	[31:11]
24'bxxxx_xxxx_xxxx_xxxx_xxx1_000	4KB	[31:12]
24'bxxxx_xxxx_xxxx_xxxx_xx10_0000	8KB	[31:13]
24'bxxxx_xxxx_xxxx_xxxx_x100_0000	16KB	[31:14]
24'bxxxx_xxxx_xxxx_xxxx_1000_0000	32KB	[31:15]
24'bxxxx_xxxx_xxxx_xxx1_0000_0000	64KB	[31:16]
24'bxxxx_xxxx_xxxx_xx10_0000_0000	128KB	[31:17]

PeDRF_HTBAR[31:08]	Size	Writable bits
24'bxxxx_xxxx_xxxx_x100_0000_0000	256KB	[31:18]
24'bxxxx_xxxx_xxxx_1000_0000_0000	512KB	[31:19]
24'bxxxx_xxxx_xxx1_0000_0000_0000	1MB	[31:20]
24'bxxxx_xxxx_xx10_0000_0000_0000	2MB	[31:21]
24'bxxxx_xxxx_x100_0000_0000_0000	4MB	[31:22]
24'bxxxx_xxxx_1000_0000_0000_0000	8MB	[31:23]
24'bxxxx_xxx1_0000_0000_0000_0000	16MB	[31:24]
24'bxxxx_xx10_0000_0000_0000_0000	32MB	[31:25]
24'bxxxx_x100_0000_0000_0000_0000	64MB	[31:26]
24'bxxxx_1000_0000_0000_0000_0000	128MB	[31:27]
24'bxxx1_0000_0000_0000_0000_0000	256MB	[31:28]
24'bxx10_0000_0000_0000_0000_0000	512MB	[31:29]
24'bx100_0000_0000_0000_0000_0000	1GB	[31:30]
24'b1000_0000_0000_0000_0000_0000	2GB	[31]
24'b0000_0000_0000_0000_0000_0000	4GB	-

If PeDRF_HTBAR[0] is 1,

[07:01] RSVD - Reserved

[31:08] PeDRF_HTBA - Base Address

These bits initially contain the size of the requested PCI Memory space. The Root Complex reads this value, then assigns an address range by writing to this register. The writable bits of this field depends on the requested size.

25.3.1.2.2 PeDRF_HTSVSID: Subsystem Vendor ID & Subsystem ID Register *(EP)*

PCIe Offset: **0x102C**

Table 25.39: PeDRF_HTSVSID

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RW	RO	16'h192A	PeDRF_HTSVEID	Subsystem Vendor ID
[31:16]	RW	RO	16'h0007	PeDRF_HTSSYID	Subsystem ID

[15:00] SVID - Subsystem Vendor ID

This 16-bit register defines the Vendor ID for PeCORE. Subsystem Vendor IDs can be obtained from the PCI-SIG and are used to identify the vendor of the add-in card or subsystem. The Subsystem Vendor ID assigned by the PCI-SIG to BiTMICRO is 192Ah. This field is accessible only if the device is configured as an Endpoint.

[31:16] SID - Subsystem ID

This 16-bit register defines the Device ID for PeCORE. This field is the same with Device ID (PeDRF_PCDEVID).

Note: A company requires only one Vendor ID. That value can be used in either the Vendor ID field of configuration space (offset 00h) or the Subsystem Vendor ID field of the configuration space (offset 2Ch). It is used in the Vendor ID field if the company built the silicon. It is used in the Subsystem Vendor ID field if the company built the add-in card. If a company built both, the same values would be used in both fields.

25.3.1.2.3 PeDRF_HTERBAR: Expansion ROM Base Address Register *(EP)*

PCIe Offset: **0x1030**

Table 25.40: PeDRF_HTERBAR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_HTERDEN	Expansion ROM Decode Enable
[10:01]	RO	RO	10'h0	RSVD	Reserved
[31:11]	RO	RW	21'h00	PeDRF_HTERBAD	Expansion ROM Base Address

[00] PeDRF_HTERDEN - Expansion ROM Decode Enable

Table 25.41: PeDRF_HTERDEN Encoding

Value	Function
1'b0	Disable decode of the Expansion ROM within system memory address space
1'b1	Enable decode of the Expansion ROM within system memory address space.

[10:01] RSVD - Reserved

These bits are hardwired to 0.

[31:11] PeDRF_HTERBAD - Expansion ROM Base Address

These bits contain the base address of the function's expansion ROM mapped into the system memory.

25.3.1.2.4 PeDRF_HTNCAPP: Capabilities Pointer Register (EP)

PCIe Offset: **0x1034**

Table 25.42: PeDRF_HTNCAPP

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[01:00]	RO	RO	2'b00	RSVD	Reserved
[07:02]	RW	RW	6'h10	PeDRF_HTCAPPT	Capabilities Pointer
[31:08]	RO	RO	24'h0	RSVD	Reserved

[01:00] RSVD - Reserved

[07:02] PeDRF_HTCAPPT - Capabilities Pointer

This 8-bit (lower 2 bits always 0) register points to the next capability in configuration space. Note that the Capabilities List field of the Status and Command register indicates whether PeCORE implements this pointer or not. This register points to the first item in the list of capabilities which is the Power Management Capability Structure (0x1040).

[31:08] RSVD - Reserved

25.3.1.2.5 PeDRF_HTMMIIL: Interrupt Pin/Interrupt Line Register (EP)

Offset: **0x103C**

Table 25.43: PeDRF_HTMMIIL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RW	8'h00	PeDRF_HTINTLN	Interrupt Line
[15:08]	RO	RO	8'h01	PeDRF_HTINTPI	Interrupt Pin
[31:16]	RO	RO	16'h00	RSVD	Reserved

[07:00] PeDRF_HTINTLN - Interrupt Line

This register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST (Power-on Self Test) software will write the routing information into this register as it initializes and configures the system. This field is read/write for both PCIe host and local processor.

[15:08] PeDRF_HTINTPI - Interrupt Pin

This field indicates which PCI interrupt line PeCORE device uses. A value of 01h indicates INTA#. INTB#, INTC# and INTD# values make sense for multifunction device only.

Table 25.44: Interrupt Pin

Value	Interrupt Pin
8'h1	INTA#
8'h2	INTB#
8'h3	INTC#
8'h4	INTD#
others	reserved

[31:16] RSVD - Reserved

These bits are hardwired to 0.

25.3.1.3 PCI Header Type 1 Specific Configuration Space Registers *(RC)*

The following subsections details the registers of PCI 3.0 Type 1 Configuration Space header for PCIe devices. These registers are instantiated per function of Type 1 Configuration space implemented by the PCIe device.

25.3.1.3.1 PeDRF_HTBAR0/1: BAR0/1 Register *(RC)*

PCIe Offset: **0x1010**, **0x1014**

Table 25.45: PeDRF_HTBAR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RO	RO	1'b0	PeDRF_HTMI	Space Indicator
[02:01]	RW	RO	2'b00	PeDRF_HTTYP	Memory Type
[03]	RW	RO	1'b1	PeDRF_HTPREF	Prefetchable
[07:04]	RO	RO	4'h00	RSVD	Reserved
[31:08]	RW	RO	24'hfff000	PeDRF_HTBA	Base Address

BAR register defines a region of System PCI Memory space owned by Talino™ EDC. This 32-bit register is enabled by the Enable bit of PeDRF_PIMS register.

The mapping of address from PCI address to local address is controlled by PIM-BAR register pair.

[00] PeDRF_HTMI - Space Indicator

This bit is hardwired to 0. It indicates that PeDRF_HTBAR defines a memory address space.

Table 25.46: PeDRF_HTMI

	Space Indicator
1'b0	Memory Space
1'b1	IO Space

[02:01] PeDRF_HTTYP - Memory Type

This field defines the memory type.

Table 25.47: PeDRF_HTTYP encoding

Value	Function
2'b00	Base address is 32-bits, and may be set anywhere within the 32-bit range
2'b01	reserved
2'b10	Base address is 64-bits, and may be set anywhere within the 64-bit range. Pairing is limited to this: BAR0-BAR1 BAR2-BAR3 BAR4-BAR5
2'b11	reserved

[03] PeDRF_HTPREF - Prefetchable

This bit indicates if the region is prefetchable.

Table 25.48: PeDRF_HTPREF Encoding

Value	Function
1'b0	Memory is not prefetchable
1'b1	Memory is prefetchable

[07:04] RSVD - Reserved

[31:08] PeDRF_HTBA - Base Address

These bits initially contain the size of the requested PCI Memory space. The Root Complex reads this value, then assigns an address range by writing to this register. The writable bits of this field depends on the requested size.

Table 25.49: PeDRF_HTBA

PeDRF_HTBAR[31:08]	Size	Writable bits
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xxx1	256B	[31:08]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xx10	512B	[31:09]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_x100	1KB	[31:10]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_1000	2KB	[31:11]
24'bxxxx_xxxx_xxxx_xxxx_xxx1_000	4KB	[31:12]
24'bxxxx_xxxx_xxxx_xxxx_xx10_0000	8KB	[31:13]
24'bxxxx_xxxx_xxxx_xxxx_x100_0000	16KB	[31:14]
24'bxxxx_xxxx_xxxx_xxxx_1000_0000	32KB	[31:15]
24'bxxxx_xxxx_xxxx_xxx1_0000_0000	64KB	[31:16]
24'bxxxx_xxxx_xxxx_xx10_0000_0000	128KB	[31:17]

PeDRF_HTBAR[31:08]	Size	Writable bits
24'bxxxx_xxxx_xxxx_x100_0000_0000	256KB	[31:18]
24'bxxxx_xxxx_xxxx_1000_0000_0000	512KB	[31:19]
24'bxxxx_xxxx_xxx1_0000_0000_0000	1MB	[31:20]
24'bxxxx_xxxx_xx10_0000_0000_0000	2MB	[31:21]
24'bxxxx_xxxx_x100_0000_0000_0000	4MB	[31:22]
24'bxxxx_xxxx_1000_0000_0000_0000	8MB	[31:23]
24'bxxxx_xxx1_0000_0000_0000_0000	16MB	[31:24]
24'bxxxx_xx10_0000_0000_0000_0000	32MB	[31:25]
24'bxxxx_x100_0000_0000_0000_0000	64MB	[31:26]
24'bxxxx_1000_0000_0000_0000_0000	128MB	[31:27]
24'bxxx1_0000_0000_0000_0000_0000	256MB	[31:28]
24'bxx10_0000_0000_0000_0000_0000	512MB	[31:29]
24'bx100_0000_0000_0000_0000_0000	1GB	[31:30]
24'b1000_0000_0000_0000_0000_0000	2GB	[31]
24'b0000_0000_0000_0000_0000_0000	4GB	-

25.3.1.3.2 PeDRF_HTSSSPB: PCIe Bus Numbers Register (RC)

PCIe Offset: **0x1018**

Table 25.50: PeDRF_HTSSSPB

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RW	8'h00	PeDRF_HTPBNUM	Primary Bus Number
[15:08]	RW	RW	8'h00	PeDRF_HTSBNUM	Secondary Bus Number
[23:16]	RW	RW	8'h00	PeDRF_HTSUNUM	Subordinate Bus Number
[31:24]	RO	RO	8'b00	RSVD	Reserved

[07:00] PeDRF_HTPBNUM - Primary Bus Number

This field gives the bus number on which this function resides.

[15:08] PeDRF_HTSBNUM - Secondary Bus Number

This registers gives the bus number immediately to the south of this bridge function.

[23:16] PeDRF_HTSUNUM - Subordinate Bus Number

This register gives the highest bus number in the bus hierarchy south of this bridge function.

[31:24] RSVD - Reserved

These bits are hardwired to 0.

25.3.1.3.3 PeDRF_HTSIOLB: Secondary Status/IO Limit/IO Base Register *(RC)*

PCIe Offset: **0x101C**

Table 25.51: PeDRF_HTSIOLB

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_HTIOBAC	I/O Base Addressing Capability
[03:01]	RO	RO	3'b000	RSVD	Reserved
[07:04]	RW	RO	4'b0000	PeDRF_HTIOBAD	I/O Base Address
[08]	RW	RO	1'b0	PeDRF_HTIOLAC	I/O Limit Address Capability
[11:09]	RO	RO	3'b000	RSVD	Reserved
[15:12]	RW	RO	4'b0000	PeDRF_HTIOLAD	I/O Limit Address
[23:16]	RO	RO	8'h00	RSVD	Reserved
[24]	RW1C	RW1C	1'b0	PeDRF_HTMDPER	Master Data Parity Error
[26:25]	RO	RO	2'b00	RSVD	Reserved
[27]	RW1C	RW1C	1'b0	PeDRF_HTSTABO	Signaled Target Abort
[28]	RW1C	RW1C	1'b0	PeDRF_HTRTABO	Received Target Abort
[29]	RW1C	RW1C	1'b0	PeDRF_HTRMABO	Received Master Abort
[30]	RW1C	RW1C	1'b0	PeDRF_HTRSERR	Received System Error
[31]	RW1C	RW1C	1'b0	PeDRF_HTDPERR	Detected Parity Error

[00] PeDRF_HTIOBAC - I/O Base Addressing Capability

This field defines the addressing capability of the bridge.

Table 25.52: PeDRF_HTIOBAC Encoding

Value	Function
1'b0	16-bit I/O Addressing
1'b1	32-bit I/O Addressing

[03:01] RSVD - Reserved

These bits are hardwired to 0.

[07:04] PeDRF_HTIOBAD - I/O Base Address

This optional field defines the start address of I/O space if the bridge supports I/O space on its downstream side to recognize and pass through to the secondary bus.

[08] PeDRF_HTIOLAC - I/O Limit Addressing Capability

This field defines the addressing capability of the bridge.

Table 25.53: PeDRF_HTIOLAC Encoding

Value	Function
1'b0	16-bit I/O Addressing
1'b1	32-bit I/O Addressing

[11:09] RSVD - Reserved

These bits are hardwired to 0.

[15:08] PeDRF_HTIOLAD - I/O Limit Address

This optional field defines the end address of the I/O space range if the bridge supports I/O space on its downstream side to recognize and pass through to the secondary bus.

[23:16] RSVD - Reserved

[24] PeDRF_HTMDPER - Master Data Parity Error

This bit is set by the Secondary side Requestor if the Command Parity Error Response bit of the Status and Command register is set and either the Requestor receives a poisoned Completion or the Requestor poisons a write Request.

[26:25] RSVD - Reserved

These bits are hardwired to 0.

[27] PeDRF_HTSTABO - Signaled Target Abort

This bit is set when the Secondary side for Type 1 Configuration Space header device, for requests completed by the Type 1 header device, completes a Request using Completer Abort (CA) Completion Status.

[28] PeDRF_HTRTABO - Received Target Abort

This bit is set when the Secondary side for Type 1 Configuration Space header device, for requests initiated by the Type 1 header device, receives a Completion with Completer Abort (CA) Completion Status.

[29] PeDRF_HTRMABO - Received Master Abort

This bit is set when the Secondary Side for Type 1 Configuration Space header device, for requests initiated by the Type 1 header device, receives a Completion with Unsupported Request (UR) Completion Status.

[30] **PeDRF_HTRSERR - Received System Error**

This bit is set when a device sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR Enable bit in the Bridge Controler register is set.

[31] **PeDRF_HTDPERR - Detected Parity Error**

This bit is set by the Secondary Side for Type 1 Configuration Space header device whenever it receives a Poisoned TLP. Note that setting of this bit is not affected by the current value of the Command Parity Error Response bit of the Status and Command register.

25.3.1.3.4 PeDRF_HTMLMBA: Memory Limit/Memory Base Address Register (RC)

PCIe Offset: **0x1020**

Table 25.54: PeDRF_HTMLMBA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[03:00]	RO	RO	4'b0000	RSVD	Reserved
[15:04]	RW	RW	12'h00	PeDRF_HTMEMBA	Memory Base Address
[19:16]	RO	RO	4'b0000	RSVD	Reserved
[31:20]	RW	RW	12'h0000	PeDRF_HTMEMLA	Memory Limit Address

[03:00] RSVD - Reserved

These bits are hardwired to 0.

[15:04] PeDRF_HTMEMBA - Memory Base Address

This field defines the start address of memory mapped I/O address space to which the bridge would respond to on its primary bus. Any memory accesses within the specified range would be passed to the bridge's secondary PCI bus, while accesses outside this range are ignored by the bridge. This field is read/write for both PCI host and local processor.

[19:16] RSVD - Reserved

These bits are hardwired to 0.

[31:16] PeDRF_HTMEMLA - Memory Limit Address

This field defines the end address of memory mapped I/O address space to which the bridge would respond to on its primary bus. Any memory accesses within the specified range would be passed to the bridge's secondary PCI bus, while accesses outside this range are ignored by the bridge. This field is read/write for both PCI host and local processor.

25.3.1.3.5 PeDRF_HTPMLMB: Prefetchable Memory Register (RC)

PCIe Offset: **0x1024**

Table 25.55: PeDRF_HTPMLMB

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b1	PeDRF_HTPMBAC	Prefetchable Memory Base Addressing Capability
[03:01]	RO	RO	3'b000	RSVD	Reserved
[15:04]	RW	RO	12'h000	PeDRF_HTPMBAD	Prefetchable Memory Base Address
[16]	RW	RO	1'b1	PeDRF_HTPMLAC	Prefetchable Memory Limit Addressing Capability
[19:17]	RO	RO	3'b000	RSVD	Reserved
[31:20]	RW	RO	12'h000	PeDRF_HTPMLAD	Prefetchable Memory Limit Address

[00] PeDRF_HTPMBAC - Prefetchable Memory Base Addressing Capability

Table 25.56: PeDRF_HTPMBAC Encoding

Value	Function
1'b0	32-bit Memory Addressing
1'b1	64-bit Memory Addressing

[03:01] RSVD - Reserved

[15:04] PeDRF_HTPMBAD - Prefetchable Memory Base Address

This optional field defines the start address of a prefetchable memory address space to which the bridge would respond to on its primary bus. Any memory accesses within the specified range would be passed to the bridge's secondary PCI bus, while accesses outside this range are ignored by the bridge. This field is read/write for both PCI host and local processor.

[16] PeDRF_HTPMLAC - Prefetchable Memory Limit Addressing Capability

Table 25.57: PeDRF_HTPMLAC Encoding

Value	Function
1'b0	32-bit Memory Addressing
1'b1	64-bit Memory Addressing

[19:17] **RSVD - Reserved**

[31:20] **PeDRF_HTPMLAD - Prefetchable Memory Limit Address**

This optional field defines the end address of a prefetchable memory address space to which the bridge would respond to on its primary bus. Any memory accesses within the specified range would be passed to the bridge's secondary PCI bus, while accesses outside this range are ignored by the bridge. This field is read/write for both PCI host and local processor.

25.3.1.3.6 PeDRF_HTPBU32: Prefetchable Base Upper 32-bit Register *(RC)*

PCIe Offset: **0x1028**

Table 25.58: PeDRF_HTPBU32

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RO	32'h00	PeDRF_HTPBU32	Prefetchable Base Upper 32-bit Address

[31:00] PeDRF_HTPBU32 - Prefetchable Base Upper 32-bit Address

This optional field is an extension of the Prefetchable Memory Base Address field if 64-bit addressing for prefetchable memory access is supported.

25.3.1.3.7 PeDRF_HTPLU32: Prefetchable Limit Upper 32-bit Register *(RC)*

PCIe Offset: **0x102C**

Table 25.59: PeDRF_HTPLU32

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RO	32'h00	PeDRF_HTPLU32	Prefetchable Base Upper 32-bit Address

[31:00] PeDRF_HTPLU32 - Prefetchable Limit Upper 32-bit Address

This optional field is an extension of the Prefetchable Memory Limit Address field if 64-bit addressing for prefetchable memory access is supported.

25.3.1.3.8 PeDRF_HTIOLBU: IO Limit Upper 16-bit/IO Base Upper 16-bit Register (RC)

PCIe Offset: **0x1030**

Table 25.60: PeDRF_HTIOLBU

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RW	RO	16'h0000	PeDRF_HTIOBU6	I/O Base Upper 16-bit Address
[31:16]	RW	RO	16'h0000	PeDRF_HTIOLU6	I/O Limit Upper 16-bits Address

[15:00] PeDRF_HTIOBU6 - I/O Base Upper 16-bit Address

This optional field is an extension of the I/O Base Address field of the Secondary Status/IO Limit/IO Base register if 32-bit addressing for I/O access is supported.

[31:16] PeDRF_HTIOLU6 - I/O Limit Upper 16-bit Address

This optional field is an extension of the I/O Limit Address field of the Secondary Status/IO Limit/IO Base register if 32-bit addressing for I/O access is supported.

25.3.1.3.9 PeDRF_HTNCAPP: Capabilities Pointer Register (RC)

PCIe Offset: **0x1034**

Table 25.61: PeDRF_HTNCAPP

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RO	8'h40	PeDRF_HTCAPPT	Capabilities Pointer
[31:08]	RO	RO	24'h0	RSVD	Reserved

[07:00] PeDRF_HTCAPPT - Capabilities Pointer

This 8-bit register points to the next capability in configuration space. Note that the Capabilities List field of the Status and Command register indicates whether PeCORE implements this pointer or not. This register points to the first item in the list of capabilities which points to the Power Management Capability Structure (0x1040).

[31:08] RSVD - Reserved

25.3.1.3.10PeDRF_HTERBAR: Expansion ROM Base Address Register (RC)

PCIe Offset: **0x1038**

Table 25.62: PeDRF_HTERBAR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_HTERDEN	Expansion ROM Decode Enable
[10:01]	RO	RO	1'b0	RSVD	Reserved
[31:11]	RO	RW	21'h00	PeDRF_HTERBAD	Expansion ROM Base Address

[00] PeDRF_HTERDEN - Expansion ROM Decode Enable

Table 25.63: PeDRF_HTPRFE2 Encoding

Value	Function
1'b0	Disable decode of the Expansion ROM within system memory address space
1'b1	Enable decode of the Expansion ROM within system memory address space.

[10:1] RSVD - Reserved

These bits are hardwired to 0.

[31:11] PeDRF_HTERBAD - Expansion ROM Base Address

These bits describe the base address of the function's expansion ROM mapped into system memory.

25.3.1.3.11PeDRF_HTBCIIL: Bridge Control/Interrupt Pin/ Interrupt Line Register *(RC)*

PCIe Offset: **0x103C**

Table 25.64: PeDRF_HTBCIIL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RW	8'h00	PeDRF_HTINTLN	Interrupt Line
[15:08]	RO	RO	8'h01	PeDRF_HTINTPI	Interrupt Pin
[16]	RW	RW	1'b0	PeDRF_HTPEREN	Parity Error Response Enable
[17]	RW	RW	1'b0	PeDRF_HTSEREN	SERR# Enable
[21:18]	RO	RO	4'h0	RSVD	Reserved
[22]	RW	RW	1'b0	PeDRF_HTSBRST	Secondary Bus Reset
[31:23]	RO	RO	9'h00	RSVD	Reserved

[07:00] PeDRF_HTINTLN - Interrupt Line

This register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST (Power-on Self Test) software will write the routing information into this register as it initializes and configures the system. This field is read/write for both PCI host and local processor.

[15:08] PeDRF_HTINTPI - Interrupt Pin

This field indicates which PCI interrupt line PeCORE device uses. This 8-bit register is read-only and the value 01h indicates INTA#.

[16] PeDRF_HTPEREN - Parity Error Response Enable

This bit controls the device's response to Poisoned TLP's.

Table 25.65: PeDRF_HTPEREN Encoding

Value	Function
1'b0	Ignore parity errors ont he secondary interface
1'b1	Enable parity error detection and reporting on the secondary interface

[17] PeDRF_HTSEREN - SERR# Enable

This bit controls the forwarding of ERR_COR, ERR_NONFATAL, and ERR_FATAL Messages from the secondary to primary side of the bridge.

Table 25.66: PeDRF_HTSEREN Encoding

Value	Function
1'b0	Do not forward errors to primary interface
1'b1	Forward errors to the primary interface

[21:18] RSVD - Reserved

These bits are hardwired to 0.

[22] PeDRF_HTSBRST - Secondary Bus Reset

Setting this bit triggers a warn reset on the corresponding PCIe port and the PCIe hierarchy domain subordinate to the port.

[31:23] RSVD - Reserved

These bits are hardwired to 0.

25.3.1.4 PCI Power Management Capability Registers

25.3.1.4.1 PeDRF_PMCPHDR: Power Management Capability Header Register

PCIe Offset: **0x1040**

Table 25.67: PeDRF_PMCPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RO	RO	8'h01	PeDRF_PMCAPID	PM Capabilities Identifier
[15:08]	RW	RO	8'h48	PeDRF_PMNCAPP	Next Capability Pointer
[18:16]	RO	RO	3'b011	PeDRF_PMVERSI	PM Capabilities Version
[19]	RO	RO	1'b0	PeDRF_PMPMECLK	PME Clock
[20]	RO	RO	2'b00	RSVD	Reserved
[21]	RO	RO	1'b0	PeDRF_PMDSINI	Device Specific Initialization
[24:22]	RO	RO	3'b000	PeDRF_PMAUXCU	Aux Current
[25]	RW	RO	1'b1	PeDRF_PMDISUP	D1 PM State Support
[26]	RW	RO	1'b1	PeDRF_PMD2SUP	D2 PM State Support
[31:27]	RW	RO	5'h19	PeDRF_PMPMESU	PME Support

[07:00] PeDRF_PMCAPID - Power Management Capabilites Identifier

This field provides information on the capabilities of the function related to power management. This field is read 01h to indicate that PeCORE supports power management and that the data structure currently being pointed to is the PCI power management capability structure. This field is read-only for both PCI host and local processor.

[15:08] PeDRF_PMNCAPP - Next Capability Pointer

This field describes the location of the next item in the function's capability list. The default value of this field points to the next capability structure. If there is no next capability structure, this field should be 00h to indicate the end of capability list of PeCORE. In this case, this register points to MSI Capability Header Register (0x1048).

[18:16] PeDRF_PMVERSI - PM Version

This field indicates the PCI Power Management version that PeCORE supports. This field returns a value of 011b when read, indicating that this function complies with the *PCI Bus Power Management Interface Specification,* version 1.2. This field is read-only for both PCI host and local processor.

[19] **PeDRF_PMPMECLK - PME Clock**

Does not apply to PCI Express and must be hardwired to 0.

[20] **RSVD - Reserved**

These bits are hardwired to 0.

[21] **PeDRF_PMDSINI - PM Device Specific Initialization**

This bit indicates whether a special initialization, beyond the standard PCI configuration header, of this function is required before the generic class driver is able to use it.

[24:22] **PeDRF_PMAUXCU - PM Aux Current**

This field indicates the 3.3Vaux auxiliary current requirements for the PCI function. If the *PM Data* field of the PM Control Status (PeDRF_PMCSDAT) is implemented or if PME# generation from D3cold is not supported (bit 31 = 0), then this field must return 0 when read.

Table 25.68: PeDRF_PMAUXCU Encoding

Value	Function
3'b111	Max. Current Required = 375 mA
3'b110	Max. Current Required = 320 mA
3'b101	Max. Current Required = 270 mA
3'b100	Max. Current Required = 220 mA
3'b011	Max. Current Required = 160 mA
3'b010	Max. Current Required = 100 mA
3'b001	Max. Current Required = 55 mA
3'b000	Max. Current Required = 0 (self-powered)

[25] **PeDRF_PMD1SUP - PM D1 Support**

This bit indicates if the function of the PCIe device supports the D1 Power Management State.

[26] **PeDRF_PMD2SUP - PM D2 Support**

This bit indicates if the function of the PCIe device supports the D2 Power Management State.

[31:27] **PeDRF_PMPMESU - PME Support**

This field indicates the power states in which the device may generate a PME. Note that bits 31, 30, and 27 must always be set to 1 for PCI-PCI Bridge structures representing Ports on Root Complexes/Switches to indicate that the bridge will forward PME Messages.

Table 25.69: PeDRF_PMPMESU Encoding

Value	Function
5'bxxxx1	PME# can be asserted from D0 state
5'bxxx1x	PME# can be asserted from D1 state
5'bxx1xx	PME# can be asserted from D2 state
5'bx1xxx	PME# can be asserted from D3hot state
5'b1xxxx	PME# can be asserted from D3cold state

25.3.1.4.2 PeDRF_PMCSDAT: PM Control Status and Data Register

PCIe Offset: **0x1044**

Table 25.70: PeDRF_PMCSDAT

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[01:00]	RW	RW	2'b00	PeDRF_PMPSTAT	Power State
[02]	RO	RO	1'b0	RSVD	Reserved
[03]	RW	RO	1'b0	PeDRF_PMNSRES	No Soft Reset
[07:04]	RO	RO	4'b0000	RSVD	Reserved
[08]	RW	RW	1'b0	PeDRF_PMEENAB	PME Enable
[12:09]	RW	RW	4'b0000	PeDRF_PMDSELE	Data Select
[14:13]	RO	RO	2'b00	PeDRF_PMDSCAL	Data Scale
[15]	RW1C	RW1C	1'b0	PeDRF_PMESTAT	PME Status
[21:16]	RO	RO	6'h00	RSVD	Reserved
[22]	RW	RO	1'b0	PeDRF_PMB2B3S	B2/B3 Support
[23]	RW	RO	1'b0	PeDRF_PMBPCCE	Bus Power/Clock Control Enable
[31:24]	RO	RO	8'h00	PeDRF_PMDATAX	PM Data

[01:00] PeDRF_PMPSTAT - Power State

This field is used to record the current power state of a function of PeCORE. This field is also used to set the function into a new power state. Note that the value of this field does not change if software attempts to write an unsupported, optional state to this field. Writing to this register causes the Power Management State Change Request register to change. This field is read/write for both PCI host and local processor.

Table 25.71: PeDRF_PMPSTAT Encoding

Value	Function
2'b00	D0
2'b01	D1
2'b10	D2
2'b11	D3-hot

[02] RSVD - Reserved

[03] PeDRF_PMNSRES - No Soft Reset

This bit indicates that devices transitioning from D3hot to D0 because of *Power State* commands do not perform an internal reset. When 0, this bit indicates that devices do perform an internal reset upon the transition via software control of the *Power State* bits.

[07:04] RSVD - Reserved

[08] **PeDRF_PMEENAB - PME Enable**

This bit enables the function to assert PME#. This bit defaults to 0 if the function does not support PME# generation from D3cold.

[12:09] **PeDRF_PMDSELE - Data Select**

This field indicates which data is to be reported through the *PM Data* and *Data Scale* fields. Note that this field must be implemented if *PM Data* is implemented.

[14:13] **PeDRF_PMDSCAL - Data Scale**

This field indicates the scaling factor to be used when interpreting the value of the *PM Data* field. Note that this field must be implemented if *PM Data* is implemented.

[15] **PeDRF_PMESTAT - PME Status**

This bit is set when the function would normally assert the PME# signal independent of the value of the *PME Enable* bit.

[21:16] **RSVD - Reserved**

[22] **PeDRF_PMB2B3S - B2/B3 Support**

This bit determines the action that is to occur as a direct result of programming the function of D3hot. When 1, the bridge function's secondary bus' PCI clock will be stopped when the function is programmed to D3hot. When 0, the bridge function's secondary bus will have its power removed when the function is programmed to D3hot.

[23] **PeDRF_PMBPCCE - Bus Power/Clock Control Enable**

This bit enables the bus power/clock control mechanism defined in the *PCI Local Bus Specification*, Revision 3.0 Section 4.7.1 is enabled. Note that when the Bus Power/Clock Control mechanism is disabled, the bridge's PeDRF_PMCSDAT Power State field cannot be used by the system software to control the power or clock the bridge's secondary bus.

[31:24] **PeDRF_PMDATAX - Data**

This optional field is used to report the state dependent data requested by the Data Select field. The value of this field is scaled by the value reported by the Data Scale field.

Table 25.72: Power Consumption/Dissipation Reporting

DSEL	Data Reported	Data Scale	Units/Accuracy
0	D0 Power Consumed		
1	D1 Power Consumed		
2	D2 Power Consumed	0 = unknown	
3	D3 Power Consumed	1 = 0.1x	watts
4	D0 Power Dissipated	2 = 0.01x	
5	D1 Power Dissipated	3 = 0.001x	
6	D2 Power Dissipated		
7	D3 Power Dissipated		
8	Common logic power consumption		
8-15	reserved	reserved	TBD

25.3.1.5 MSI Capability Registers

25.3.1.5.1 PeDRF_MSCPHDR: MSI Capability Header Register

PCIe Offset: **0x1048**

Table 25.73: PeDRF_MSCPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RO	RO	8'h05	PeDRF_MSCAPID	MSI Capability Identifier
[15:08]	RW	RO	8'h60	PeDRF_MSNCAPP	Next Capability Pointer
[16]	RW	RW	1'b0	PeDRF_MSIENAN	MSI Enable
[19:17]	RW	RO	3'b101	PeDRF_MSMMCAP	Multiple Message Capable
[22:20]	RW	RW	3'b000	PeDRF_MSMMENA	Multiple Message Enable
[23]	RW	RO	1'b1	PeDRF_MS64ADD	64bit Address Capable
[24]	RW	RO	1'b1	PeDRF_MSPVMCA	Per-Vector Masking Capable
[31:25]	RO	RO	7'h00	RSVD	Reserved

The PeDRF_MSCPHDR register is divided into MSI Capability Identifier, Next Item Pointer, and Message Control registers. Details on these registers are discussed below.

[07:00] PeDRF_MSCAPID - MSI Capability Identifier

This 8-bit register reads a value of 05h to indicate an MSI capability structure.

[15:08] PeDRF_MSNCAPP - Next Capability Pointer

This 8-bit register points to the next item in the device's capability list. In this case, this register points to PCIe Capability Header Register (0x1060).

[16] PeDRF_MSIENAN - Message Control: MSI Enable

This bit enables/disables the MSI capability of PeCORE. When 1, the MSI capability is enabled and the bridge is prohibited from using INTx# pin (if implemented). When 0, the MSI capability is disabled. Note that this bit must be cleared to 0 after reset to maintain backward compatibility.

[19:17] PeDRF_MSMMCAP - Message Control: Multiple Message Capable

This field indicates the number of messages requested by PeCORE. Note that this field is written by the system software. PeCORE requests 32 messages, thus this field is hardwired to 001b.

Table 25.74: PeDRF_MSMMCAP Encoding

Value	Function
3'b000	1 message requested
3'b001	2 messages requested
3'b010	4 messages requested
3'b011	8 messages requested
3'b100	16 messages requested
3'b101	32 messages requested
3'b110	reserved
3'b111	reserved

[22:20] PeDRF_MSMMENA - Message Control: Multiple Message Enable

This field indicates the number of allocated messages for PeCORE. System software writes to this field and the number of allocated messages is aligned to a power of 2.

Table 25.75: PeDRF_MSMMENA Encoding

Value	Function
3'b000	1 message allocated
3'b001	2 messages allocated
3'b010	4 messages allocated
3'b011	8 messages allocated
3'b100	16 messages allocated
3'b101	32 messages allocated
3'b110	reserved
3'b111	reserved

[23] PeDRF_MS64ADD - Message Control: 64-bit Address Capable

This bit indicates that PeCORE is capable of generating 64-bit message addresses thus this bit is hardwired to 1.

[24] PeDRF_MSPVMCA - Per-Vector Masking Capable

This bit, when set, indicates that PCIe device supports MSI per-vector masking.

[31:24] RSVD - Reserved

25.3.1.5.2 PeDRF_MSIMADD: MSI Message Address Register

PCIe Offset: **0x104C**

Table 25.76: PeDRF_MSIMADD

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[01:00]	RO	RO	2'b0	RSVD	Reserved
[31:02]	RW	RW	30'h00	PeDRF_MSIMADX	MSI Message Address

[01:00] RSVD - Reserved

[31:02] PeDRF_MSIMADX - MSI Message Address

This field indicates specific message address. If the Message Enable bit of the MSI Capability Identifier\Next Item Pointer\Message Control register is set, the contents of this register specify the 4-byte aligned address for the MSI memory write transaction. Address bits 1 to 0 are driven to zero during the address phase.

25.3.1.5.3 PeDRF_MSMUADD: MSI Message Upper Address Register

PCIe Offset: **0x1050**

Table 25.77: PeDRF_MSMUADD

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RW	32'h00	PeDRF_MSMUADD	MSI Message Upper Address

[31:00] PeDRF_MSMUADD - MSI Message Upper Address

This register is optional and only implemented when the device supports a 64-bit message address. This register specifies the upper 32-bits of a 64-bit message address. This register is enabled using the Message Control: 64-bit Address Capable bit of the MSI Capability Identifier\Next Item Pointer\Message Control register. PeCORE uses the 32-bit address specified by the MSI Message Address register if the contents of this register are zero. Note that this field is configured by the system.

25.3.1.5.4 PeDRF_MSMEIMD: Message Data Register

PCIe Offset: **0x1054**

Table 25.78: PeDRF_MSMEIMD

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RW	RW	16'h00	PeDRF_MSMDATA	Message Data
[31:16]	RO	RO	16'h00	RSVD	Reserved

[15:00] PeDRF_MSMDATA - Message Data

This 16-bit register indicates a system-specific message. The value of this register is written to the message address when PeCORE issues an MSI message if the Message Enable bit of the PeDRF_MSCPHDR register is 1. The least significant bit of this register selects which of the two messages is being reported if in case PeCORE is allocated with two messages by the system. If only one message is allocated, then PeCORE cannot modify this data on a MSI write. This field is read/write for both PCI host and local processor.

[31:16] RSVD - Reserved

25.3.1.5.5 PeDRF_MSMASKX: MSI Mask Register

PCIe Offset: **0x1058**

Table 25.79: PeDRF_MSMASKX

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RW	32'h0	PeDRF_MSMASKX	MSI Mask

[31:00] PeDRF_MSMASKX - MSI Mask

This register is optional and is used by software to disable message sending on a per-vector basis. For each Mask bit that is set, the function is prohibited from sending the associated message.

MSI vectors are numbered 0 through N-1, where N is the number of vectors allocated by software. Each vector is associated with a correspondingly numbered bit in the MSI Mask register. The Multiple Message Capable field indicates how many vectors (with associated Mask bits) are implemented. All unimplemented Mask bits are reserved with a value of 0.

25.3.1.5.6 PeDRF_MSPENDI: MSI Pending Register

PCIe Offset: **0x105C**

Table 25.80: PeDRF_MSPENDI

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RO	32'h0	PeDRF_MSPENDI	MSI Pending

[31:00] PeDRF_MSPENDI - MSI Pending

This register is optional and is used by software to defer message sending on a per-vector basis. For each Pending bit that is set, the function has a pending associated message.

MSI vectors are numbered 0 through N-1, where N is the number of vectors allocated by software. Each vector is associated with a correspondingly numbered bit in the MSI Pending register. The Multiple Message Capable field indicates how many vectors (with associated Pending bits) are implemented. All unimplemented Pending bits are reserved with a value of 0.

25.3.1.6 PCIe Capability Registers

These registers are instantiated per function implemented by the PCIe device.

25.3.1.6.1 PeDRF_PECPHDR: PCIe Capability Header Register

PCIe Offset: **0x1060**

Table 25.81: PeDRF_PECPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RO	RO	8'h10	PeDRF_PECAPID	Capability Identifier
[15:08]	RW	RO	8'h00	PeDRF_PENCAPP	Next Capability Pointer
[19:16]	RO	RO	4'b0010/ 4'b0001	PeDRF_PECAPVE	Capability Version
[23:20]	RW	RO	4'b0000	PeDRF_PEDPTYP	Device/Port Type
[24]	RW	RO	1'b0	PeDRF_PESIMPL	Slot Implemented
[29:25]	RW	RO	5'h00	PeDRF_PEIMNUM	Interrupt Message Number
[31:30]	RO	RO	2'b00	RSVD	Reserved

[07:00] PeDRF_PECAPID - Capability Identifier

This field indicates the PCIe Capability structure and returns 10h when read.

[15:08] PeDRF_PENCAPP - Next Item Pointer

This field is a pointer to the next item in the capability list function of PeCORE. This field is 0 if no items exist in the linked list of capabilities.

[19:16] PeDRF_PECAPVE - Capability Version

This field indicates the PCI-SIG defined PCIe capability structure version number. If device is Gen2 compliant, the value of this field is 4'h2. Otherwise, the value of this field is 4'h1.

[23:20] PeDRF_PEDPTYP - Device/Port Type

This field indicates the type of PCIe logical device.

Table 25.82: PeDRF_PEDPTYP Encoding

Value	Function
4'b0000	PCIe Endpoint device
4'b0001	Legacy PCIe Endpoint device
4'b0100	Root Port of PCIe Root Complex (Type 1)
4'b0101	Upstream Port of PCIe Switch (Type 1)
4'b0110	Downstream Port of PCIe Switch (Type 1)

Table 25.82: PeDRF_PEDPTYP Encoding

Value	Function
4'b0111	PCIe -to-PCI/PCI-X Bridge (Type 1)
4'b1000	PCI/PCI-X to PCIe Bridge (Type 1)
4'b1001	Root Complex Integrated Endpoint device
4'b1010	Root Complex Event Collector
Other Encoding	reserved

[24] PeDRF_PESIMPL - Slot Implemented

This bit indicates that the PCIe Link associated with this Port is connected to a slot (as compared to being connected to an integrated component or being disabled). Note that this bit is valid for Root Port of PCIe Root Complex or Downstream Port of PCIe Switch. This bit is hardware initialized.

[29:25] PeDRF_PEIMNUM - Interrupt Message Number

This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with the status bits in either the Slot Status register or Root Status register. For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated. For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message. If both MSI and MSI-X is implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. However if the software enables both, the value in this register is undefined.

[31:30] RSVD - Reserved

25.3.1.6.2 PeDRF_PEDCAPA: Device Capabilities Register

PCIe Offset: **0x1064**

Table 25.83: PeDRF_PEDCAPA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[02:00]	RW	RO	3'b100	PeDRF_PEMXPSS	Maximum Payload Size Supported
[04:03]	RW	RO	2'b00	PeDRF_PEPFSUP	Phantom Functions Supported
[05]	RW	RO	1'b1	PeDRF_PEETFSU	Extended Tag Field Supported
[08:06]	RW	RO	3'b000/ 3'b111	PeDRF_PEELSAL	Endpoint L0s Acceptable Latency
[11:09]	RW	RO	3'b000/ 3'b111	PeDRF_PEEL1AL	Endpoint L1 Acceptable Latency
[14:12]	RO	RO	3'b000	RSVD	Reserved
[15]	RW	RO	1'b1	PeDRF_PERBERE	Role-Based Error Reporting
[17:16]	RO	RO	2'b0	RSVD	Reserved
[25:18]	RW	RO	8'b0	PeDRF_PECSPLV	Captured Slot Power Limit Value
[27:26]	RW	RO	2'b0	PeDRF_PECSPLS	Captured Slot Power Limit Scale
[31:28]	RO	RO	4'b0	RSVD	Reserved

[02:00] PeDRF_PEMXPSS - Maximum Payload Size Supported

This field indicates the maximum payload size that the device/function can support for TLPs.

Table 25.84: PeDRF_PEMXPSS Encoding

Value	Function
3'b000	Max. Payload Size = 128 bytes
3'b001	Max. Payload Size = 256 bytes
3'b010	Max. Payload Size = 512 bytes
3'b011	Max. Payload Size = 1024 bytes
3'b100	Max. Payload Size = 2048 bytes
3'b101	Max. Payload Size = 4096 bytes
3'b110	reserved
3'b111	reserved

[04:03] PeDRF_PEPFSUP - Phantom Functions Supported

This field indicates the support for use of unclaimed function numbers to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with Tag ID.

Table 25.85: PeDRF_PEPFSUP Encoding

Value	Function
2'b00	No function number bits used for Phantom Functions; device may implement all function numbers

Table 25.85: PeDRF_PEPFSUP Encoding

Value	Function
2'b01	Functions 0, 1, 2, and 3 may claim functions 4, 5, 6, and 7 as Phantom Functions respectively.
2'b10	Function 0 may claim functions 2,4 and 6 as Phantom Functions. Function 1 may claim functions 3, 5, and 7 as Phantom Functions.
2'b11	Device must be a single function 0 device that may claim all other functions as Phantom Functions.

Note that Phantom Function support for the device must be enabled by the corresponding control field in the Device Status and Control register.

[05] PeDRF_PEETFSU - Extended Tag Field Supported

This bit indicates the maximum supported size of the Tag field as a Requester. When 0, 5-bit Tag field is supported. When 1, 8-bit Tag field is supported. Note that the 8-bit Tag field support must be enabled by the corresponding control field in the Device Status and Control register.

[08:06] PeDRF_PEELSAL - Endpoint L0s Acceptable Latency

This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state.

Table 25.86: PeDRF_PEELSAL Encoding

Value	Function
3'b000	Maximum of 64 ns
3'b001	Maximum of 128 ns
3'b010	Maximum of 256 ns
3'b011	Maximum of 512 ns
3'b100	Maximum of 1 us
3'b101	Maximum of 2 us
3'b110	Maximum of 4 us
3'b111	No limit

Note that for devices other than Endpoints, this field is reserved and must be 0. Hence during Root Complex mode, this field is set to 3'b000. While during Endpoint mode, this field is set 3'b111.

[11:09] PeDRF_PEEL1AL - Endpoint L1 Acceptable Latency

This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L1 state to the L0 state.

Table 25.87: PeDRF_PEEL1AL Encoding

Value	Function
3'b000	Maximum of 1 us
3'b001	Maximum of 2 us

Table 25.87: PeDRF_PEEL1AL Encoding

Value	Function
3'b010	Maximum of 4 us
3'b011	Maximum of 8 us
3'b100	Maximum of 16 us
3'b101	Maximum of 32 us
3'b110	Maximum of 64 us
3'b111	No limit

Note that for devices other than Endpoints, this field is reserved and must be 0. Hence during Root Complex mode, this field is set to 3'b000. While during Endpoint mode, this field is set 3'b111.

[14:12] RSVD - Reserved

The field is undefined. In the previous PCI Express specification, this field was used to indicate Attention Button, Attention Indicator, and Power Indicator are implemented on the adapter. System software must ignore the value read from this field but is permitted to write any value. This field is read-only for both PCI host and local processor.

[15] PeDRF_PERBERE - Role-Based Error Reporting

This bit indicates that the device implements the functionality originally defined in the Error Reporting ECN for *PCI Express Base Specification,* Revision 1.0a, and later incorporated into the Revision 1.1 of the same specification.

[17:16] RSVD - Reserved

[25:18] PeDRF_PECSPLV - Captured Slot Power Limit Value

This field, together with the Captured Slot Power Limit Scale field, specifies the upper limit on power supplied by slot. The power limit, in watts unit, is calculated by multiplying the value of this field to the value of the PeDRF_PECSPLS. This value is set by the Set_Slot_Power_Limit Message or hardwired to 0.

Note that this field is used for Upstream Ports only.

[27:26] PeDRF_PECSPLS - Captured Slot Power Limit Scale

This field specifis the scale used for Slot Power Limit Value. This value is set by the Set_Slot_Power_Limit Message or hardwired to 0.

Table 25.88: PeDRF_PECSPLS Encoding

Value	Function
2'b00	1.0x
2'b01	0.1x
2'b10	0.01x

Table 25.88: PeDRF_PECSPLS Encoding

Value	Function
2'b11	0.001x

Note that this field is used for Upstream Ports only.

[31:28] **RSVD - Reserved**

25.3.1.6.3 PeDRF_PEDSCTL: Device Status and Control Register

PCIe Offset: **0x1068**

Table 25.89: PeDRF_PEDSCTL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_PECEREN	Correctable Error Reporting Enable
[01]	RW	RW	1'b0	PeDRF_PENFERE	Non-fatal Error Reporting Enable
[02]	RW	RW	1'b0	PeDRF_PEFEREN	Fatal Error Reporting Enable
[03]	RW	RW	1'b0	PeDRF_PEURREN	Unsupported Request Reporting Enable
[04]	RW	RW	1'b1	PeDRF_PEERORD	Enable Relaxed Ordering
[07:05]	RW	RW	3'b0	PeDRF_PEMXPSZ	Max_Payload_Size
[08]	RW	RW	1'b0	PeDRF_PEETFEN	Extended Tag Field Enable
[09]	RW	RW	1'b0	PeDRF_PEPHFEN	Phantom Functions Enable
[10]	RWS	RWS	1'b0	PeDRF_PEAPPME	Aux Power PM Enable
[11]	RW	RW	1'b1	PeDRF_PEENSNO	Enable Snoop Not Required
[14:12]	RW	RW	3'b010	PeDRF_PEMXRRS	Max_Read_Request_Size
[15]	RO	RO	1'b0	RSVD	Reserved
[16]	RWC1	RWC1	1'b0	PeDRF_PECEDET	Correctable Error Detected
[17]	RWC1	RWC1	1'b0	PeDRF_PENFEDE	Non-fatal Error Detected
[18]	RWC1	RWC1	1'b0	PeDRF_PEFEDET	Fatal Error Detected
[19]	RWC1	RWC1	1'b0	PeDRF_PEURDET	Unsupported Request Detected
[20]	RO	RO	1'b0	PeDRF_PEAPDET	AUX Power Detected
[21]	RO	RO	1'b0	PeDRF_PETPEND	Transaction Pending
[31:22]	RO	RO	10'h00	RSVD	Reserved

[00] PeDRF_PECEREN - Correctable Error Reporting Enable

This bit, together with other bits, controls the sending of ERR_COR Messages. For multi-function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of correctable errors is internal to the root and no external ERR_COR Message is generated.

[01] PeDRF_PENFERE - Non-fatal Error Reporting Enable

This bit, together with other bits, controls the sending of ERR_NONFATAL Messages. For multi-function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root Port, the reporting of correctable errors is internal to the root and no external ERR_NONFATAL Message is generated.

[02] PeDRF_PEFEREN - Fatal Error Reporting Enable

This bit, together with other bits, controls the sending of ERR_FATAL Messages. For multi-function device, this bit controls error reporting for each function from point-of-view of the respective function. For a Root

Port, the reporting of correctable errors is internal to the root and no external ERR_FATAL Message is generated.

[03] PeDRF_PEURREN - Unsupported Request Reporting Enable

This bit, together with other bits, controls the signalling of Unsupported Requests by sending of Error Messages. For multi-function device, this bit controls error reporting for each function from point-of-view of the respective function.

[04] PeDRF_PEERORD - Enable Relaxed Ordering

This bit enables the setting of the Relaxed Ordering bit in the Attributes field of transactions the device initiates that do not require strong ordering. This bit may be hardwired to 0 if a device never sets the RO attribute.

[07:05] PeDRF_PEMXPSZ - Max_Payload_Size

This field sets the maximum TLP payload size of the device or function. As a receiver, the device must handle TLPs as large as the set value. As a transmitter, the device must not generate TLPs exceeding the set value. This field is read/write for both PCI host and local processor.

Table 25.90: PeDRF_PEMXPSZ Encoding

Value	Function
3'b000	Maximum Payload Size = 128 bytes
3'b001	Maximum Payload Size = 256 bytes
3'b010	Maximum Payload Size = 512 bytes
3'b011	Maximum Payload Size = 1024 bytes
3'b100	Maximum Payload Size = 2048 bytes
3'b101	Maximum Payload Size = 4096 bytes
3'b110	Reserved
3'b111	Reserved

[08] PeDRF_PEETFEN - Extended Tag Field Enable

This bit enables the device or function to use an 8-bit Tag field as a requester. If this bit is 0, the device is restricted to a 5-bit Tag field.

[09] PeDRF_PEPHFEN - Phantom Functions Enable

This bit enables a device or function to use unclaimed functions a Phantom Functions to extend the number of outstanding transaction identifiers.

[10] PeDRF_PEAPPME - Auxiliary Power PM Enable

This bit enables a device to draw AUX power independent of PME AUX power. Devices that do not implement this capability hardwire this bit to 0. This bit is read/write (sticky) for both PCI host and local processor.

[11] PeDRF_PEENSNO - Enable Snoop Not Required

This bit permits the device or function to set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency.

[14:12] PeDRF_PEMXRRS - Max_Read_Request_Size

This field sets the maximum Read Request size for the device as a Requester.

Table 25.91: PeDRF_PEMXRRS Encoding

Value	Function
3'b000	Maximum Read Request Size = 128 bytes
3'b001	Maximum Read Request Size = 256 bytes
3'b010	Maximum Read Request Size = 512 bytes
3'b011	Maximum Read Request Size = 1024 bytes
3'b100	Maximum Read Request Size = 2048 bytes
3'b101	Maximum Read Request Size = 4096 bytes
3'b110	Reserved
3'b111	Reserved

Note that devices that do not generate Read Request larger than 128 bytes are permitted to implement this field as read-only with a value of 0.

[15] RSVD - Reserved

[16] PeDRF_PECEDET - Correctable Error Detected

This bit indicates status of correctable errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in this register. For devices supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the correctable error mask register.

[17] PeDRF_PENFEDE - Non-fatal Error Detected

This bit indicates status of non-fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in this register. For devices supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the correctable error mask register.

[18] PeDRF_PEFEDET - Fatal Error Detected

This bit indicates status of fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in this register. For devices supporting Advanced Error Handling, errors are

logged in this register regardless of the settings of the correctable error mask register.

[19] PeDRF_PEURDET - Unsupported Request Detected

This bit indicates that the device received an Unsupported Request (UR). Errors are logged in this register regardless of whether error reporting is enabled or not in this register.

[20] PeDRF_PEAPDET - AUX Power Detected

This bit is set if the devices that require AUX power report detected AUX power.

[21] PeDRF_PETPEND - Status Transactions Pending

This bit is set to indicate that the device (Endpoint) has issued Non-Posted Requests (NP) which have not been completed. For Root and Switch Ports, this bit is set to indicate that a Port has issued NP on its behalf using its own Requester ID which have not been completed. This bit is reported as cleared when all outstanding NPs have been completed or have been terminated by the Completion Timeout mechanism.

[31:22] RSVD - Reserved

25.3.1.6.4 PeDRF_PELKCAP: Link Capabilities Register

PCIe Offset: **0x106C**

Table 25.92: PeDRF_PELKCAP

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[03:00]	RO	RO	4'b0010	PeDRF_PEMXLKS	Supported Link Speeds
[09:04]	RO	RO	6'h08	PeDRF_PEMXLWD	Maximum Link Width
[11:10]	RW	RO	2'b11	PeDRF_PEASPMS	ASPM Support
[14:12]	RO	RO	3'b010	PeDRF_PEL0SEL	L0s Exit Latency
[17:15]	RW	RO	3'b001	PeDRF_PEL1ELA	L1 Exit Latency
[18]	RW	RO	1'b0	PeDRF_PECPMGT	Clock Power Management
[19]	RW	RO	1'b0	PeDRF_PESDERC	Surprise Down Error Reporting Capable
[20]	RW	RO	1'b0	PeDRF_PEDLARC	Data Link Layer Active Reporting Capable
[21]	RW	RO	1'b1	PeDRF_PELBWNC	Link Bandwidth Notification Capable
[23:22]	RO	RO	2'b00	RSVD	Reserved
[31:24]	RW	RO	8'h00	PeDRF_PEPORTN	Port Number

[03:00] PeDRF_PEMXLKS - Supported Link Speed

This field indicates the maximum Link speed of the given PCIe Link. The *PCI Express Base Specification,* Revision 2.1 defines values of 4h1 and 4h2 encoding which is equivalent to 2.5GT/s and 5.0 GT/s Link speed respectively. Other encodings are reserved.

[09:04] PeDRF_PEMXLWD - Maximum Link Width

This field indicates the maximum Link width implemented by the component.

Table 25.93: PeDRF_PEMXLWD Encoding

Value	Function
6'b000000	Reserved
6'b000001	x1
6'b000010	x2
6'b000100	x4
6'b001000	x8
6'b001100	x12
6'b010000	x16
6'b100000	x32

[11:10] PeDRF_PEASPMS - Active State Power Management Support

This field indicates the level of ASPM supported on the given PCIe Link.

Table 25.94: PeDRF_PEASPMS Encoding

Value	Function
2'00	Reserved
2'b01	L0s Entry Supported
2'b10	Reserved
2'b11	L0s and L1 Supported

[14:12] PeDRF_PEL0SEL - L0s Exit State

This field indicates the L0s exit latency for the given PCIe Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. The length of time indicated is determined by NFTS value of the other device. But when extended sync bit is set, this field's value would be set to 3'b111.

Table 25.95: PeDRF_PEL0SEL Encoding

Value	Function
3'b000	< 64 ns
3'b001	64 ns < 128 ns
3'b010	128 ns < 256 ns
3'b011	256 ns < 512 ns
3'b100	512 ns < 1 us
3'b101	1 us < 2 us
3'b110	2 us - 4 us
3'b111	> 4 us

[17:15] PeDRF_PEL1ELA - L1 Exit State

This field indicates the L1 exit latency for the given PCIe Link. The value reported indicates the length of time this Port requires to complete transition from L1 to L0. When extended sync bit is set, this field's value would always be set to 3'b111 regardless of what FW tries to write in this field.

Table 25.96: PeDRF_PEL1ELA Encoding

Value	Function
3'b000	< 1 us
3'b001	1 us < 2 us
3'b010	2 us < 4 us
3'b011	4 us < 8 us
3'b100	8 us < 16 us
3'b101	16 us < 32 us
3'b110	32 us - 64 us
3'b111	> 64 us

[18] **PeDRF_PECPMGT - Clock Power Management**

This bit indicates the the component tolerates the removal of any reference clock/s via the “clock request” (CLKREQ#) mechanism when the link is in the L1 and L2/3 Ready link states. If this bit is 0, the component does not have this capability and the reference clock/s must not be removed in this link states. Note that PM configuration software must only permit reference clock removal if all functions of a multifunction device indicate a 1 in this bit.

[19] **PeDRF_PESDERC - Surprise Down Error Reporting Capable**

This bit is set to indicate that the component supports the optional capability of detecting and reporting a Surprise Down error condition. Note that this bit is only applicable to Downstream Ports and must be hardwired to 0 if Upstream Port or do not support the optional capability.

[20] **PeDRF_PEDLARC - Data Link Layer Link Active Repoting Capable**

This bit is set to indicate that the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine (DLCMSM). For a hot-plug capable Downstream Port, this bit must be set to 1. Note that this bit is only applicable for Downstream Ports and must be hardwired to 0 if Upstream Port or do not support the optional capability.

[21] **PeDRF_PELBWNC - Link Bandwidth Notification Capable**

This bit is set to indicate that the component supports the capability of Link Bandwidth Notification Status and interrupt mechanism. A value of 1b allows the setting of Link Autonomous Bandwidth Status and Link Bandwidth Management Status. Also a value of 1b, together with Link Bandwidth Management Interrupt Enable and Link Autonomous Bandwidth Interrupt Enable, allows generation of an interrupt to indicate Link Bandwidth Change.

[23:21] **RSVD - Reserved**

[31:24] **PeDRF_PEPORTN - Port Number**

This field indicates the PCIe Port number for the given PCIe Link. This field is hardware initialized.

25.3.1.6.5 PeDRF_PELKSCO: Link Status and Control Register

PCIe Offset: **0x1070**

Table 25.97: PeDRF_PELKSCO

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[01:00]	RW	RW	2'b00	PeDRF_PEASPMC	ASPM Control
[02]	RO	RO	1'b0	RSVD	Reserved
[03]	RW	RW	1'b1	PeDRF_PERCBOU	Read Completion Boundary
[04]	RW	RW	1'b0	PeDRF_PELKDIS	Link Disable
[05]	RW	RO	1'b0	PeDRF_PERETLK	Retrain Link
[06]	RW	RW	1'b0	PeDRF_PECCCFG	Common Clock Configuration
[07]	RW	RW	1'b0	PeDRF_PEESYNC	Extended Sync
[08]	RW	RW	1'b0	PeDRF_PEECPMA	Enable Clock Power Management
[09]	RW	RW	1'b0	PeDRF_HAWD	Hardware Autonomous Width Disable
[10]	RW	RW	1'b0	PeDRF_LBMIE	Link Bandwidth Mgmt Interrupt Enable
[11]	RW	RW	1'b0	PeDRF_LABIE	Link Autonomous Bandwidth Interrupt Enable
[15:12]	RO	RO	4'b0	RSVD	Reserved
[19:16]	RO	RO	4'h1	PeDRF_PELKSPE	Link Speed
[25:20]	RO	RO	6'hxx	PeDRF_PENLKWI	Negotiated Link Width
[26]	RO	RO	1'b0	UDEF	Undefined
[27]	RO	RO	1'b0	PeDRF_PELKTRA	Link Training
[28]	RW	RO	1'b0	PeDRF_PESCCFG	Slot Clock Configuration
[29]	RO	RO	1'b0	PeDRF_PEDLKAC	Data Link Later Link Active
[30]	RW1C	RO	1'b0	PeDRF_LBMS	Link Bandwidth Mgmt Status
[31]	RW1C	RO	1'b0	peDRF_LABS	Link Autonomous Bandwidth Status

[01:00] PeDRF_PEASPMC - Control Active State Power Management Control

This field controls the level of ASPM supported on the given PCIe Link.

Table 25.98: PeDRF_PEASPMC Encoding

Value	Function
2'b00	Disabled
2'b01	L0s Entry Enabled
2'b10	L1 Entry Enabled
2'b11	L0s and L1 Entry Enabled

[02] RSVD - Reserved

[03] PeDRF_PERCBOU - Control Read Completion Boundary

This field indicates the RCB value for the Root Port if Root Port, may be set configuration software to indicate the RCB value of the Root Port from

the Endpoint if Endpoint, or hardwired to 0 if Switch Port. This bit is read-only if Root Port or Switch Port, and read/write if Endpoint.

Table 25.99: PeDRF_PERCBOU Encoding

Value	Function
1'b0	64 bytes
1'b1	128 bytes

[04] PeDRF_PELKDIS - Control Link Disable

This bit disables the Link and is not applicable and reserved for Endpoints, PCIe to PCI/PCI-X Bridges, and Upstream Ports of Switches. This bit is read/write for both PCI host and local procesor.

[05] PeDRF_PERETLK - Control Retrain Link

This bit enables the initiation on Link retraining by directing the Physical Layer LTSSM to the Recovery state. This bit is not applicable and reserved for Endpoints, PCIe to PCI/PCI-X Bridges, and Upstream Ports of Switches. This bit is read/write for both PCI host and local processor but returns 0 when read.

[06] PeDRF_PECCCFG - Control Common Clock Configuration

This bit indicates that this component and the component at the opposite end of this Link are operating with a distributed common reference clock. Note that after changing this bit's value in both components on a Link, softwre must trigger the Link to retrain.

[07] PeDRF_PEESYNC - Control Extened Synch

This bit forces the transmission of additional ordered sets when exiting the L0s state. This mode provides external devices, such as logic analyzers, monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 state and resumes communication.

[08] PeDRF_PEECPMA - Control Enable Clock Power Management

This bit permits the device to use CLKREQ# signal to power manage link clock according to protocol defined in appropriate form factor specification. When 0, clock power management is disabled and the device must hold CLKREQ# signal low. Note that this feature is only applicable for form factors that support a "clock request" (CLKREQ#) mechanism.

[09] PeDRF_HAWD - Hardware Autonomous Width Disable

This bit, when set to 1, disables the device from changing the linkwidth for reasons other than attempting to correct unreliable link by reducing Link width.

[10] PeDRF_LBMIE - Link Bandwidth Management Interrupt Enable

This bit, when set to 1, enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set.

[11] PeDRF_LABIE - Link Autonomous Bandwidth Interrupt Enable

This bit, when set to 1, enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set.

[15:12] RSVD - Reserved

[19:16] PeDRF_PELKSPE - Status Link Speed

This field indicates the current link speed of the given PCIe Link. The *PCI Express Base Specification,* Revision 2.1 defines the values 4h1 and 4h2 encoding which is equivalent to 2.5GT/s and 5.0 GT/s Link speed respectively. Other encodings are reserved. Note that the value of this field is undefined when the Link is down.

[25:20] PeDRF_PENLKWI - Status Negotiated Link Width

This field indicates the negotiated width of the given PCIe Link. This field is read-only for both PCI host and local processor.

Table 25.100: PeDRF_PENLKWI Encoding

Value	Function
6'b000000	Reserved
6'b000001	x1
6'b000010	x2
6'b000100	x4
6'b001000	x8
6'b001100	x12
6'b010000	x16
6'b100000	x32

Note that the value of this field is undefined when the Link is down.

[26] UDEF - Undefined

The value read from this bit is undefined. In the previous version of the specification, this bit is used to indicate a Link Training Error.

[27] PeDRF_PELKTRA - Status Link Training

This bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or that 1 was written to the Control Retrain Link bit (CRL) of this register but Link training has not yet begun. This bit is not applicable

and reserved for Endpoint devices and Upstream Ports of Switches, and must be hardwire to 0.

[28] PeDRF_PESCCFG - Status Slot Clock Configuration

This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. This bit must be 0 if the device uses an independent clock irrespective of the presence of a reference on the connector. This bit is hardware-initialized.

[29] PeDRF_PEDLKAC - Status Data Link Layer Link Active

This bit indicates the status of the Data Link Control and Management State Machine. When 1, indicates the DL_Active state. Note that this bit must be implemented if the corresponding Data Link Layer Active Capability bit is implemented.

[30] PeDRF_LBMS - Link Bandwidth Management Status

This bit, when set to 1, indicates either the Link retraining has completed after writing 1 to Retrain Link Bit or the device has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. This also means that the change was not an autonomous change.

[31] PeDRF_LABS - Link Autonomous Bandwidth Status

This bit, when set to 1, indicates that the device has autonomously changed Link speed or width for reasons other than to attempt to correct unreliable Link operation.

25.3.1.6.6 PeDRF_PESCAPA: Slot Capabilities Register

PCIe Offset: **0x9074**

Table 25.101: PeDRF_PESCAPA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_PEABPRE	Attention Button Present
[01]	RW	RO	1'b0	PeDRF_PEPCPRE	Power Controller Present
[02]	RW	RO	1'b0	PeDRF_PEMRLSP	MRL Sensor Present
[03]	RW	RO	1'b0	PeDRF_PEAIPRE	Attention Indicator Present
[04]	RW	RO	1'b0	PeDRF_PEPIPRE	Power Indicator Present
[05]	RW	RO	1'b0	PeDRF_PEHPSUR	Hot-Plug Surprise
[06]	RW	RO	1'b0	PeDRF_PEHPCAP	Hot-Plug Capable
[14:07]	RW	RO	8'h00	PeDRF_PESPLVA	Slot Power Limit Value
[16:15]	RW	RO	2'b00	PeDRF_PESPLSC	Slot Power Limit Scale
[17]	RW	RO	1'b0	PeDRF_PEEMIPR	Electromechanical Interlock Present
[18]	RW	RO	1'b0	PeDRF_PENCCSU	No Command Completed Support
[31:19]	RW	RO	13'h00	PeDRF_PEPSNUM	Physical Slot Number

[00] PeDRF_PEABPRE - Attention Button Present

This bit indicates that an Attention Button for this slot is electrically controlled by the chassis. This bit is hardware initialized.

[01] PeDRF_PEPCPRE - Power Controller Present

This bit indicates that a software programmable Power Controller is implemented for this slot/adapter, depending on form factor. This bit is hardware initialized.

[02] PeDRF_PEMRLSP - MRL Sensor Present

This bit indicates that an MRL Sensor is implemented on the chassis for this slot. This bit is hardware initialized.

[03] PeDRF_PEAIPRE - Attention Indicatot Present

This bit indicates that an Attention Indicator is electrically controlled by the chassis. This bit is hardware initialized.

[04] PeDRF_PEPIPRE - Power Indicator Present

This bit indicates that an Power Indicator is electrically controlled by the chassis for this slot. This bit is hardware initialized.

[05] PeDRF_PEHPSUR - Hot-Plug Surprise

This bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This bit is an indication to the operating system to allow such removal without impacting continued software operation. This bit is hardware initialized.

[06] PeDRF_PEHPCAP - Hot-Plug Capable

This bit indicates that this slot is capable of supporting hot-plug operations. This bit is hardware initialized.

[14:07] PeDRF_PESPLVA - Slot Power Limit Value

This field, together with Slot Power Limit Scale Value, specifies the upper limit on power supplied by slot. The power limit, in watts unit, is calculated by multiplying this value with SPLSV value. Note that this register must be implemented if the Slot Implemented bit is set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. This bit is hardware initialized.

[16:15] PeDRF_PESPLSC - Slot Power Limit Scale

This field indicates the scale used for the Slot Power Limit Value field. Note that this register must be implemented if the Slot Implemented bit is set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. This bit is hardware initialized.

Table 25.102: PeDRF_PESPLSC Encoding

Value	Function
2'b00	1.0x
2'b01	0.1x
2'b10	0.01x
2'b11	0.001x

[17] PeDRF_PEEMIPR - Electromechanical Interlock Present

This bit indicates that an Electromechanical Interlock is implemented on the chassis for this slot. This bit is hardware initialized.

[18] PeDRF_PENCCSU - No Command Completed Support

This bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be set if the hot-plug capable port is able to accept writes to all fields of the Slot Control register without delay between successive writes. This bit is hardware initialized.

[31:19] **PeDRF_PEPSNUM - Physical Slot Number**

This field indicates 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 chassism regardless of the form factor associated with the slot. This field must be initialized to 0 for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.

25.3.1.6.7 PeDRF_PESSCTL: Slot Status and Control Register

PCIe Offset: **0x9078**

Table 25.103: PeDRF_PESSCTL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_PEATPEN	Control Attention Button Pressed Enable
[01]	RW	RW	1'b0	PeDRF_PECPFDEN	Control Power Fault Detected Enable
[02]	RW	RW	1'b0	PeDRF_PEMSCEN	Control MRL Sensor Changed Enable
[03]	RW	RW	1'b0	PeDRF_PEPDCEN	Control Presence Detect Changed Enable
[04]	RW	RW	1'b0	PeDRF_PECCIEN	Control Command Completed Interrupt Enable
[05]	RW	RW	1'b0	PeDRF_PEHPIEN	Control Hot-Plug Interrupt Enable
[07:06]	RW	RW	2'b11	PeDRF_PEAICTL	Control Attention Indicator Control
[09:08]	RW	RW	2'b11	PeDRF_PEPICTL	Control Power Indicator Control
[10]	RW	RW	1'b0	PeDRF_PEPCCTL	Control Power Controller Control
[11]	RW	RW	1'b0	PeDRF_PEEICTL	Control Electromechanical Interlock Control
[12]	RW	RW	1'b1	PeDRF_PEDSCEN	Control DLL State Changed Enable
[15:13]	RO	RO	3'b0	RSVD	Reserved
[16]	RO	RWC1	1'b0	PeDRF_PEABPRE	Status Attention Button Pressed
[17]	RO	RWC1	1'b0	PeDRF_PEPFDET	Status Power Fault Detected
[18]	RO	RWC1	1'b0	PeDRF_PEMRLSC	Status MRL Sensor Changed
[19]	RO	RWC1	1'b0	PeDRF_PEPDCHA	Status Presence Detect Changed
[20]	RO	RWC1	1'b0	PeDRF_PECCOMP	Status Command Completed
[21]	RO	RO	1'b0	PeDRF_PEMRLSS	Status MRL Sensor State
[22]	RO	RO	1'b0	PeDRF_PEPDSTA	Status Presence Detect State
[23]	RO	RO	1'b0	PeDRF_PEEMIST	Status Electromechanical Interlock Status
[24]	RO	RWC1	1'b0	PeDRF_PEDLLSC	Status Data Link Later State Changed
[31:25]	RO	RO	7'b0	RSVD	Reserved

[00] PeDRF_PEATPEN - Attention Button Pressed Enable

This bit enables software notification on an attention button pressed event.

[01] PeDRF_PECPFDEN - Power Fault Detected Enable

This bit enables software notification on a power fault event. Note that if Power Fault detection is not supported, this bit is permitted to be read-only with a value of 0.

[02] PeDRF_PEMSCEN - MRL Sensor Changed Enable

This bit enables software notification on a MRL Sensor changed event.

[03] PeDRF_PEPDCEN - Presence Detect Changed Enable

This bit enables software notification on a presence detect changed event.

[04] PeDRF_PECCIEN - Command Completed Interrupt Enable

This bit enables software notification when a hot-plug command is completed by the Hot-Plug Controller if Command Completed notification is supported. If not supported, this bit must be hardwired to 0.

[05] PeDRF_PEHPIEN - Hot-Plug Interrupt Enable

This bit enables generation of an interrupt on enabled hot-plug events. If the Hot Plug Capable field in the Slot Capabilities register (PeDRF_PESCAPA) is 0, this bit is permitted to be read-only with a value of 0.

[07:06] PeDRF_PEAICTL - Attention Indicator Control

Writes to this field set the Attention Indicator to the written state if an Attention Indicator is present. Reads of this field must reflect the value from the latest write.

Table 25.104: PeDRF_PEAICTL Encoding

Value	Function
2'b00	Reserved
2'b01	On
2'b10	Blink
2'b11	Off

[09:08] PeDRF_PEPICTL - Power Indicator Control

Writes to this field set the Power Indicator to the written state if a Power Indicator is present. Reads of this field must reflect the value from the latest write.

Table 25.105: PeDRF_PEPICTL Encoding

Value	Function
2'b00	Reserved
2'b01	On
2'b10	Blink
2'b11	Off

[10] PeDRF_PEPCCTL - Power Controller Control

This field, when written, sets the power state of the slot per the defined encodings. Reads of this field must reflect the value from the latest write.

Depending on the form factor, the power is turned on/off either to the slot or within the adapter.

Table 25.106: PeDRF_PEPCCTL Encoding

Value	Function
1'b0	Power On
1'b1	Power Off

Note that if the Power Controller Present field in the Slot Capabilities register (PeDRF_PESCAPA) is 0, then writes to this field have no effect and the read value is undefined.

[11] PeDRF_PEEICTL - Electromechanical Interlock Control

This field, when set, causes the state of the interlock to toggle if an Electromechanical Interlock is implemented. Reads to this bit always returns 0.

[12] PeDRF_PEDSCEN - DLL State Changed Enable

This bit, when set, enables software notification when Data Link Later Link Active field is changed if the Data Link Later Link Active capability is implemented. This bit read/write for both PCI host and local processor.

[15:13] RSVD - Reserved

[16] PeDRF_PEABPRE - Status Attention Button Pressed

This bit is set when the attention button is pressed if an Attention Button is implemented.

[17] PeDRF_PEPFDET - Status Power Fault Detected

This bit is set when the Power Controller detects a power fault at this slot if a Power Controller that supports power fault detection is implemented.

[18] PeDRF_PEMRLSC - Status MRL Sensor Changed

This bit is set when a MRL Sensor state change is detected if an MRL sensor is implemented.

[19] PeDRF_PEPDCHA - Status Presence Detect Changed

This bit is set when the value reported in Presence Detect State is changed.

[20] PeDRF_PECCOMP - Status Command Completed

This bit is set when a hot-plug command has completed and the Hot-Plug Controller is ready to accept a subsequent command if Command Completed notification is supported. This bit is set as an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command. If Command Completed notification is not supported, this bit must be 0.

[21] PeDRF_PEMRLSS - Status MRL Sensor State

This bit reports the status of the MRL sensor if implemented. When 0, MRL is closed. When 1, MRL is open.

[22] PeDRF_PEPDSTA - Status Presence Detect State

This bit indicates the presence of an adapter in the slot, reflected by the logical OR of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. When 0, the slot is empty. When 1, a card is present in the slot. Note that this bit must be implemented on all Downstream Ports that implement slots.

[23] PeDRF_PEEMIST - Status Electromechanical Interlock Status

This bit indicates the current status of the Electromechanical Interlock if implemented. When 0, EI is disengaged. When 1, EI is engaged.

[24] PeDRF_PEDLLSC - Status Data Link Later State Changed

This bit is set when the value reported in the Status Data Link Layer Link Active field of the Link Status and Control register is changed. In response to a Data Link Layer State Changed event, software must read the Status Data Link Layer Active field to determine if the link is active before initiating configuration cycles to the hot plugged device.

[31:25] RSVD - Reserved

25.3.1.6.8 PeDRF_PERCCTL Root Capabilities and Control

PCIe Offset: **0x107C**

Table 25.107: PeDRF_PERCCTL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_SECEE	System Error on Corr Error Enable
[01]	RW	RW	1'b0	PeDRF_SENFEE	System Error on Non-fatal Error Enable
[02]	RW	RW	1'b0	PeDRF_SEFEE	System Error on Fatal Error Enable
[03]	RW	RW	1'b0	PeDRF_PMEIE	PME InterruptEnable
[04]	RW	RW	1'b0	PeDRF_CRSSVE	CRS Software Visibility Enable
[15:05]	RO	RO	11'h0	RSVD	Reserved
[16]	RW	RO	1'b1	PeDRF_CRSSVC	CRS Software Visibility Capable
[31:17]	RO	RO	15'h0	RSVD	Reserved

[00] PeDRF_SECEE - System Error on Correctable Error Enable

If set, this bit indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the heirarchy associated with this Root Port, or by the Root Port itself.

[01] PeDRF_SENFEE - System Error on Non-Fatal Error Enable

If set, this bit indicates that a System Error should be generated if a Non-fatal error (ERR_NONFATAL) is reported by any of the devices in the heirarchy associated with this Root Port, or by the Root Port itself.

[02] PeDRF_SEFEE - System Error on Fatal Error Enable

If set, this bit indicates that a System Error should be generated if a Fatal error (ERR_FATAL) is reported by any of the devices in the heirarchy associated with this Root Port, or by the Root Port itself.

[03] PeDRF_PMEIE - PME Interrupt Enable

If set, this bit enables interrupt generation upon receipt of a PME Message as reflected in the PME Status (PeDRF_PmeStat) register bit. A PME interrupt is also generated if the PME Status Status register bit is set when this bit is set from a cleared state.

[04] PeDRF_CRSSVE - CRS Software Visibility Enable

If set, this bit enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software.

[15:05] **Reserved**

[16] **PeDRF_CRSSVC - CRS Software Visibility Capable**

If set, this bit indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software.

[31:17] **Reserved**

25.3.1.6.9 PeDRF_PERSTAT Root Status

PCIe Offset: **0x1080**

Table 25.108: PeDRF_PERSTAT

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RO	RO	16'b0	PeDRF_PMEREQID	PME Requestor ID
[16]	RWC1	RO	1'b0	PeDRF_PMESTAT	PME Status
[17]	RO	RO	1'b0	PeDRF_PMEPEND	PME Pending
[31:18]	RO	RO	14'h0	RSVD	Reserved

[15:00] **PeDRF_PMEREQID - PME Requestor ID**

This field indicates the PCI requestor ID of the last PME requestor.

[16] **PeDRF_PMESTAT - PME Status**

This field indicates that PME was asserted by the requestor ID indicated in the PME Requestor ID field (PeDRF_PERSTAT_PmeReqId). Subsequent PME's are kept pending until the status register is cleared by software.

[17] **PeDRF_PMEPEND - PME Pending**

This bit indicates that another PME is pending when the PME Status bit is set. When the PME Status is cleared by software, the PME is delivered by hardware by setting the PME status bit again and updating the Requestor ID appropriately. This bit is cleared by hardware if no more PME's are pending.

[31:18] **Reserved**

25.3.1.6.10PeDRF_PEDCAP2: Device Capabilities 2 Register

PCIe Offset: **0x1084**

Table 25.109: PeDRF_PEDCAP2

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RO		PeDRF_PECTRS	Completion Timeout Ranges Supported
[4]	RW	RO		PeDRF_PECTDS	Completion Timeout Disable Supported
[5]	RW	RO		PeDRF_PEARIFS	ARI Forwarding Supported
[6]	RW	RO		PeDRF_ATOMRS	AtomicOp Routing Supported
[7]	RW	RO		PeDRF_ATOMCS32	32-bit AtomicOp Completer Supported
[8]	RW	RO		PeDRF_ATOMCS64	64-bit AtomicOp Completer Supported
[9]	RW	RO		PeDRF_CASCS128	128-bit CAS Completer Supported
[10]	RW	RO		PeDRF_NOROPRP	No RO-enabled PR-PR Passing
[11]	RW	RO		PeDRF_LTRMS	LTR Mechanism Supported
[13:12]	RW	RO		PeDRF_TPHCS	TPH Completer Supported
[19:14]	RO	RO		RSVD	Reserved
[20]	RW	RO		PeDRF_EXFMTFS	Extended Fmt Field Supported
[21]	RW	RO		PeDRF_EETLPPS	End-End TLP Prefix Supported
[23:22]	RW	RO		PeDRF_MAXEETLPP	Max End-End TLP Prefixes
[31:24]	RO	RO		RSVD	Reserved

[3:0] PeDRF_PECTRS - Completion Timeout Ranges Supported

This field indicates device Function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value.

Table 25.110: PeDRF_PECTRS

Value	Function
3'b000	Completion Timeout programming not supported. Timeout value is in default range 50us to 50ms
3'b001	50us to 10ms
3'b010	10ms to 250ms
3'b011	50us to 250ms
3'b100	10ms to 4s
3'b101	50us to 4s
3'b110	10ms to 64s
3'b111	50us to 64s

[4] PeDRF_PECTDS - Completion Timeout Disable Supported

A value of 1 indicates support for the Completion Timeout Disable mechanism.Note that Phantom Function support for the device must be

enabled by the corresponding control field in the Device Status and Control register.

[5] PeDRF_PEARIFS - ARI Forwarding Supported

For Root Ports, this bit must be set to 1 if this optional capability is supported.

[6] PeDRF_ATOMRS - AtomicOp Routing Supported

For Root Ports, this bit must be set to 1 if this optional capability is supported.

[7] PeDRF_ATOMCS32 - 32-bit AtomicOp Completer Supported

For Root Ports, this bit must be set to 1 if this optional capability is supported; includes FetchAdd, Swap, and CAS AtomicOps

[8] PeDRF_ATOMCS364 - 64-bit AtomicOp Completer Supported

For Root Ports, this bit must be set to 1 if this optional capability is supported; includes FetchAdd, Swap, and CAS AtomicOps

[9] PeDRF_CASCS128 - 128-bit CAS Completer Supported

For Root Ports, this bit must be set to 1 if this optional capability is supported.

[10] PeDRF_NOROPRP - No RO-enabled PR-PR Passing

For RCs that support peer-to-peer traffic between Root Ports, this bit is set to 1 when routing element never carries out the passing permitted associated with the Relaxed Oredring Attribute field. Applies to Posted Requests being forwarded through the RC and does not apply to traffic originating or terminating within the Switch or RC itself.

[11] PeDRF_LTRMS - LTR Mechanism Supported

For Root Ports, this bit must be set to 1 if this optional Latency Tolerance Reporting (LTR) mechanism.

[13:12] PeDRF_TPHCS - TPH Completer Supported

For Root Ports and Endpoints:

Table 25.111: PeDRF_PECTRS

Value	Function
2'b00	TPH and Extended TPH Completer not supported

Table 25.111: PeDRF_PECTRS

Value	Function
2'b01	TPH Completer supported; Extended TPH completer not supported
2'b10	Reserved
2'b11	Both TPH and Extended TPH Completer supported

[19:14] RSVD - Reserved

[20] PeDRF_EXFMTFS - Extended Fmt Field Supported

If Set, the Function supports the 3-bit definition of the Fmt field. If Clear, the Function supports a 2-bit definition of the Fmt field.

[21] PeDRF_EETLPPS - End-End TLP Prefix Supported

This field indicates whether End-End TLP Prefix support is offered by a Function. If set to 1, support is provided to receive TLPs containing End-End TLP Prefixes.

[23:22] PeDRF_MAXEETLPP - Max End-End TLP Prefixes

This field indicates the maximum number of End-End TLP Prefixes supported:

Table 25.112: PeDRF_PECTRS

Value	Function
2'b00	4 End-End TLP Prefixes
2'b01	1 End-End TLP Prefix
2'b10	2 End-End TLP Prefixes
2'b11	3 End-End TLP Prefixes

[31:24] RSVD - Reserved

25.3.1.6.11PeDRF_PEDSCTL2: Device Status 2 and Control 2 Register

PCIe Offset: **0x1088**

Table 25.113: PeDRF_PEDSCTL2

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RW	4'd0	PeDRF_CTIMEOUT	Completion Timeout Value
[4]	RW	RW	1'b0	PeDRF_CTIMEDIS	Completion Timeout Disable
[5]	RW	RW	1'b0	PeDRF_ARIFE	ARI Forwarding Enable
[6]	RW	RW	1'b0	PeDRF_ATOMRE	AtomicOp Requester Enable
[7]	RW	RW	1'b0	PeDRF_ATOMEB	AtomicOp Egress Blocking
[8]	RW	RW	1'b0	PeDRF_IDOREQEN	IDO Request Enable
[9]	RW	RW	1'b0	PeDRF_IDOCOMEN	IDO Completion Enable
[10]	RW	RW	1'b0	PeDRF_LTRMEN	LTR Mechanism Enable
[14:11]	RO	RO	4'd0	RSVD	Reserved
[15]	RW	RW	1'b0	PeDRF_EETLPPB	End-End TLP Prefix Blocking
[31:16]	RO	RO		RSVD	Reserved

[3:0] PeDRF_CTIMEOUT - Completion Timeout Value

For Root Ports and Endpoints that issue Requests on their own behalf, supporting Completion Timeout programmability, this field allows system software to modify the Completion Timeout value.

Table 25.114: PeDRF_CTIMEOUT

Value	Function
4'b0000	50us to 50ms (Default Range)
4'b0001	50us to 100us
4'b0010	1ms to 10ms
4'b0011	Reserved
4'b0100	Reserved
4'b0101	16ms to 55ms
4'b0110	65ms to 210ms
4'b0111	Reserved
4'b1000	Reserved
4'b1001	260ms to 900ms
4'b1010	1s to 3.5s
4'b1011	Reserved
4'b1100	Reserved
4'b1101	4s to 13s
4'b1110	17s to 64s
4'b1111	Reserved

[4] PeDRF_CTIMEDIS - Completion Timeout Disable

When set, this bit disables the Completion Timeout mechanism.

[5] **PeDRF_ARIFE - ARI Forwarding Enable**

When set, the Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port.

[6] **PeDRF_ATOMRE - AtomicOp Requester Enable**

For Endpoints and Root Ports, when this bit is set and the Bus Master Enable bit in the Command register is also set, the device is allowed to initiate AtomicOp Requests.

[7] **PeDRF_ATOMEB - AtomicOp Egress Blocking**

For Root Porst that implement AtomicOp routing capability, setting this field to 1 will block AtomicOp Requests that target going out this Egress Port.

[8] **PeDRF_IDOREQEN - IDO Request Enable**

For Endpoints and Root Ports supporting this capability, if this field is 1, the device is permitted to set the ID-Based Ordering (IDO) bit of Requests it initiates.

[9] **PeDRF_IDOCOMEN - IDO Completion Enable**

For Endpoints and Root Ports supporting this capability, if this field is 1, the device is permitted to set the ID-Based Ordering (IDO) bit of Completions it returns.

[10] **PeDRF_LTRMEN - LTR Mechanism Enable**

Setting this field to 1 enables the Latency Tolerance Reporting (LTR) mechanism.

[14:11] **RSVD - Reserved**

[15] **PeDRF_EETLPPB - End-End TLP Prefix Blocking**

When this field is 1 (Forwarding Enabled), the device is permitted to send TLPs with End-End TLP Prefixes

When this field is 0 (Forwarding Blocked), the device is not permitted to send TLPs with End-End TLP Prefixes

25.3.1.6.12PeDRF_PELKSCO2: Link Status 2 and Control 2 Register

PCIe Offset: **0x1090**

Table 25.115: PeDRF_PELKSCO2

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RWS	RWS	4'b0010	PeDRF_TRGLNKSP	Target Link Speed
[4]	RWS	RWS	1'b0	PeDRF_ENTCOMP	Enter Compliance
[5]	RWS	RWS	1'b0	PeDRF_HWASD	Hardware Autonomous Speed Disable
[6]	RW	RW	1'b0	PeDRF_SELDEMP	Selectable De-emphasis
[9:7]	RWS	RWS	3'b000	PeDRF_TXMARG	Transmit Margin
[10]	RWS	RWS	1'b0	PeDRF_EMODCOMP	Enter Modified Compliance
[11]	RWS	RWS	1'b0	PeDRF_COMPSOS	Compliance SOS
[12]	RWS	RWS	1'b0	PeDRF_COMPDEMP	Compliance De-emphasis
[15:13]	RO	RO	3'b000	RSVD	Reserved
[16]	RW	RO	1'b1	PeDRF_CURDEMPLV	Current De-emphasis Level
[31:17]	RO	RO	14'd0	RSVD	Reserved

[3:0] PeDRF_TRGLNKSP - Target Link Speed

For Downstream Ports, this field sets an upper limit on Link operational speed by restricting the values advertised by the Upstream component in its training sequences.

Table 25.116: PeDRF_TRGLNKSP Encoding

Value	Function
4'b0000	Reserved
4'b0001	2.5GT/s Target Link Speed
4'b0010	5.0GT/s Target Link Speed
4'b0011 - 4'b1111	Reserved

[4] PeDRF_ENTCOMP - Enter Compliance

Software is permitted to force a Link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1 in both components on a Link, and then initiating a hot reset on the Link.

[5] PeDRF_HWASD - Hardware Autonomous Speed Disable

If this field is 1, this bit disables hardware from changing the Link speed for device-specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. Initial transition to the highest supported common link speed is not blocked by this bit.

[6] PeDRF_SELDEMP - Selectable De-emphasis

When the Link is operating at 5.0GT/s speed, this bit is used to control the transmit de-emphasis of the link in Recovery and Loopback states. When this field is 1, de-emphasis is -3.5dB. When this field is 0, de-emphasis is -6dB. Firmware must ensure that this field must be set to the desired value first before directing the Link to transition to Recovery or Loopback.

[9:7] PeDRF_TXMARG - Transmit Margin

This field controls the value of the non-deemphasized voltage level at the Transmitter pins. This field is reset to 3'b000 on entry to the LTSSM Polling.Configuration substate.

Table 25.117: PeDRF_TXMARG Encoding

Value	Function
3'b000	Normal operating range
3'b001 to 3'b111	TBD

[10] PeDRF_EMODCOMP - Enter Modified Compliance

When this field is 1, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate.

[11] PeDRF_COMPSOS - Compliance SOS

When this field is 1, the LTSSM is required to send SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern.

[12] PeDRF_COMPDEMP - Compliance De-emphasis

This field sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the PeDRF_ENTCOMP being 1. When this field is 1, Compliance De-emphasis is -3.5dB. When this field is 0, Compliance De-emphasis is -6dB. Firmware must ensure that this field must be set to the desired value first before setting PeDRF_ENTCOMP to 1. Furthermore, Firmware are allowed to modify this field only during debug and compliance purposes.

[16] PeDRF_CURDEMPLV - Current De-emphasis Level

When this Link is operating at 5.0GT/s speed, this field reflects the level of de-emphasis. When this field is 1, de-emphasis is -3.5dB. When this field is 0, de-emphasis is -6dB.

[15:13] [31:17] RSVD - Reserved

25.3.1.6.13PeDRF_PIML0/1/2/3/4/5: PCIe Inbound Map Local 0/1/2/3/4/5 Register

PCIe Offset: **0x10A0**, **0x10A4, 0x10A8, 0x10AC, 0x1200, 0x1204**

Table 25.118: PeDRF_PIML

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RO	RO	8'h0	RSVD	Reserved
[31:08]	RW	RW	24'h0	PeDRF_PIML	PIM Local Base Address

[07:00] RSVD - Reserved

This is hardwired to 0.

[31:08] PeDRF_PIML - PIM Local Base Address

If PeDRF_PIMS[0] = 1, the address of Memory/IO Request received through a BAR is translated to its equivalent PIM Local address. This is writable, but writable bits depend on the value of PeDRF_PIMS0/1/2/3/4/5.

Table 25.119: PIM Local Base Address Writable Bits

PeDRF_PIMS[31:08]	Writable Bits
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xxx1	[31:8]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xx10	[31:9]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_x100	[31:10]
24'bxxxx_xxxx_xxxx_xxxx_xxxx_1000	[31:11]
24'bxxxx_xxxx_xxxx_xxxx_xxx1_000	[31:12]
24'bxxxx_xxxx_xxxx_xxxx_xx10_0000	[31:13]
24'bxxxx_xxxx_xxxx_xxxx_x100_0000	[31:14]
24'bxxxx_xxxx_xxxx_xxxx_1000_0000	[31:15]
24'bxxxx_xxxx_xxxx_xxx1_0000_0000	[31:16]
24'bxxxx_xxxx_xxxx_xx10_0000_0000	[31:17]
24'bxxxx_xxxx_xxxx_x100_0000_0000	[31:18]
24'bxxxx_xxxx_xxxx_1000_0000_0000	[31:19]
24'bxxxx_xxxx_xxx1_0000_0000_0000	[31:20]
24'bxxxx_xxxx_xx10_0000_0000_0000	[31:21]
24'bxxxx_xxxx_x100_0000_0000_0000	[31:22]
24'bxxxx_xxxx_1000_0000_0000_0000	[31:23]
24'bxxxx_xxx1_0000_0000_0000_0000	[31:24]
24'bxxxx_xx10_0000_0000_0000_0000	[31:25]
24'bxxxx_x100_0000_0000_0000_0000	[31:26]
24'bxxxx_1000_0000_0000_0000_0000	[31:27]
24'bxxx1_0000_0000_0000_0000_0000	[31:28]
24'bxx10_0000_0000_0000_0000_0000	[31:29]
24'bx100_0000_0000_0000_0000_0000	[31:30]
24'b1000_0000_0000_0000_0000_0000	[31]
24'b0000_0000_0000_0000_0000_0000	-

25.3.1.6.14PeDRF_PIMS0/1/2/3/4/5: PCIe Inbound Map Size 0/1/2/3/4/5 Register

PCIe Offset: **0x10B0**, **0x10B4, 0x10B8, 0x10BC, 0x1210, 0x1214**

Table 25.120: PeDRF_PIMS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_PIMEN	PIM Enable
[07:01]	RO	RO	7'h0	RSVD	Reserved
[31:08]	RW	RO	24'hfff000	PeDRF_PIMS	PIM Size

[00] PeDRF_PIMEN - PIM Enable

If set, this bit enables BAR registers and decoders.

[07:01] RSVD - Reserved

[31:08] PeDRF_PIMS - PIM Size

This field indicates the size of PCI Memory Space Talino™ EDC is requesting to the Root Complex. While the Root Complex has not yet assigned an address range to a BAR, any write access to this field is also reflected in the corresponding PeDRF_HTBAR[31:8] field.

Table 25.121: PeDRF_PIMS

PeDRF_PIMS[31:08]	Size
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xxx1	256B
24'bxxxx_xxxx_xxxx_xxxx_xxxx_xx10	512B
24'bxxxx_xxxx_xxxx_xxxx_xxxx_x100	1KB
24'bxxxx_xxxx_xxxx_xxxx_xxxx_1000	2KB
24'bxxxx_xxxx_xxxx_xxxx_xxx1_000	4KB
24'bxxxx_xxxx_xxxx_xxxx_xx10_0000	8KB
24'bxxxx_xxxx_xxxx_xxxx_x100_0000	16KB
24'bxxxx_xxxx_xxxx_xxxx_1000_0000	32KB
24'bxxxx_xxxx_xxxx_xxx1_0000_0000	64KB
24'bxxxx_xxxx_xxxx_xx10_0000_0000	128KB
24'bxxxx_xxxx_xxxx_x100_0000_0000	256KB
24'bxxxx_xxxx_xxxx_1000_0000_0000	512KB
24'bxxxx_xxxx_xxx1_0000_0000_0000	1MB
24'bxxxx_xxxx_xx10_0000_0000_0000	2MB
24'bxxxx_xxxx_x100_0000_0000_0000	4MB
24'bxxxx_xxxx_1000_0000_0000_0000	8MB
24'bxxxx_xxx1_0000_0000_0000_0000	16MB
24'bxxxx_xx10_0000_0000_0000_0000	32MB

PeDRF_PIMS[31:08]	Size
24'bxxxx_x100_0000_0000_0000_0000	64MB
24'bxxxx_1000_0000_0000_0000_0000	128MB
24'bxxx1_0000_0000_0000_0000_0000	256MB
24'bxx10_0000_0000_0000_0000_0000	512MB
24'bx100_0000_0000_0000_0000_0000	1GB
24'b1000_0000_0000_0000_0000_0000	2GB
24'b0000_0000_0000_0000_0000_0000	4GB

25.3.1.6.15PeDRF_POMP0: PCIe Outbound Map PCIe 0 Register

PCIe Offset: **0x10C0**

Table 25.122: PeDRF_POMP0

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[11:00]	RO	RO	12'h0	RSVD	Reserved
[31:12]	RW	RW	20'h0	PeDRF_POMP0	POMP0 Base Address

[11:00] RSVD - Reserved

[31:12] PeDRF_POMP0 - POMP0 Base Address

Any local CPU access to a local address configured as PCI Memory or IO Space triggers the generation of a Memory/IO Request TLP. The actual TLP address that is used is the equivalent PCI address of the accessed local address, that is, access to a local address that falls within the POML0 range is translated to an equivalent POMP0 address.

Minimum address range is 4KBytes.

25.3.1.6.16PeDRF_POMP1/2/3/4/5: PCIe Outbound Map PCIe 1/2/3/4/5 Register

PCIe Offset: **0x10C4, 0x10C8, 0x10CC, 0x1220, 0x1224**

Table 25.123: PeDRF_POMP

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[19:00]	RO	RO	20'h0	RSVD	Reserved
[31:20]	RW	RW	12'h0	PeDRF_POMP	POMP Base Address

[19:00] RSVD - Reserved

[31:20] PeDRF_POMP - POMP Base Address

Any local CPU access to a local address configured as PCI Memory or IO Space triggers the generation of a Memory/IO Request TLP. The actual TLP address that is used is the equivalent PCI address of the accessed local address, that is, access to a local address that falls within the POML range is translated to an equivalent POMP address.

Minimum address range is 1MBytes.

25.3.1.6.17PeDRF_POML0/1/2/3/4/5: PCIe Outbound Map Local 0/1/2/3/4/5 Register

PCIe Offset: **0x10D0**, **0x10D4, 0x10D8, 0x10DC, 0x1230, 0x1234**

Table 25.124: PeDRF_POML0

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[19:00]	RO	RO	20'h0	RSVD	Reserved
[31:20]	RW	RW	12'h0	PeDRF_POML	POML Base Address

[19:00] RSVD - Reserved

[31:20] PeDRF_POML - POML Base Address

Any local CPU access to a local address configured as PCI Memory or IO Space triggers the generation of a Memory/IO Request TLP. The actual TLP address that is used is the equivalent PCI address of the accessed local address, that is, access to a local address that falls within the POML range is translated to an equivalent POMP address.

Minimum address range is 1MBytes.

25.3.1.6.18PeDRF_POMS0/1/2/3/4/5: PCIe Outbound Map Size 0/1/2/3/4/5 Register

PCIe Offset: **0x10E0**, **0x10E4, 0x10E8, 0x10EC, 0x1240, 0x1244**

Table 25.125: PeDRF_POMS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_POMEN	POM Enable
[01]	RW	RO	1'b0	PeDRF_POMATEN	POM Address Translation Enable
[19:02]	RO	RO	18'h0	RSVD	Reserved
[31:20]	RW	RW	12'h0	PeDRF_POMS	POMS Size

[00] PeDRF_POMEN - POM Enable

If set, this bit enables the TLP generation if the POM local address range is accessed by the local CPU.

[01] PeDRF_POMATEN - POM Address Translation Enable

If set, this bit enables the mapping of a POM local address to the equivalent POM PCI address.

[19:02] RSVD - Reserved

[31:20] PeDRF_POMS - POM Size

POM Size together with POML base address determines how much address range a local PCI Memory/IO space is mapped to the System PCI Memory/IO space.

25.3.1.6.19PeDRF_EXPIML: Expansion ROM Local Address Register

PCIe Offset: **0x1250**

Table 25.126: PeDRF_EXPIML

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[10:00]	RO	RO	11'h0	RSVD	Reserved
[31:11]	RW	RO	21'h0	PeDRF_EXPIML	Local Address

[10:00] RSVD - Reserved

[31:11] PeDRF_EXPIML - Expansion ROM Local Address

If the device received an access targeting its Expansion ROM space, the address of the request is translated into an equivalent Expansion ROM local address.

25.3.1.6.20PeDRF_EXPIMS: Expansion ROM Size Register

PCIe Offset: **0x1254**

Table 25.127: PeDRF_EXPIMS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[10:00]	RO	RO	11'h0	RSVD	Reserved
[31:11]	RW	RO	21'h1f_ffff	PeDRF_EXPIMS	Size

[10:00] RSVD - Reserved

[31:11] PeDRF_EXPIMS - Expansion ROM Size

This field determines the size of the expansion ROM of the device.

Table 25.128: PeDRF_EXPIMS

PeDRF_EXPIMS[31:11]	Size
21'bx_xxxx_xxxx_xxxx_xxxx_xxx1	2KB
21'bx_xxxx_xxxx_xxxx_xxxx_xx10	4KB
21'bx_xxxx_xxxx_xxxx_xxxx_x100	8KB
21'bx_xxxx_xxxx_xxxx_xxxx_1000	16KB
21'bx_xxxx_xxxx_xxxx_xxx1_000	32KB
21'bx_xxxx_xxxx_xxxx_xx10_0000	64KB
21'bx_xxxx_xxxx_xxxx_x100_0000	128KB
21'bx_xxxx_xxxx_xxxx_1000_0000	256KB
21'bx_xxxx_xxxx_xxx1_0000_0000	512KB
21'bx_xxxx_xxxx_xx10_0000_0000	1MB
21'bx_xxxx_xxxx_x100_0000_0000	2MB
21'bx_xxxx_xxxx_1000_0000_0000	4MB
21'bx_xxxx_xxx1_0000_0000_0000	8MB
21'bx_xxxx_xx10_0000_0000_0000	16MB
21'bx_xxxx_x100_0000_0000_0000	32MB
21'bx_xxxx_1000_0000_0000_0000	64MB
21'bx_xxx1_0000_0000_0000_0000	128MB
21'bx_xx10_0000_0000_0000_0000	256MB
21'bx_x100_0000_0000_0000_0000	512MB
21'bx_1000_0000_0000_0000_0000	1GB
21'h1_0000_0000_0000_0000_0000	2GB
21'b0_0000_0000_0000_0000_0000	4GB

25.3.1.7 Advanced Error Reporting Capability Registers

These registers are instantiated per function implemented by the PCIe device.

25.3.1.7.1 PeDRF_AECPHDR: Adv Error Reporting Capability Header Register

PCIe Offset: **0x1100**

Table 25.129: PeDRF_AECPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RO	RO	16'h0001	PeDRF_AECAPID	Advance Error Reporting Capability ID
[19:16]	RO	RO	4'h1	PeDRF_AECAPVE	Capability Version
[31:20]	RW	RO	12'h138	PeDRF_AENCAPP	Next Capability Offset

[15:00] PeDRF_AECAPID - Advanced Error Reporting Capability ID

PCI-SIG defines a value of 0001h for the Advanced Error Reporting Capability structure. This field is read-only for both PCI host and local processor.

[19:16] PeDRF_AECAPVE - Advanced Error Reporting Capability Version

The value of this field is 2h for the current version of the PCIe Gen2 specification. This field is read-only for both PCI host and local processor.

[31:20] PeDRF_AENCAPP - Next Capability Offset

This field contains the offset to the next PCIe capability structure or 000h if no other items exist in the linked list of capabilities. In this design, the next capability offset is the Device Serial Number Capability Header (0x1138).

25.3.1.7.2 PeDRF_AEUESTA: Uncorrectable Error Status Register

PCIe Offset: **0x1104**

Table 25.130: PeDRF_AEUESTA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RO	RO	1'bx	UDEF	Undefined
[03:01]	RO	RO	3'h0	RSVD	Reserved
[04]	RW1CS	RW1CS	1'b0	PeDRF_AEDLPES	Data Link Protocol Error Status
[05]	RW1CS	RW1CS	1'b0	PeDRF_AESDEST	Surprise Down Error Status
[11:06]	RO	RO	6'h0	RSVD	Reserved
[12]	RW1CS	RW1CS	1'b0	PeDRF_AEPTLPS	Poisoned TLP Status
[13]	RW1CS	RW1CS	1'b0	PeDRF_AEFCPES	Flow Control Protocol Error Status
[14]	RW1CS	RW1CS	1'b0	PeDRF_AECTSTA	Completion Timeout Status
[15]	RW1CS	RW1CS	1'b0	PeDRF_AECASTA	Completer Abort Status
[16]	RW1CS	RW1CS	1'b0	PeDRF_AEUCSTA	Unexpected Completion Status
[17]	RW1CS	RW1CS	1'b0	PeDRF_AEROSTA	Receiver Overflow Status
[18]	RW1CS	RW1CS	1'b0	PeDRF_AEMTSTA	Malformed TLP Status
[19]	RW1CS	RW1CS	1'b0	PeDRF_AEEESTA	ECRC Error Status
[20]	RW1CS	RW1CS	1'b0	PeDRF_AEUREST	Unsupported Request Error Status
[31:21]	RO	RO	11'h0	RSVD	Reserved

[00] UDEF - Undefined

This bit has undefined value. In the *PCI Express Base Specification,* Revision 1.0a, this bit is used to indicate a Link Training Error. System software must ignore the value read from this bit but is permitted to write any value to this bit.

[03:01] RSVD - Reserved

[04] PeDRF_AEDLPES - Data Link Protocol Error Status

This bit indicates the status of a detected Data Link Protocol Error. This refers to ACK/NAK DLLP with bad sequence number.

[05] PeDRF_AESDEST - Surprise Down Error Status

This bit indicates the status of a detected Surprise Down Error. This error is flagged when an invalid transition of the link from DL_Active to DL_Inactive occurs.

[11:06] RSVD - Reserved

[12] **PeDRF_AEPTLPS - Poisoned TLP Status**

This bit indicates the status of a detected Poisoned TLP.

[13] **PeDRF_AEFCPES - Flow Control Protocol Error Status**

This bit indicates the status of a detected Flow Control Protocol Error (FCPE). This feature is optional and may not be implemented.

[14] **PeDRF_AECTSTA - Completion Timeout Status**

This bit indicates the status of a detected Completion Timeout. This error is flagged when the Requester failed to receive an expected Completion after 50 ms.

[15] **PeDRF_AECASTA - Completer Abort Status**

This bit indicates the status of a detected Completer Abort (CA) Error . This feature is optional and may not be implemented.

[16] **PeDRF_AEUCSTA - Unexpected Completion Status**

This bit indicates the status of a detected Unexpected Competion Error. This error is flagged when the device receives a Completion that does not corresponds to any of the outstanding Requests issued by the device.

[17] **PeDRF_AEROSTA - Receiver Overflow Status**

This bit indicates the status of a detected Receiver Overflow Error . This is flagged when the received TLP exceeds the credits allocated.

[18] **PeDRF_AEMTSTA - Malformed TLP Status**

This bit indicates the status of a detected Malformed TLP Error.

[19] **PeDRF_AEEESTA - ECRC Error Status**

This bit indicates the status of a detected ECRC Error .

[20] **PeDRF_AEUREST - Unsupported Request Error Status**

This bit indicates the status of a detected Unsupported Request (UR) Error.

[31:21] **RSVD - Reserved**

25.3.1.7.3 PeDRF_AEUEMAS: Uncorrectable Error Mask Register

PCIe Offset: **0x1108**

Table 25.131: PeDRF_AEUEMAS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RO	RO	1'bx	UDEF	Undefined
[03:01]	RO	RO	3'h0	RSVD	Reserved
[04]	RWS	RWS	1'b0	PeDRF_AEDLPEM	Data Link Protocol Error Mask
[05]	RWS	RWS	1'b0	PeDRF_AESDEMA	Surprise Down Error Mask
[11:06]	RO	RO	6'h0	RSVD	Reserved
[12]	RWS	RWS	1'b0	PeDRF_AEPTLPM	Poisoned TLP Mask
[13]	RWS	RWS	1'b0	PeDRF_AEFCPEM	Flow Control Protocol Error Mask
[14]	RWS	RWS	1'b0	PeDRF_AECTMAS	Completion Timeout Mask
[15]	RWS	RWS	1'b0	PeDRF_AECAMAS	Completer Abort Mask
[16]	RWS	RWS	1'b0	PeDRF_AEUCMAS	Unexpected Completion Mask
[17]	RWS	RWS	1'b0	PeDRF_AEROMAS	Receiver Overflow Mask
[18]	RWS	RWS	1'b0	PeDRF_AEMTMAS	Malformed TLP Mask
[19]	RWS	RWS	1'b0	PeDRF_AEEEMAS	ECRC Error Mask
[20]	RWS	RWS	1'b0	PeDRF_AEUREMA	Unsupported Request Error Mask
[31:21]	RO	RO	11'h0	RSVD	Reserved

[00] UDEF - Undefined

This bit has undefined value. In the *PCI Express Base Specification,* Revision 1.0a, this bit is used to indicate a Link Training Error. System software must ignore the value read from this bit but is permitted to write any value to this bit.

[03:01] Reserved

[04] PeDRF_AEDLPEM - Data Link Protocol Error Mask

This bit, when set, prevents the logging of a detected Data Link Protocol Error in the Header Log register (PeDRF_AEHDLOG), updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[05] PeDRF_AESDEMA - Surprise Down Error Mask

This bit, when set, prevents the logging of a detected Surprise Down Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex. This feature is optional and may not be implemented.

[11:06] Reserved

[12] PeDRF_AEPTLPM - Poisoned TLP Mask

This bit, when set, prevents the logging of a detected Poisoned TLP in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[13] PeDRF_AEFCPEM - Flow Control Protocol Error Mask

This bit, when set, prevents the logging of a detected Flow Control Protocol Error (FCPE) in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[14] PeDRF_AECTMAS - Completion Timeout Mask

This bit, when set, prevents the logging of a detected Completion Timeout Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[15] PeDRF_AECAMAS - Completer Abort Mask

This bit, when set, prevents the logging of a detected Completer Abort (CA) Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex. This feature is optional and may not be implemented.

[16] PeDRF_AEUCMAS - Unexpected Completion Mask

This bit, when set, prevents the logging of a detected Unexpected Competion Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[17] PeDRF_AEROMAS - Receiver Overflow Mask

This bit, when set, prevents the logging of a detected Receiver Overflow Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex. This feature is optional and may not be implemented. If not implemented, this bit is read-only with a value of 0.

[18] PeDRF_AEMTMAS - Malformed TLP Mask

This bit, when set, prevents the logging of a detected Malformed TLP Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[19] **PeDRF_AEEEMAS - ECRC Error Mask**

This bit, when set, prevents the logging of a detected ECRC Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex. This feature is optional and may not be implemented.

[20] **PeDRF_AEUREMA - Unsupported Request Error Mask**

This bit, when set, prevents the logging of a detected Unsupported Request (UR) Error in the Header Log register, updating of the First Error Pointer, and reporting of the error to the PCIe Root Complex.

[31:21] **RSVD - Reserved**

25.3.1.7.4 PeDRF_AEUESVT: Uncorrectable Error Severity Register

PCIe Offset: **0x110C**

Table 25.132: PeDRF_AEUESVT

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RO	RO	1'bx	UDEF	Undefined
[03:01]	RO	RO	3'h0	RSVD	Reserved
[04]	RWS	RWS	1'b1	PeDRF_AEDLPEV	Data Link Protocol Error Severity
[05]	RWS	RWS	1'b1	PeDRF_AESDESV	Surprise Down Error Severity
[11:06]	RO	RO	6'h00	RSVD	Reserved
[12]	RWS	RWS	1'b0	PeDRF_AEPTLPV	Poisoned TLP Severity
[13]	RWS	RWS	1'b1	PeDRF_AEFCPEV	Flow Control Protocol Error Severity
[14]	RWS	RWS	1'b0	PeDRF_AECTESV	Completion Timeout Severity
[15]	RWS	RWS	1'b0	PeDRF_AECAESV	Completer Abort Severity
[16]	RWS	RWS	1'b0	PeDRF_AEUCESV	Unexpected Completion Severity
[17]	RWS	RWS	1'b1	PeDRF_AEROESV	Receiver Overflow Severity
[18]	RWS	RWS	1'b1	PeDRF_AEMTSVT	Malformed TLP Severity
[19]	RWS	RWS	1'b0	PeDRF_AEEESVT	ECRC Error Severity
[20]	RWS	RWS	1'b0	PeDRF_AEURESV	Unsupported Request Error Severity
[31:21]	RO	RO	11'h00	RSVD	Reserved

[00] UDEF - Undefined

This bit has undefined value. In the *PCI Express Base Specification,* Revision 1.0a, this bit is used to indicate a Link Training Error. System software must ignore the value read from this bit but is permitted to write any value to this bit.

[03:01] RSVD - Reserved

[04] PeDRF_AEDLPEV - Data Link Protocol Error Severity

This bit, when set, reports a detected Data Link Protocol Error as a fatal error. Else, the detected error is reported as a non-fatal error.

[05] PeDRF_AESDESV - Surprise Down Error Severity

This bit, when set, reports a detected Surprise Down Error as a fatal error. Else, the detected error is reported as a non-fatal error. This feature is optional and may not be implemented.

[11:06] RSVD - Reserved

[12] **PeDRF_AEPTLPV - Poisoned TLP Severity**

This bit, when set, reports a detected Poisoned TLP as a fatal error. Else, the detected error is reported as a non-fatal error.

[13] **PeDRF_AEFCPEV - Flow Control Protocol Error Severity**

This bit, when set, reports a detected Flow Control Protocol Error as a fatal errror. Else, the detected error is reported as a non-fatal error.

[14] **PeDRF_AECTESV - Completion Timeout Severity**

This bit, when set, reports a detected Completion Timer Error as a fatal error. Else, the detected error is reported as a non-fatal error.

[15] **PeDRF_AECAESV - Completer Abort Severity**

This bit, when set, reports a detected Completer Abort (CA) Error as a fatal error. Else, the detected error is reported as a non-fatal error. This feature is optional and may not be implemented.

[16] **PeDRF_AEUCESV - Unexpected Completion Severity**

This bit, when set, reports a detected Unexpected Completion Error as a fatal error. Else, the detected error is reported as a non-fatal error.

[17] **PeDRF_AEROESV - Receiver Overflow Severity**

This bit, when set, reports a detected Receiver Overflow Error as a fatal error. Else, the detected error is reported as a non-fatal error. This feature is optional and may not be implemented.

[18] **PeDRF_AEMTSVT - Malformed TLP Severity**

This bit, when set, reports a detected Malformed TLP Error as a fatal error. Else, the detected error is reported as a non-fatal error.

[19] **PeDRF_AEEESVT - ECRC Error Severity**

This bit, when set, reports a detected ECRC Error as a fatal error. Else, the detected error is reported as a non-fatal error. This feature is optional and may not be implemented.

[20] **PeDRF_AEURESV - Unsupported Request Error Severity**

This bit, when set, reports a detected Unsupported Request (UR) Error as a fatal error. Else, the detected error is reported as a non-fatal error.

[31:21] **RSVD - Reserved**

25.3.1.7.5 PeDRF_AECESTA: Correctable Error Status Register

PCIe Offset: **0x1110**

Table 25.133: PeDRF_AECESTA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW1CS	RW1CS	1'b0	PeDRF_AERXESTA	Receiver Error Status
[05:01]	RO	RO	5'h0	RSVD	Reserved
[06]	RW1CS	RW1CS	1'b0	PeDRF_AEBTLPS	Bad TLP Status
[07]	RW1CS	RW1CS	1'b0	PeDRF_AEBDLPS	Bad DLLP Status
[08]	RW1CS	RW1CS	1'b0	PeDRF_AERNRST	REPLAY_NUM Rollover Status
[11:09]	RO	RO	3'h0	RSVD	Reserved
[12]	RW1CS	RW1CS	1'b0	PeDRF_AERTTST	Replay Timer Timeout Status
[13]	RW1CS	RW1CS	1'b0	PeDRF_AEANFES	Advisory Non-fatal Error Status
[31:14]	RO	RO	18'h0	RSVD	Reserved

[00] PeDRF_AERXESTA - Receiver Error Status

This bit, when set, indicates that a Receiver Error has been detected. Amongst the many types of receiver error, only the 8b/10b decoding error can cause this bit to be set.

[05:01] RSVD - Reserved

[06] PeDRF_AEBTLPS - Bad TLP Status

This bit, when set, indicates that a Bad TLP has been detected. This is due to the Link Layer reception of a TLP with CRC Check error, End Bad TLP (EDB) character, and incorrect Sequence Number

[07] PeDRF_AEBDLPS - Bad DLLP Status

This bit, when set, indicates that a Bad DLLP has been detected. This is caused by a DLLP with bad CRC

[08] PeDRF_AERNRST - REPLAY_NUM Rollover Status

This bit, when set, indicates that REPLAY_NUM rollover has occured. This event is caused when a transaction was retransmitted three times without successful delivery.

[11:09] RSVD - Reserved

[12] **PeDRF_AERTTST - Replay Timer Timeout Status**

This bit, when set, indicates that the Replay Timer has timed out. Time-out occurs when unacknowledge TLP have not received an acknowledgment (ACK) within the time-out period of (TBD).

[13] **PeDRF_AEANFES - Advisory Non-fatal Error Status**

This bit, when set, indicates that a Advisory Non-fatal Error has been detected. Advisory Non-Fatal errors are uncorrectable errors of NON-FATAL severity in which, if detected and AER is supported, ERR_COR Message will be transmitted instead of ERR_NONFATAL Message. These errors include Unsupported Requests, Bad ECRC, Unexpected Completion, Completer Abort, Completion Time-out, and Poisoned TLP. On the other hand, if the severity of these errors was set to FATAL, ERR_FATAL Message will be transmitted.

[31:14] **RSVD - Reserved**

25.3.1.7.6 PeDRF_AECEMAS: Correctable Error Mask Register

PCIe Offset: **0x1114**

Table 25.134: PeDRF_AECEMAS

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RWS	RWS	1'b0	PeDRF_AEREMAS	Receiver Error Mask
[05:01]	RO	RO	5'h0	RSVD	Reserved
[06]	RWS	RWS	1'b0	PeDRF_AEBTLPM	Bad TLP Mask
[07]	RWS	RWS	1'b0	PeDRF_AEBDLPM	Bad DLLP Mask
[08]	RWS	RWS	1'b0	PeDRF_AERNRMA	REPLAY_NUM Rollover Mask
[11:09]	RO	RO	3'h0	RSVD	Reserved
[12]	RWS	RWS	1'b0	PeDRF_AERTTMA	Replay Timer Timeout Mask
[13]	RWS	RWS	1'b1	ANEM	Advisory Non-fatal Error Mask
[31:14]	RO	RO	18'h0	RSVD	Reserved

[00] PeDRF_AEREMAS - Receiver Error Status

This bit, when set, prevents the reporting of a detected Receiver Error to the PCIe Root Complex.

[05:01] RSVD - Reserved

[06] PeDRF_AEBTLPM - Bad TLP Status

This bit, when set, prevents the reporting of a detected Bad TLP to the PCIe Root Complex.

[07] PeDRF_AEBDLPM - Bad DLLP Status

This bit, when set, prevents the reporting of a detected Bad DLLP to the PCIe Root Complex.

[08] PeDRF_AERNRMA - REPLAY_NUM Rollover Status

This bit, when set, prevents the reporting of a detected REPLAY_NUM Rollover to the PCIe Root Complex.

[11:09] RSVD - Reserved

[12] PeDRF_AERTTMA - Replay Timer Timeout Status

This bit, when set, prevents the reporting of a detected Replay Timer Timeout event to the PCIe Root Complex.

[13] **PeDRF_AEANFEM - Advisory Non-fatal Error Status**

This bit, when set, prevents the reporting of a detected Advisory Non-fatal Error to the PCIe Root Complex. This bit is set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.

[31:14] **RSVD - Reserved**

25.3.1.7.7 PeDRF_AECCTRL: Advanced Error Capabilities and Control Register

PCIe Offset: **0x1118**

Table 25.135: PeDRF_AECCTRL

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[04:00]	ROS	ROS	5'h00	PeDRF_AEFEPTR	First Error Pointer
[05]	RW	RO	1'b1	PeDRF_AEEGCAP	ECRC Generation Capable
[06]	RWS	RWS	1'b0	PeDRF_AEEGENA	ECRC Generation Enable
[07]	RW	RO	1'b1	PeDRF_AEECCAP	ECRC Check Capable
[08]	RWS	RWS	1'b0	PeDRF_AEECENA	ECRC Check Enable
[31:09]	RO	RO	23'h0	RSVD	Reserved

[04:00] PeDRF_AEFEPTR - First Error Pointer

This field identifies the bit position of the first error reported in the Uncorrectabe Error Status register.

[05] PeDRF_AEEGCAP - ECRC Generation Capable

This bit indicates that the device is capable of generating ECRC.

[06] PeDRF_AEEGENA - ECRC Generation Enable

This bit, when set, enables ECRC generation.

[07] PeDRF_AEECCAP - ECRC Check Capable

This bit indicates that the device is capable of checking ECRC.

[08] PeDRF_AEECENA - ECRC Check Enable

This bit, when set, enables ECRC checking.

[31:09] RSVD - Reserved

25.3.1.7.8 PeDRF_AEHDLOG: Header Log Registers

PCIe Offset: **0x111C - 0x1128**

Table 25.136: PeDRF_AEHDLOG

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RWS	ROS	32'h0	PeDRF_AEHW0LO	Header Log Register 0
[31:00]	RWS	ROS	32'h0	PeDRF_AEHW1LO	Header Log Register 1
[31:00]	RWS	ROS	32'h0	PeDRF_AEHW2LO	Header Log Register 2
[31:00]	RWS	ROS	32'h0	PeDRF_AEHW3LO	Header Log Register 3

[127:00] PeDRF_AEHDLOG - Header Log Registers

This 16-byte register captures the Header for the TLP corresponding to a detected error. For 12-byte headers, PeDRF_AEHW3LO is not used and has undefined value.

25.3.1.7.9 PeDRF_AERECOM: Root Error Command

PCIe Offset: **0x112C**

Table 25.137: PeDRF_AERECOM

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_AECERE	Correctable Error Reporting Enable
[01]	RW	RW	1'b0	PeDRF_AENERE	Non-Fatal Error Reporting Enable
[02]	RW	RW	1'b0	PeDRF_AEFERE	Fatal Error Reporting Enable
[31:03]	RO	RO	29'h0	RSVD	Reserved

[00] PeDRF_AECERE - Correctable Error Reporting Enable

This bit, when set, enables the generation of an interrupt when a Correctable error is reported by any of the devices in the heirarchy associated with this Root Port.

[01] PeDRF_AENERE - Non-Fatal Error Reporting Enable

This bit, when set, enables the generation of an interrupt when a Non-fatal error is reported by any of the devices in the heirarchy associated with this Root Port.

[02] PeDRF_AEFERE - Fatal Error Reporting Enable

This bit, when set, enables the generation of an interrupt when a Fatal error is reported by any of the devices in the heirarchy associated with this Root Port.

[31:03] RSVD - Reserved

25.3.1.7.10PeDRF_AERESTA: Root Error Status Register

PCIe Offset: **0x1130**

Table 25.138: PeDRF_AERESTA

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW1CS	RW1CS	1'b0	PeDRF_AERCESTA	Correctable Error Received
[01]	RW1CS	RW1CS	1'b0	PeDRF_AEMCESTA	Multiple Correctable Error Received
[02]	RW1CS	RW1CS	1'b0	PeDRF_AEFNFESTA	Fatal/Non-Fatal Error Received
[03]	RW1CS	RW1CS	1'b0	PeDRF_AEMFNFESTA	Multiple Fatal/Non-Fatal Error Received
[04]	RW1CS	RW1CS	1'b0	PeDRF_AEFUEFSTA	First Uncorrectable Error Fatal Received
[05]	RW1CS	RW1CS	1'b0	PeDRF_AENFESTA	Non-Fatal Error Received
[06]	RW1CS	RW1CS	1'b0	PeDRF_AEFESTA	Fatal Error Received
[26:07]	RO	RO	20'h0	RSVD	Reserved
[31:27]	RO	RO	5'h1	PeDRF_AEIMN	Interrupt Message Number

[00] PeDRF_AERCESTA - Correctable Error Status

This bit, when set, indicates that a correctable error is received (ERR_COR) or detected and this bit is not already set.

[01] PeDRF_AEMCESTA - Multiple Correctable Error Status

This bit, when set, indicates that a correctable error is received (ERR_COR) or detected and the Correctable Error Status bit is already set.

[02] PeDRF_AEFNFESTA - Fatal/Non-Fatal Error Status

This bit, when set, indicates that a non-fatal/fatal error is received (ERR_NONFATAL/ERR_FATAL) or detected and this bit is not already set.

[03] PeDRF_AEMFNFESTA - Multiple Fatal/Non-Fatal Error Status

This bit, when set, indicates that a non-fatal/fatal error is received (ERR_NONFATAL/ERR_FATAL) or detected and the Fatal/NonFatal Error Status bit is already set.

[04] PeDRF_AEFUEFSTA - First Uncorrectable Error Fatal Status

This bit, when set, indicates that the first Uncorrectable Error received or detected is for a Fatal error.

[05] PeDRF_AENFESTA - Non-Fatal Error Status

This bit, when set, indicates that one or more Non-Fatal Error have been received or detected.

[06] **PeDRF_AEFESTA - Fatal Error Status**

This bit, when set, indicates that one or more Fatal Error messages have been received or detected**.**

[26:07] **RSVD - Reserved**

[31:27] **PeDRF_AEIMN - Interrupt Message Number**

This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this capability.

25.3.1.7.11PeDRF_AEESOID: Error Source ID Register

PCIe Offset: **0x1134**

Table 25.139: PeDRF_AEESOID

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RWS	ROS	16'h0	PeDRF_AEECSID	ERR_COR Source ID
[31:16]	RWS	ROS	16'h0	PeDRF_AEEFSID	ERR_FATAL/NONFATAL Source ID

[15:00] PeDRF_AEECSID - ERR_COR Source Identification

This field is loaded with the Requestor ID indicated in the received ERR_COR Message when the ERR_COR Received bit (ECR) of the Root Error Status register is not already set. This field is read-only for both PCI host and local processor.

[31:16] PeDRF_AEEFSID - ERR_FATAL/NONFATAL Source Identification

This field is loaded with the Requestor ID indicated in the received ERR_FATAL or ERR_NONFATAL Message when the ERR_FATAL/ NONFATAL Received register is not already set. This field is read-only for both PCI host and local processor.

25.3.1.8 Device Serial Number Capability Registers

25.3.1.8.1 PeDRF_SNCPHDR: Device Serial Number Capability Header Register

PCIe Offset: **0x1138**

Table 25.140: PeDRF_SNCPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RO	RO	16'h0003	PeDRF_SNCAPID	Device Serial Number Capability ID
[19:16]	RO	RO	4'h1	PeDRF_SNCAPVE	Device Serial Number Capability Version
[31:20]	RW	RO	12'h144	PeDRF_SNNCAPP	Next Capability Offset

[15:00] PeDRF_SNCAPID - Device Serial Number Capability ID

PCI-SIG defines a value of 0003h for the Device Serial Number Capability structure. This field is read-only for both PCI host and local processor.

[19:16] PeDRF_SNCAPVE - Capability Version

The value of this field is 1h for the current version of the specification. This field is read-only for both PCI host and local processor.

[31:20] PeDRF_PBNCAPP - Next Capability Offset

This field contains the offset to the next PCIe capability structure or 000h is no other items exist in the linked list of capabilities. In this design, the next capability offset is the Power Budgeting Capability Header (0x1144).

25.3.1.8.2 PeDRF_SNDSNUM: Device Serial Number Register

PCIe Offset: **0x113C - 0x1140**

Table 25.141: PeDRF_SNDSNUM

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RO	32'h01234567	PeDRF_SNSN1DW	PCIe Device Serial Number 1
[63:32]	RW	RO	32'h89ABCDEF	PeDRF_SNSN2DW	PCIe Device Serial Number 2

[63:00] SERNUM - PCIe Device Serial Number

This field contains the IEEE defined 64-bit extended unique identifier (EUI-64 TM). This identifier includes a 24-bit company ID value assigned by IEEE registration authority and a 40-bit extension ID assigned by the manufacturer. This field is read-only for both PCI host and local processor.

25.3.1.9 Power Budgeting Capability Registers

25.3.1.9.1 PeDRF_PBCPHDR: Power Budgeting Capability Header Register

PCIe Offset: **0x1144**

Table 25.142: PeDRF_PBCPHDR

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:00]	RO	RO	16'h0004	PeDRF_PBCAPID	Power Budgeting Capability ID
[19:16]	RO	RO	4'b0001	PeDRF_PBCAPVE	Power Budgeting Version
[31:20]	RW	RO	12'h000	PeDRF_PBNCAPP	Next Capability Offset

[15:00] PeDRF_PBCAPID - Power Budgeting Capability ID

PCI-SIG defines a value of 0004h for the Power Budgeting Capability structure. This field is read-only for both PCI host and local processor.

[19:16] PeDRF_PBCAPVE - Power Budgeting Capability Version

The value of this field is 1h for the current version of the specification. This field is read-only for both PCI host and local processor.

[31:20] PeDRF_PBNCAPP - Next Capability Offset

This field contains the offset to the next PCIe capability structure or 000h is no other items exist in the linked list of capabilities.

25.3.1.9.2 PeDRF_PBDSELE: Power Budgeting Data Select Register

PCIe Offset: **0x1148**

Table 25.143: PeDRF_PBDSELE

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RW	RW	8'h00	PeDRF_PBDASEL	Data Select
[31:08]	RO	RO	24'b0	RSVD	Reserved

[07:00] PeDRF_PBDASEL - Data Select

This field indexes the Power Budgeting Data reported through the Data register and selects the 4-byte of Power Budgeting Data that should appear in the Data register. This field is read/write for both PCI host and local processor.

[31:08] RSVD - Reserved

25.3.1.9.3 PeDRF_PBDATAX: Power Budgeting Data Register

PCIe Offset: **0x114C**

Table 25.144: PeDRF_PBDATAX

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[07:00]	RO	RO	8'hx	PeDRF_PBBPOWE	Base Power
[09:08]	RW	RO	2'bxx	PeDRF_PBDSCAL	Data Scale
[12:10]	RW	RO	3'hx	PeDRF_PBPMSST	PM Sub State
[14:13]	RW	RO	2'bxx	PeDRF_PBPMSTA	PM State
[17:15]	RW	RO	3'hx	PeDRF_PBTYPEX	Type
[20:18]	RW	RO	3'hx	PeDRF_PBPRAIL	Power Rail
[31:21]	RO	RO	11'b0	RSVD	Reserved

[07:00] PeDRF_PBBPOWE - Base Power

This field specifies the base power value, in watts, in the given operating condition. The actual power consumption value is the value of this field multiplied by the data scale field value.

[09:08] PeDRF_PBDSCAL - Data Scale

The field specifies the scale value to be multiplied with the Base Power field value to get the actual base power consumption value.

Table 25.145: PeDRF_PBDSCAL Encoding

Value	Function
2'b00	1.0x
2'b01	0.1x
2'b10	0.01x
2'b11	0.001x

[12:10] PeDRF_PBPMSST - PM Sub State

This field specifies the power management sub state of the operating condition being described.

Table 25.146: PeDRF_PBPMSST Encoding

Value	Function
3'b000	Default Sub State
3'b001 - 3'b111	Device Specific Sub State

[14:13] PeDRF_PBPMSTA - PM State

This field specifies the power management state of the operating condition being described. Note that a device returns 2'b11 in this field and Aux or

PME Aux in the Type field to specify the D3-Cold PM State. An encoding of 2'b11 along with any other TYP value specifies the D3-Hot State.

Table 25.147: PeDRF_PBPMSTA Encoding

Value	Function
2'b00	D0
2'b01	D1
2'b10	D2
2'b11	D3

[17:15] PeDRF_PBTYPEX - Type

This field indicates the type of operating condition being described.

Table 25.148: TYP Encoding

Value	Function
3'b000	PME Aux
3'b001	Auxiliary
3'b010	Idle
3'b011	Sustained
3'b111	Maximum
others	Reserved

[20:18] PeDRF_PBPRAIL - Power Rail

This field indicates the power rail of the operating condition being described.

Table 25.149: PeDRF_PBPRAIL Encoding

Value	Function
3'b000	Power (12V)
3'b001	Power (3.3V)
3'b010	Power (1.8V)
3'b111	Thermal
others	Reserved

[31:21] RSVD - Reserved

25.3.1.9.4 PeDRF_PBCAPAB: Power Budgeting Capability Register

PCIe Offset: **0x1150**

Table 25.150: PeDRF_PBCAPAB

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_PBSALLO	System Allocated
[31:1]	RO	RO	31'b0	RSVD	Reserved

[00] PeDRF_PBSALLO - System Allocated

This bit, when set, indicates that the power budget for the device is included within the system power budget. Reported Power Budgeting Data for this device should be ignored by software for power budgeting decisions if this bit is set. This bit is hardware initialized.

[31:01] RSVD - Reserved

25.3.2 PeCORE Internal Registers

PeCORE internal registers are not accessible through PCI Configuration Request.

25.3.2.1 PeDRF_CIR - Core ID Register

PCIe Offset: **0x0000**

Table 25.151: PeDRF_CIR

Field	Type (LCB)	Default	Field Name	Field Description
[07:00]	RW	8'h01	PeDRF_INFOREV	Revision ID
[15:08]	RW	8'h01	PeDRF_INFOVER	Version ID
[31:16]	RW	16'h0007	PeDRF_INFORID	Core ID

[07:00] PeDRF_INFOREV - Revision ID

This field contains the revision number of PeCORE.

[15:08] PeDRF_INFOVER - Version ID

This field contains the version number of PeCORE.

[31:16] PeDRF_INFORID - Core ID

This field contains the standard core ID of PeCORE. This ID is assigned by BiTMICRO to PCI Express core (aka PeCORE).

25.3.2.2 PeDRF_COPT - CORE Options Register

PCIe Offset: **0x0004**

Table 25.152: PeDRF_COPT

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RO	1'b0	PeDRF_PBM	PeCORE Bridge Mode
[01]	RW	1'b0	PeDRF_HCE	Host Configuration Enable
[04:02]	RO	3'b000	RSVD	Reserved
[07:05]	RW	3'b000	PeDRF_BNFWD	Bus Number Field Width
[08]	RW	1'b1	PeDRF_DPTLP	Discard Poisoned TLP
[09]	RW	1'b1	PeDRF_RPTLP	Retry Poisoned TLP
[10]	RW	1'b1	PeDRF_RNTLP	Retry Nullified TLP
[11]	RW	1'b1	PeDRF_PTLPE	Poison TLP Enable
[12]	RW	1'b1	PeDRF_NTLPE	Nullify TLP Enable
[13]	RW	1'b0	PeDRF_URDIS	Unsupported Request Disable
[14]	RW	1'b0	PeDRF_CADIS	Completer Abort Disable
[15]	RW	1'b0	PeDRF_TXINSTRDIS	Tx Inst Retry Disable
[16]	RW	1'b0	PeDRF_TXINJRDIS	Tx Inject Retry Disable
[17]	RW	1'b0	PeDRF_LC1D	LC1 To Disable
[18]	RW	1'b0	PeDRF_LC2D	LC2 To Disable
[19]	RW	1'b0	PeDRF_CPLCR	Cmpletion Credit Return
[20]	RW	1'b0	PeDRF_AUSTEN	Auto Status Enable
[31:21]	RO	11'h0	RSVD	Reserved

[00] PeDRF_PBM - PeCORE Bridge Mode

0 - Bridge-Adapter mode. PeCORE functions as PCIe endpoint.

1 - Bridge-Host mode. PeCORE functions as PCIe Root Complex.

[01] PeDRF_HCE - Host Configuration Enable

If this bit is not set and PeCORE is in Endpoint mode, Configuration Accesses by the Host to PeCORE is retried, that is, PeCORE returns a Completion TLP with CRS completion status to the Host.

If set, it means that EDC or PeCORE is ready to receive Configuration request from the host.

This can be used by FW to signal to the Host when EDC is ready to be accessed by the Host.

[04:02] RSVD - Reserved

[07:05] PeDRF_BNFWD - Bus Number Field Width

This fields determines how many bits are used for the Bus Number Field of PeDRF_CRAR.

Table 25.153: Bus Number Field Width

Value	Bus Number Field Width
3'b000	8 bits
3'b001	1 bit
3'b010	2 bits
3'b011	3 bits
3'b100	4 bits
3'b101	5 bits
3'b110	6 bits
3'b111	7 bits

[08] PeDRF_DPTLP - Discard Poisoned TLP

When enabled, poisoned TLPs received by PeCORE are discarded. Otherwise, poisoned TLPs receive by PeCORE are treated as good TLPs.

[09] PeDRF_RPTLP - Retry Poisoned TLP

When enabled, poisoned TLPs sent by PeCORE are retried up to 3 times. After 3 retries, a "maximum retry reached" interrupt is sent to FW. If this bit is not set, poisoned TLPs sent by PeCORE are not retried.

[10] PeDRF_RNTLP - Retry Nullified TLP

When enabled, nullified TLPs sent by PeCORE are retried up to 3 times. Otherwise, nullified TLPs sent by PeCORE are not retried.

This feature applies only to MWr data. Completion data are always not retried.

[11] PeDRF_PTLPE - Poison TLP Enable

When enabled, TLPs are poisoned if there is error. Otherwise, TLPs are not poisoned even if there is error.

[12] PeDRF_NTLPE - Nullify TLP Enable

When set, TLPs with parity error sent by PeCORE are nullified. Otherwise, TLPs are not nullified even if parity error is detected.

[13] PeDRF_URDIS - Unsupported Request Disable

When set, detection and reporting of PeCORE of Unsupported Request (UR) and replying with CplStat=UR (if NP request) is disabled.

[14] PeDRF_CADIS - Completer Abort Disable

When set, detection and reporting of PeCORE of Completer Abort (CA) and replying with CplStat=CA (if NP request) is disabled.

[15] **PeDRF_TXINSTRDIS - Tx Inst Retry Disable**

When set, retry of Tx Instruction when parity error is detected while fetching it from the command buffer in IOB is disabled.

[16] **PeDRF_TXINJRDIS - Tx Inject Retry Disable**

When set, retry of Tx Inject (Instruction) when parity error is detected while fetching it from the command buffer in IOB is disabled.

[17] **PeDRF_CL1ToDIS -**

[18] **PeDRF_CL2ToDIS -**

[19] **PeDRF_CPLCR -**

[20] **PeDRF_AUSTEN -**

[31:21] **RSVD - Reserved**

25.3.2.3 PeDRF_CINT - CORE Interrupt Status Register

PCIe Offset: **0x0008**

Table 25.154: PeDRF_CINT

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RO	1'b0	RSVD	Reserved
[01]	RO	1'b0	PeDRF_MDNINT	Memory Request Done Interrupt
[02]	RO	1'b0	PeDRF_EMPINT	EMP Port Attention Interrupt
[03]	RO	1'b0	PeDRF_SYSTINT	System Interrupt
[04]	RO	1'b0	PeDRF_INTxINT	INTx Interrupt
[05]	RCW1	1'b0	PeDRF_MSIINT	MSI Interrupt
[06]	RO	1'b0	PeDRF_LNKBCINT	Link Bandwidth Change Interrupt
[08:07]	RO	2'h0	RSVD	Reserved
[09]	RCW1	1'b0	PeDRF_RBPERRINT	Link Retry Buffer Parity Error Interrupt
[10]	RCW1	1'b0	PeDRF_HBPERRINT	Rx Header Buffer Parity Error Interrupt
[11]	RCW1	1'b0	PeDRF_CBPERRINT	Tx Cmd Buffer Parity Error Interrupt
[16:12]	RO	5'h0	RSVD	Reserved
[17]	RW	1'b0	PeDRF_MDNINTM	Memory Request Done Interrupt Mask
[18]	RW	1'b0	PeDRF_EMPINTM	EMP Port Attention Interrupt Mask
[19]	RW	1'b0	PeDRF_SYSTINTM	System Interrupt Mask
[20]	RW	1'b0	PeDRF_INTxINTM	INTx Interrupt Mask
[21]	RW	1'b0	PeDRF_MSIINTM	MSI Interrupt Mask
[22]	RW	1'b0	PeDRF_LNKBCINTM	Link Bandwidth Change Interrupt Mask
[24:23]	RO	2'h0	RSVD	Reserved
[25]	RW	1'b0	PeDRF_RBPERRINTM	Link Retry Buffer Parity Error Int Mask
[26]	RW	1'b0	PeDRF_HBPERRINTM	Rx Header Buffer Parity Error Int Mask
[27]	RW	1'b0	PeDRF_CBPERRINTM	Tx Cmd Buffer Parity Error Int Mask
[31:28]	RO	4'h0	RSVD	Reserved

This register lists the summary of all the interrupt sources of interrupt line #42 and #169 of PeCORE. Once interrupt line number #42 or #169 from PeCORE is received, this register is the first interrupt register to read. From here, all other sources of interrupt can be traced.

[00] RSVD - Reserved

[01] PeDRF_MDNINT - Memory Request Done Interrupt Status

This bit is set if a CSR-based descriptor DMA request in any of the 4 (pseudo) DMA channels is completed, and its corresponding Interrupt Enable (PeDRF_MRCR[21]) is enabled. This has Read-Only access. While any of the 4 (pseudo) DMA Channels Done status (PeDRF_MRSR[0]) is set, this bit is set.

This bit is also set while bit 1, 2 or 6 of PeDRF_MRSR and their corresponding enable are set.

[02] **PeDRF_EMPINT - EMP Port Attention Interrupt Status**

While any of the bits in the EMP Port Attention Register (PeDRF_PATR) is set, this bit is set.

[03] **PeDRF_SYSTINT - System Interrupt Status**

While any of bits [5:0] of the System Interrupt Register (PeDRF_SYSINT[5:0]) is set, this bit is set.

[04] **PeDRF_INTxINT - INTx Interrupt Status**

This bit is set while any of the INTx status in PeDRF_SYSINT (bits [9:6]) is set.

[05] **PeDRF_MSIINT - MSI Interrupt Status**

This bit is set if PeCORE, in Root Complex mode, has detected an MSI.

[06] **PeDRF_LNKBCINT - Link Bandwidth Change Interrupt Status**

This bit is set if there is a change in either Link Speed or Link Width. A normal interrupt is generated.

[08:07] **RSVD - Reserved**

[09] **PeDRF_RBPERRINT - Link Retry Buffer Parity Error Status**

This bit is set when a parity error is detected while reading data in Link Retry Buffer. Automatic recovery will be done by the hardware up to four times. If error still occurs after four retries, only then this bit will be set.

[10] **PeDRF_HBPERRINT - Rx Header Buffer Parity Error Status**

This bit is set when a parity error is detected while reading data in Rx Header Buffer. Automatic recovery will be done by the hardware up to four times. If error still occurs after four retries, only then this bit will be set.

[11] **PeDRF_CBPERRINT - Tx Cmd Buffer Parity Error Status**

This bit is set when a parity error is detected while reading data in Tx Cmd Buffer. Automatic recovery will be done by the hardware up to four times. If error still occurs after four retries, only then this bit will be set.

[16:12] **RSVD - Reserved**

[17] **PeDRF_MDNINTM - Memory Request Done Interrupt Status Mask**

If set, this bit masks the Memory Request Done Interrupt.

[18] **PeDRF_EMPINTM - EMP Port Attention Interrupt Status Mask**

If set, this bit masks the EMP Port Attention Interrupt.

[19] **PeDRF_SYSTINTM - System Interrupt Status Mask**

If set, this bit masks the System Interrupt.

[20] **PeDRF_ERRINTM - Error Interrupt Status Mask**

If set, this bit masks the Error Interrupt.

[21] **PeDRF_MSIINTM - MSI Interrupt Status Mask**

If set, this bit masks the MSI Interrupt

[22] **PeDRF_LNKBCINTM - Link Bandwidth Change Interrupt Status Mask**

If set, this bit masks the Link Bandwidth Change Interrupt.

[24:23] **RSVD - Reserved**

[25] **PeDRF_RBPERRINTM - Link Retry Buffer Parity Error Interrupt Status Mask**

If set, this bit masks the Link Retry Buffer Parity Error Interrupt.

[26] **PeDRF_HBPERRINTM - Rx Header Buffer Parity Error Interrupt Status Mask**

If set, this bit masks the Rx Header Buffer Parity Error Interrupt.

[27] **PeDRF_CBPERRINTM - Tx Cmd Buffer Parity Error Interrupt Status Mask**

If set, this bit masks the Tx Cmd Buffer Parity Error Interrupt.

[31:28] **RSVD - Reserved**

25.3.2.4 PeDRF_SYSINT System Interrupt Register

PCIe Offset: **0x000C**

Table 25.155: PeDRF_SYSINT

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RO	1'b0	PeDRF_ERRCORINT	Correctable Error Interrupt Status
[01]	RO	1'b0	PeDRF_ERRFATINT	Fatal Error Interrupt Status
[02]	RO	1'b0	PeDRF_ERRNONFATINT	Non-Fatal Error Interrupt Status
[03]	RO	1'b0	PeDRF_PMEINT	PME Interrupt Status
[04]	RO	1'b0	RSVD	Reserved
[05]	RO	1'b0	PeDRF_SYSPCI	System PCI Interrupt Status
[06]	RO	1'b0	PeDRF_INTA	Interrupt A Status
[07]	RO	1'b0	PeDRF_INTB	Interrupt B Status
[08]	RO	1'b0	PeDRF_INTC	Interrupt C Status
[09]	RO	1'b0	PeDRF_INTD	Interrupt D Status
[31:10]	RO	22'h0	RSVD	Reserved

[00] PeDRF_ERRCORINT - Correctable Error Interrupt Status

This bit is set for any ERR_COR messages received from downstream devices or any correctable errors detected by the Root Port itself.

[01] PeDRF_ERRFATINT - Fatal Error Interrupt Status

This bit is set for any ERR_FATAL messages received from downstream devices or any uncorrectable (fatal) errors detected by the Root Port itself.

[02] PeDRF_ERRNONFATINT - Non-Fatal Error Interrupt Status

This bit is set for any ERR_NONFAT messages received from downstream devices or any uncorrectable (non-fatal) errors detected by the Root Port itself.

[03] PeDRF_PMEINT - PME Interrupt Status

This bit, when PERCCTL[3] is enabled, is set for any PME messages received from downstream devices.

[04] RSVD - Reserved

[05] PeDRF_SYSPCI - System PCI Interrupt Status

This bit is set while any of the bits of the Status Register (PeDRF_PCSTCMD[24,27:31]) is set.

[06] PeDRF_INTA - Interrupt A Status

This bit is set when an Assert_INTA message is received by the device, and cleared when a Deassert_INTA message is received.

[07] PeDRF_INTB - Interrupt B Status

This bit is set when an Assert_INTB message is received by the device, and cleared when a Deassert_INTB message is received.

[08] PeDRF_INTC - Interrupt C Status

This bit is set when an Assert_INTC message is received by the device, and cleared when a Deassert_INTC message is received.

[09] PeDRF_INTD - Interrupt D Status

This bit is set when an Assert_INTD message is received by the device, and cleared when a Deassert_INTD message is received.

[31:10] RSVD - Reserved

25.3.2.5 PeDRF_RXERR Receiver Error Status Register

PCIe Offset: **0x0010**

Table 25.156: PeDRF_RXERR

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RO	1'b0	PeDRF_DECERR	8b/10b Decoding Error
[01]	RO	1'b0	PeDRF_FRAERR	Framing Error
[02]	RO	1'b0	PeDRF_LSLERR	Loss of Symbol Lock Error
[03]	RO	1'b0	PeDRF_LDSERR	Link De-Skew Error
[04]	RO	1'b0	PeDRF_EBOERR	Elastic Buffer Overflow
[05]	RO	1'b0	PeDRF_EBUERR	Elastic Buffer UnderFlow
[06]	RO	1'b0	PeDRF_RDPERR	Receive Disparity Error
[31:07]	RO	25'h0	RSVD	Reserved

[00] PeDRF_DECERR - 8b/1b Decoding Error

This bit is the only receiver error that can cause Receiver Error bit of PeDRF_AECESTA to be set. This bit is set when a specific number of 8b/10b decode error occurrences is detected by PeCORE. That number of 8b/10b decode error can be programmed through PeDRF_DERRL.

This bit is automatically cleared when PeDRF_AECESTA[0] is cleared by FW.

[01] PeDRF_FRAERR - Framing Error

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[02] PeDRF_LSLERR - Loss of Symbol Lock

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[03] PeDRF_LDSERR - Link De-Skew Error

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[04] PeDRF_EBOERR - Elastic Buffer Overflow

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[05] PeDRF_EBUERR - Elastic Buffer Underflow

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[06] **PeDRF_RDPERR - Receive Disparity Error**

This bit does not cause Receiver Error bit of PeDRF_AECESTA to be set. No interrupt is generated if this bit is set.

[31:07] **RSVD - Reserved**

25.3.2.6 PeDRF_CACTL Completer Abort Control Register

PCIe Offset: **0x0014**

Table 25.157: PeDRF_CACTL

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_CADISTYP1	Completer Abort Type 1 Disable
[01]	RW	1'b0	PeDRF_CADISTYP2	Completer Abort Type 2 Disable
[02]	RW	1'b0	PeDRF_CADISTYP3	Completer Abort Type 3 Disable
[03]	RW	1'b0	PeDRF_CADISTYP4	Completer Abort Type 4 Disable
[31:04]	RO	28'h0	RSVD	Reserved

[00] PeDRF_CADISTYP1 - Completer Abort Type 1 Disable

When set, this bit disables detection and reporting of Completer Abort (CA) caused by memory accesses which are out of BAR's range.

[01] PeDRF_CADISTYP2 - Completer Abort Type 2 Disable

When set, this bit disables detection and reporting of Completer Abort (CA) caused by memory access to internal CSR with length not equal to 1.

[02] PeDRF_CADISTYP3 - Completer Abort Type 3 Disable

When set, this bit disables detection and reporting of Completer Abort (CA) caused by an access to non-existent AWB address.

[03] PeDRF_CADISTYP4 - Completer Abort Type 4 Disable

When set, this bit disables detection and reporting of Completer Abort (CA) caused by an IO access to a valid AWB address.

[31:04] RSVD - Reserved

25.3.2.7 PeDRF_URCTL Unsupported Request Control Register

PCIe Offset: **0x0018**

Table 25.158: PeDRF_URCTL

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_URDISTYP1	Unsupported Request Type 1 Disable
[01]	RW	1'b0	PeDRF_URDISTYP2	Unsupported Request Type 2 Disable
[02]	RW	1'b0	PeDRF_URDISTYP3	Unsupported Request Type 3 Disable
[03]	RW	1'b0	PeDRF_URDISTYP4	Unsupported Request Type 4 Disable
[04]	RW	1'b0	PeDRF_URDISTYP5	Unsupported Request Type 5 Disable
[31:05]	RO	27'h0	RSVD	Reserved

[00] PeDRF_URDISTYP1 - Unsupported Request Type 1 Disable

When set, this bit disables detection and reporting of Unsupported Request (UR) caused by received MRd Lk request.

[01] PeDRF_URDISTYP2 - Unsupported Request Type 2 Disable

When set, this bit disables detection and reporting of Unsupported Request (UR) caused by received Type 1 Config Request.

[02] PeDRF_URDISTYP3 - Unsupported Request Type 3 Disable

When set, this bit disables detection and reporting of Unsupported Request (UR) caused by an invalid MsgCode.

[03] PeDRF_URDISTYP4 - Unsupported Request Type 4 Disable

When set, this bit disables detection and reporting of Unsupported Request (UR) caused by no BAR-match.

[04] PeDRF_URDISTYP5 - Unsupported Request Type 5 Disable

When set, this bit disables detection and reportining of Unsupported Request (UR) caused by a memory request during D1 or D2 state.

[31:05] RSVD - Reserved

25.3.2.8 PeDRF_MALFCTL Malformed Detection Control Register

PCIe Offset: **0x001C**

Table 25.159: PeDRF_MALFCTL

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_MTDISTYP1	Malformed TLP Type 1 Disable
[01]	RW	1'b0	PeDRF_MTDISTYP2	Malformed TLP Type 2 Disable
[02]	RW	1'b0	PeDRF_MTDISTYP3	Malformed TLP Type 3 Disable
[03]	RW	1'b0	PeDRF_MTDISTYP4	Malformed TLP Type 4 Disable
[04]	RW	1'b0	PeDRF_MTDISTYP5	Malformed TLP Type 5 Disable
[05]	RW	1'b0	PeDRF_MTDISTYP6	Malformed TLP Type 6 Disable
[06]	RW	1'b0	PeDRF_MTDISTYP7	Malformed TLP Type 7 Disable
[07]	RW	1'b0	PeDRF_MTDISTYP8	Malformed TLP Type 8 Disable
[08]	RW	1'b0	PeDRF_MTDISTYP9	Malformed TLP Type 9 Disable
[31:09]	RO	23'h0	RSVD	Reserved

[00] PeDRF_MTDISTYP1 - Malformed TLP Type 1 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by TLP length greater than maximum payload size.

[01] PeDRF_MTDISTYP2 - Malformed TLP Type 2 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by an invalid FmType.

[02] PeDRF_MTDISTYP3 - Malformed TLP Type 3 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by an error in Config or IO TLP header.

[03] PeDRF_MTDISTYP4 - Malformed TLP Type 4 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by an error in byte enables.

[04] PeDRF_MTDISTYP5 - Malformed TLP Type 5 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by PeCORE receiving an INTx message while in Endpoint mode. Endpoint should never receive an INTx message. INTx messages should be issued only by an upstream device like an Endpoint.

[05] PeDRF_MTDISTYP6 - Malformed TLP Type 6 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by a 4KB-boundary error.

[06] PeDRF_MTDISTYP7 - Malformed TLP Type 7 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by Traffic Class (TC) errors.

[07] PeDRF_MTDISTYP8 - Malformed TLP Type 8 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by Address Type (AT) error.

[08] PeDRF_MTDISTYP9 - Malformed TLP Type 9 Disable

When set, this bit disables detection and reporting of Malformed TLP caused by TLP Length error.

[31:09] RSVD - Reserved

25.3.2.9 PeDRF_PDCR Pause DMA Control Register

PCIe Offset: **0x0020**

Table 25.160: PeDRF_PDCR

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_PDCRPR	Pause Request
[01]	RW	1'b1	PeDRF_PDCRAC	Pause Request Auto Clear
[31:02]	RO	30'h0	RSVD	Reserved

[00] PeDRF_PDCRPR - Pause Request

FW can use this bit to gracefully pause the data transfer of PeCORE during the flushing of ICBM.TxBuffer sequence. Once flushing of ICBM.TxBuffer is done, this bit is automatically cleared.

[01] PeDRF_PDCRAC - Pause Request Auto Clear

If this bit is set, pause request bit is auto-cleared when flushing of ICBM.TxBuffer is done.

[31:02] RSVD - Reserved

25.3.2.10 PeDRF_PDSR Pause DMA Status Register

PCIe Offset: **0x0024**

Table 25.161: PeDRF_PDSR

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_PDSRPAS	Pause Ack Status
[31:01]	RO	31'h0	RSVD	Reserved

[00] PeDRF_PDSRPAS - Pause Ack Status

This bit is set by PeCORE when it has acknowledged the Pause DMA request of FW.

[31:01] RSVD - Reserved

25.3.2.11 PeDRF_DERRL: 8b10b Decode Error Limit Register

PCIe Offset: **0x0028**

Table 25.162: PeDRF_DERRL

Field	Type (LCB)	Default	Field Name	Field Description
[11:00]	RW	12'h1e	PeDRF_DERRL	8b10b Decode Error Limit
[31:12]	RO	20'h0	RSVD	Reserved

[11:00] PeDRF_DERRL - 8b10b Decode Error Limit

Once the number of detected 8b10b decode errors has reached the count programmed in this register, Link Training is initiated and Receiver Error status (bit 0) of PeDRF_AECESTA is set.

[31:12] RSVD - Reserved

25.3.2.12 PeDRF_PHYERCTL: Physical Layer Error Control Register

PCIe Offset: **0x002C**

Table 25.163: PeDRF_PHYERCTL

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_MEERCTL	Missing END error control
[31:01]	RO	31'h0	RSVD	Reserved

[00] PeDRF_MEERCTL - Missing END error control

If this bit is set, detection of missing END error in Transaction Layer is disabled. This register is more of a workaround register. This is added just in case PeCORE incorrectly interprets a good packet as a packet with missing END.

[31:01] RSVD - Reserved

25.3.2.13 PeDRF_UPFC: Update FC Latency Register

PCIe Offset: **0x0030**

Table 25.164: PeDRF_UPFC

Field	Type (LCB)	Default	Field Name	Field Description
[02:00]	RW	3'h0	PeDRF_UPFC	UpdateFC latency
[3]	RW	1'b0	PeDRF_IFCDIS	InitFC Cpl Checking Disable
[31:04]	RO	28'h0	RSVD	Reserved

[02:00] PeDRF_UPFC - UpdateFC Latency

This field can be used to adjust the gap between sets of UpdateFC DLLPs. By default, UpdateFC latency is dependent on LinkWidth, LinkSpeed, and MaximumPayloadSize settings. This value, however, can still be adjusted through this field. See Table 25.165.

Table 25.165: UpdateFC Latency

Value	
3'b000	default
3'b001	multiply by 2
3'b010	multiply by 4
3'b011	multiply by 8
3'b100	divide by 2
3'b101	divide by 4
3'b110	divide by 8
3'b111	divide by 16

[03] PeDRF_IFCDIS - InitFC Cpl Chekcing Disable

If this bit is set, checking of InitFC DLLP for Cpl for possible Flow Control Protocol Error is disabled.

[31:04] RSVD - Reserved

25.3.2.14 PeDRF_CPMR: Completion Per MRd Register

PCIe Offset: **0x0034**

Table 25.166: PeDRF_CPMR

Field	Type (LCB)	Default	Field Name	Field Description
[05:00]	RW	6'h8	PeDRF_CPMR	Cpl Per MRd
[31:06]	RO	26'h0	RSVD	Reserved

[05:00] PeDRF_CPMR - Completion Per MRd

This field is used to set the expected maximum number of completions per MRd the completer will send. This is used to budget the actual total number of completion header credits of PeCORE. To avoid completion header credits overflow, this field must be set equal to or greater than the maximum number of completion the completer can send per MRd request.

Valid value is from 1 to 32.

[31:06] RSVD - Reserved

25.3.2.15 PeDRF_AKLAT: Ack/Nak Latency Control Register

PCIe Offset: **0x0038**

Table 25.167: PeDRF_AKLAT

Field	Type (LCB)	Default	Field Name	Field Description
[02:00]	RW	3'h0	PeDRF_AKLAT	Ack/Nak Latency
[31:03]	RO	29'h0	RSVD	Reserved

[02:00] PeDRF_AKLAT - AckNak Latency

Table 25.168: Ack/Nak Latency

Value	
3'b000	default
3'b001	divide by 2 (x2 faster)
3'b010	divide by 4 (x4 faster)
3'b011	divide by 8 (x8 faster)
3'b100	multiply by 2 (x2 slower)
3'b101	multiply by 4 (x4 slower)
3'b110	multiply by 8 (x8 slower)
3'b111	reserved

[31:03] RSVD - Reserved

25.3.2.16 PeDRF_CRAR: Configuration Request Address Register

PCIe Offset: **0x0040, 0x0CF8**

Table 25.169: PeDRF_CRAR

Field	Type (LCB)	Default	Field Name	Field Description
[01:00]	RO	2'b00	RSVD	Reserved
[07:02]	RW	6'h0	PeDRF_REGNUM	Register Number
[10:08]	RW	3'h0	PeDRF_FUNCNUM	Function Number
[15:11]	RW	5'h0	PeDRF_DEVNUM	Device Number
[23:16]	RW	8'h0	PeDRF_BUSNUM	Bus Number
[30:24]	RO	7'h0	RSVD	Reserved
[31]	RW	1'b0	PeDRF_CFGEN	Configuration Enable

[01:00] RSVD - Reserved

[07:02] PeDRF_REGNUM - Register Number

[10:08] PeDRF_FUNCNUM - Function Number

[15:11] PeDRF_DEVNUM - Device Number

[m:16] PeDRF_BUSNUM - Bus Number

m = 16 + (n-1) where n = PeDRF_COPT[7:5]. Note that when PeDRF_COPT[7:5] = 0, n = 8.

[30:m+1] RSVD - Reserved

[31] PeDRF_CFGEN - Configuration Enable

If this bit is set, access to PeDRF_CRDR triggers generation and transmission of CfgWr/CfgRd TLP.

Note that PeDRF_CRAR is also accessible through LCB address 0x0CF8.

25.3.2.17 PeDRF_CRDR: Configuration Request Data Register

PCIe Offset: **0x0044, 0x0CFC**

Table 25.170: PeDRF_CRDR

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RW	32'h00	PeDRF_CRDR	Host Configuration Data

[31:00] PeDRF_CRDR - Host Configuration Data

During write access, content of this register is the payload of a CfgWr TLP. During read access, Completion payload of a CfgRd/IORd request is saved here.

Note that PeDRF_CRDR is also accessible through LCB address 0x0CFC.

25.3.2.18 PeDRF_CRCR: Configuration Request Control Register

PCIe Offset: **0x0048**

Table 25.171: PeDRF_CRCR

Field	Type (LCB)	Default	Field Name	Field Description
[03:00]	RO	4'h0	RSVD	Reserved
[07:04]	RW	4'hf	PeDRF_HCFBE	Host Config First Byte Enable
[31:08]	RO	24'h0	RSVD	Reserved

[03:00] RSVD - Reserved

[07:04] PeDRF_HCFBE - Host Configuration First-Words Byte Enable

FW can set FBE field of Cfg and IO Request Header here.

[31:08] RSVD - Reserved

25.3.2.19 PeDRF_CRSR: Configuration Request Status Register

PCIe Offset: **0x004C**

Table 25.172: PeDRF_CRSR

Field	Type (LCB)	Default	Field Name	Field Description
[02:00]	RO	3'h7	PeDRF_HCCPLSTAT	Host Config Cpl Status Encoding
[03]	RW1C	1'b0	PeDRF_CFGMAXRETSTA	Config Request Max Retry Status
[31:04]	RO	28'h0	RSVD	Reserved

[02:00] PeDRF_HCCPLSTAT - Host Configuration Cpl Status Encoding

Table 25.173: Config CplStat

Value	CplStat
3'b000	SC (Successful Completion)
3'b001	UR (Unsupported Request)
3'b010	CRS (Config Request Retry Status)
3'b100	CA (Completer Abort)
others	Reserved

[03] PeDRF_CFGMAXRETSTA - Config Request Max Retry Status

This bit is set when maximum number of retry of Config or IO request due to Completion Timeout is reached.

[31:04] RSVD - Reserved

25.3.2.20 PeDRF_CRBAR: Configuration Request Base Address Register

PCIe Offset: **0x0050**

Table 25.174: PeDRF_CRBAR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[20:0]	RO	RO	21'h0	RSVD	Reserved
[31:21]	RW	RO	11'h0	PeDRF_CRBAR	Host Enhanced Config Base Address

[20:00] RSVD - Reserved

[31:21] PeDRF_CRBAR - Host Enhanced Config Base Address

This determines the base address ([31:20+n], where n = PeDRF_COPT[7:5]) of the memory space that is programmed for PCIe enhanced configuration requests.

25.3.2.21 PeDRF_MRLAR 0/1/2/3 Memory Request Local Address Register 0/1/2/3

PCIe Offset: **0x0100, 0x0104, 0x0108, 0x010C**

Table 25.175: PeDRF_MRLAR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_MRT	Memory Request Type
[01]	RO	RO	1'b0	RSVD	Reserved
[31:02]	RW	RW	30'h0	PeDRF_MRLAR	Memory Request Local Address Register

[00] PeDRF_MRT - Memory Request Type

This bit is writable but always returns 0 when read.

Table 25.176: Memory Request Type

Value	Request Type
1'b0	Memory Read
1'b1	Memory Write

[01] RSVD - Reserved

[31:02] PeDRF_MRLAR - Memory Request Local Address Register

If the request is a MRd, this register should be programmed to the destination address of the data payload of the MRd Completion. If the request is a MWr, this register should be programmed to the source of the data payload of the MWr request.

25.3.2.22 PeDRF_MRALR 0/1/2/3: Memory Request Address Low Register 0/1/2/3

PCIe Offset: **0x0110, 0x0114, 0x0118, 0x011C**

Table 25.177: PeDRF_MRALR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[01:00]	RO	RO	2'b0	RSVD	Reserved
[31:02]	RW	RW	30'h0	PeDRF_MRALR	Mem Request Address Low

[01:00] RSVD - Reserved

[31:02] PeDRF_MRALR - Memory Request Address Low Register

This register should be programmed by FW to the lower 32-bit address (bits [1:0]=00) of the PCI address of the memory request.

25.3.2.23 PeDRF_MRAHR 0/1/2/3: Memory Request Address High Register 0/1/2/3

PCIe Offset: **0x0120, 0x0124, 0x0128, 0x012C**

Table 25.178: PeDRF_MRAHR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:00]	RW	RW	32'h0	PeDRF_MRAHR	Mem Request Address High

[31:00] PeDRF_MRAHR - Memory Request Address High

If 64-bit addressing mode (PeDRF_MRCR[0]=1) is enabled, this register is the upper 32-bit PCI address of the memory request.

25.3.2.24 PeDRF_MRCR 0/1/2/3: Memory Request Control Register 0/1/2/3

PCIe Offset: **0x0130, 0x0134, 0x0138, 0x013C**

Table 25.179: PeDRF_MRCR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RW	1'b0	PeDRF_MRAM	Mem Request Addressing Mode
[01]	RW	RW	1'b0	PeDRF_MRM	Mem Request Mode
[02]	RW	RW	1'b0	PeDRF_MRMAXRETINTEN	Mem Request Maximum Retry Interrupt Enable
[03]	RW	RW	1'b0	PeDRF_DMAEN	Dma Channel Enable
[11:04]	RO	RO	8'h0	RSVD	Reserved
[14:12]	RW	RW	3'h0	PeDRF_MRBSZ	Mem Request Burst Size
[15]	RW	RW	1'b0	PeDRF_MRLNKEN	Mem Request Link Enable
[17:16]	RW	RW	2'b00	PeDRF_MRLNKNXT	Mem Request Link Next
[18]	RW	RW	1'b0	PeDRF_MRSTRPEN	Mem Request Striping Enable
[19]	RW	RW	1'b0	PeDRF_MRLEINTEN	Mem Request Link Error Int Enable
[20]	RW	RW	1'b0	PeDRF_MRBNINTEN	Mem Request Buzz Next Int Enable
[21]	RW	RW	1'b0	PeDRF_MRDNINTEN	Mem Request Done Int Enable
[23:22]	RW	RW	2'b00	PeDRF_MRDNTOLIM	Mem Request Done TO Limit
[27:24]	RW	RW	4'h0	PeDRF_MRFBE	Mem Request FBE (First Byte Enable)
[31:28]	RW	RW	4'h0	PeDRF_MRLBE	Mem Request LBE (Last Byte Enable)

[00] PeDRF_MRAM - Memory Request Addressing Mode

Table 25.180: Addressing Mode

Value	Addressing Mode
1'b0	32bit addressing mode
1'b1	64bit addressing mode

[01] PeDRF_MRM - Memory Request Mode

Table 25.181: Request Mode

Value	Completion Direction
1'b0	During MRd, data is returned through PeDRF_MRDR registers
	During MWr, data is transmitted from PeDRF_MRDR registers
1'b1	During MRd, data is returned through ICBM.RxBUF
	During MWr, data is transmitted from ICBM.TxBUF

[02] PeDRF_MRMAXRETINTEN - Memory Request Maximum Retry Interrupt Enable

When set, this bit enables interrupt due to “maximum retry reached” status of PeDRF_MRSR.

[03] PeDRF_DMAEN - Dma Channel Enable

This bit is not used.

[11:04] RSVD - Reserved

[14:12] PeDRF_MRBSZ - Memory Request Burst Size

Table 25.182: Burst Size

Value	Burst Size
3’d0	128B
3’d1	256B
3’d2	512B
3’d3	1KB
3’d4	2KB
3’d5	4KB
others	reserved

[15] PeDRF_MRLNKEN - Memory Request Link Enable

If this bit is set, a DMA request will not be started unless the previous DMA request from another DMA channel linked to this DMA is completed. In the same way, the next DMA request from another DMA channel pointed to by Memory Request Link Next will not be started unless this DMA request is completed.

[17:16] PeDRF_MRLNKNXT - Memory Request Link Next

Table 25.183: Link Next

Value	Link Next
2’d0	Next DMA is DMA Channel 0
3’d1	Next DMA is DMA Channel 1
3’d2	Next DMA is DMA Channel 2
3’d3	Next DMA is DMA Channel 3

[18] PeDRF_MRSTRPEN - Memory Request Striping Enable

If this bit is set, a DMA request is chunked into multiple smaller request with size equal to Burst Size. In this way, after the first burst of a DMA

request is completed, the next DMA request has the chance to be processed. When Striping is disabled, the next DMA request has to wait for the current DMA request length (PeDRF_MRLR) to be exhausted before it is processed.

[19] PeDRF_MRLEINTEN - Memory Request Link Error Interrupt Enable

This enables DMA Linking/Chaining Error Interrupt.

[20] PeDRF_MRBNINTEN - Memory Request Buzz Next Interrupt Enable

This enables Buzz Next Interrupt.

[21] PeDRF_MRDNINTEN - Memory Request Done Interrupt Enable

This enables Memory Request Done interrupt.

[23:22] PeDRF_MRMRDNTOLIM - Memory Request Done Timeout Limit

Table 25.184: Mem Request Done Timeout Limit

Value	Timeout
2'd0	no timeout
2'd1	32 clock cycles
2'd2	64 clock cycles
2'd3	128 clock cycles

[27:24] PeDRF_MRFBE - Memory Request First Byte Enable

This is the byte enable of the Memory Request first dword.

[31:28] PeDRF_MRLBE - Memory Request Last Byte Enable

This is the byte enable of the Memory Request last dword.

25.3.2.25 PeDRF_MRSR 0/1/2/3: Memory Request Status Register 0/1/2/3

PCIe Offset: **0x0140, 0x0144, 0x0148, 0x014C**

Table 25.185: PeDRF_MRSR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RWC1	RWC1	1'b1	PeDRF_MRDS	Mem Request Done
[01]	RWC1	RWC1	1'b0	PeDRF_MRBZNXT	Buzz Next DMA
[02]	RWC1	RWC1	1'b0	PeDRF_MRLKER	DMA Chaining/Linking Error
[05:03]	RO	RO	3'b0	PeDRF_MRCPLSTAT	Completion Status
[06]	RWC1	RWC1	1'b0	PeDRF_MRMXRTRY	Maximum Retry
[31:07]	RO	RO	25'h0	RSVD	Reserved

[00] PeDRF_MRDS - Memory Request Done Status

This bit, if asserted, indicates that the Memory Request is completed. The status of this bit can be checked through polling or interrupt. When polled through LCB read access, the access will be acknowledged only when the status is 1. Timeout limit, however, can be set through Memory Request Done Timeout Limit (PeDRF_MRCR[23:22]). If the timeout limit is reached, even if the status is still 0, the LCB read access is acknowledged. To check this bit through interrupt, Memory Request Done Interrupt Enable (PeDRF_MRCR[21]) must be enabled.

[01] PeDRF_MRBZNXT - Buzz Next DMA

If the current DMA channel is done processing a request and the next DMA channel (assuming Chaining/Linking is enabled) is not yet ready, this bit is set to interrupt the processor.

[02] PeDRF_MRLKER - DMA Chaining/Linking Error

If the DMA Chaining configuration is invalid, this bit is set. An invalid DMA Chaining configuration is when more than 2 DMA channels point to the same DMA channel as their Link Next.

[05:03] PeDRF_MRCPLSTAT - MRd Completion Status

The CplStat field of the MRd Completion TLP is returned is this field.

[06] PeDRF_MRMXRTRY - Maximum Retry

This bit is set when the memory request has been retried 3 times due to Completion Timeout (if MRd) or due parity error (if MWr) detected in the data payload.

[31:07] RSVD - Reserved

25.3.2.26 PeDRF_MRLR 0/1/2/3: Memory Request Length Register 0/1/2/3

PCIe Offset: **0x0150, 0x0154, 0x0158, 0x015C**

Table 25.186: PeDRF_MRLR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[11:00]	RW	RW	12'h1	PeDRF_MRLR	Mem Request Length
[31:12]	RO	RO	20'h0	RSVD	Reserved

[11:00] PeDRF_MRLR0/1/2/3 - Memory Request Length

When doing CSR-DMA, this field should be programmed to the transfer size of the request. Unit is in word.

[31:12] RSVD - Reserved

25.3.2.27 PeDRF_MRDR 0/1/2/3: Memory Request Data Register 0/1/2/3

PCIe Offset: **0x0160, 0x0164, 0x0168, 0x016C**

Table 25.187: PeDRF_MRDR

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RW	32'h0	PeDRF_MRDR	Mem Request Data

[31:00] PeDRF_MRDR0/1/2/3 - Memory Request Request Data

During MWr, the content of these registers is sent as the payload of MWr TLP, while during MRd, the payload of the received Completion TLP is saved in these registers. To access these registers without complications, set the corresponding PeDRF_MRCR[1] to 1.

25.3.2.28 PeDRF_TDMT: Tx Data Minimum Threshold Register

PCIe Offset: **0x0300**

Table 25.188: PeDRF_TDMT

Field	Type (LCB)	Default	Field Name	Field Description
[10:00]	RW	11'h20	PeDRF_TDMT	Tx Data Minimum Threshold
[31:11]	RO	21'h0	RSVD	Reserved

[10:00] PeDRF_TDMT - Tx Data Minimum Threshold

When transmitting Tx Data (MWr), the total memory request size is normally not sent in one burst or TLP. Data are immediately transmitted as long as enough data is available from Tx Buffer. The minimum threshold when the transmission will be started can be set through this register. The initial value of this register is 32 dwords (128 bytes). It means that even if the request is say, 4KB, as long as there is at least 128 bytes available in the Tx Buffer, transmission is started.

[31:11] RSVD - Reserved

25.3.2.29 PeDRF_MORP: Maximum Outstanding Request Partition Register

PCIe Offset: **0x0304**

Table 25.189: PeDRF_MORP

Field	Type (LCB)	Default	Field Name	Field Description
[05:00]	RW	6'h20	PeDRF_TXINST	Tx Inst Max Outstanding Request
[07:06]	RO	2'h0	RSVD	Reserved
[13:08]	RW	6'h00	PeDRF_TXINJ	Tx Inject Max Outstanding Request
[15:14]	RO	2'b00	RSVD	Reserved
[16]	RW	1'b0	PeDRF_MORPEN	Max Outstanding Request Partition Enable
[31:17]	RO	15'h0	RSVD	Reserved

[05:00] PeDRF_TXINST - Tx Inst Maximum Outstanding Request

Request from ICBM can come from two paths: TxInst and TxInject. Both these paths must share the maximum outstanding request of PeCORE, which is 32. How much of that is allocated to TxInst path can be programmed through this field.

[07:06] RSVD - Reserved

[13:08] PeDRF_TXINJ - Tx Inject Maximum Outstanding Request

How much of PeCORE's maximum outstanding request is allocated to TxInject path can be programmed through this field.

[15:14] RSVD - Reserved

[16] PeDRF_MORPEN - Maximum Outstanding Request Partition Enable

TBD

[31:14] RSVD - Reserved

25.3.2.30 PeDRF_CPP0 to CPP7: EMP Command Put Pointer 0 to 7

PCIe Offset: **0x0400 to 0x041C**

Table 25.190: PeDRF_CPP0 to PeDRF_CPP7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:0]	RO	RW	16'd0	PeDRF_CPP0 to 7	Command Put Pointer 0 to 7
[31:16]	RO	RO	16'd0	RSVD	Reserved

[15:0] PeDRF_CPP0 to PeDRF_CPP7 - Command Put Pointer 0 to 7

[31:16] RSVD - Reserved

25.3.2.31 PeDRF_CGP0 to CGP7: EMP Command Get Pointer 0 to 7

PCIe Offset: **0x0420 to 0x043C**

Table 25.191: PeDRF_CGP0 to PeDRF_CGP7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:0]	RW	RO	16'd0	PeDRF_CGP0 to 7	Command Get Pointer 0 to 7
[31:16]	RO	RO	16'd0	RSVD	Reserved

[15:0] PeDRF_CGP0 to PeDRF_CGP7 - Command Get Pointer 0 to 7

[31:16] RSVD - Reserved

25.3.2.32 PeDRF_RPP0 to RPP7: EMP Response Put Pointer 0 to 7

PCIe Offset: **0x0440 to 0x045C**

Table 25.192: PeDRF_RPP0 to PeDRF_RPP7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:0]	RW	RO	16'd0	PeDRF_RPP0 to 7	Response Put Pointer 0 to 7
[31:16]	RO	RO	16'd0	RSVD	Reserved

[15:0] PeDRF_RPP0 to PeDRF_RPP7 - Response Put Pointer 0 to 7

[31:16] RSVD - Reserved

25.3.2.33 PeDRF_RGP0 to RGP7: EMP Response Get Pointer 0 to 7

PCIe Offset: **0x0460 to 0x047C**

Table 25.193: PeDRF_RGP0 to PeDRF_RGP7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:0]	RO	RW	16'd0	PeDRF_RGP0 to 7	Response Get Pointer 0 to 7
[31:16]	RO	RO	16'd0	RSVD	Reserved

[15:0] PeDRF_RGP0 to PeDRF_RGP7 - Response Get Pointer 0 to 7

[31:16] RSVD - Reserved

25.3.2.34 PeDRF_HATR: EMP Host Attention Register

PCIe Offset: **0x0480**

Table 25.194: PeDRF_HATR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RS	RW1C	8'd0	PeDRF_HRRSPERQ	Ring[7:0] Response Empty
[15:8]	RS	RW1C	8'd0	PeDRF_HRCMDE	Ring[7:0] Command Empty
[23:16]	RS	RW1C	8'd0	PeDRF_HRINGATN	Ring[7:0] Attention
[29:24]	RS	RW1C	6'd0	RSVD	Reserved Host Attention
[30]	RS	RW1C	1'b0	PeDRF_HMBXATN	Mailbox Attention
[31]	RS	RW1C	1'b0	PeDRF_HERRATN	Error Attention

*RS = Readable, Settable (Can write 1 but not 0. Writing 0 has no effect)

[7:0] PeDRF_HRRSPERQ - Ring [7:0] Response Empty Request

This is set by the Port when the Response Ring is full, and it wants notification from the Host that it (Host) has emptied an entry in the Response Ring.

[15:8] PeDRF_HRCMDE - Ring [7:0] Command Empty

This is set by the Port when after receiving a "Command Ring Empty Notification" request from the Host, it has emptied at least one entry in the Command Ring.

The Host will request for a "Command Ring Empty Notification" from Port after the Host has determined that the Command Ring is full. It's like Host saying to Port "Hey, Port! The Command Ring is now full. Notify me if you have emptied at least one entry, so I can start giving you some more commands."

[23:16] PeDRF_HRINGATN - Ring [7:0] Attention

This bit is automatically set when Response Put Pointer is updated by the Port.

Aside from the automatic assertion of this bit due to the updating of Response Put Pointer, this bit is also usually set by FW after setting Ring Response Empty Request bit or Ring Command Empty bit.

[29:24] RSVD - Reserved Host Attention

Can be used by FW for future "host attention" requirements.

[30] PeDRF_HMBXATN - Mailbox Attention

This is set by the Port after updating the mailbox command, signaling Host that it is the Host's turn to process the mailbox command.

[31] **PeDRF_HERRATN - Error Attention**

This is set by the Port when it intends to send an error attention to the Host.

25.3.2.35 PeDRF_PATR: EMP Port Attention Register

PCIe Offset: **0x0484**

Table 25.195: PeDRF_PATR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW1C	RS	8'd0	PeDRF_PRCMDERQ	Ring[7:0] Command Empty
[15:8]	RW1C	RS	8'd0	PeDRF_PRRSPE	Ring[7:0] Response Empty
[23:16]	RW1C	RS	8'd0	PeDRF_PRINGATN	Ring[7:0] Attention
[27:24]	RW	RO	4'd0	PeDRF_ATNPRIO	Attention Priority Encoding
[29:28]	RO	RS	2'd0	RSVD	Reserved Attention
[30]	RW1C	RS	1'b0	PeDRF_PMBXATN	Mailbox Attention
[31]	RO	RS	1'b0	RSVD	Reserved Attention

*RS = Readable, Settable (Can write 1 but not 0. Writing 0 has no effect)

[7:0] PeDRF_PRCMDERQ - Ring [7:0] Command Empty Request

This is set by the Host when the Command Ring is full, and it wants notification from the Port that it (Port) has emptied an entry in the Command Ring.

[15:8] PeDRF_PRRSPE - Ring [7:0] Response Empty

This is set by the Host when after receiving a "Response Ring Empty Notification" request from the Port, it (Host) has emptied at least one entry in the Response Ring.

The Port will request for a "Response Ring Empty Notification" from Host after the Port has determined that the Response Ring is full. It's like Port saying to Host "Hey, Host! The Response Ring is now full. Notify me if you have emptied at least one entry, so I can start giving you some more responses."

[23:16] PeDRF_PRINGATN - Ring [7:0] Attention

This bit is automatically set when Command Put Pointer is updated by the Host.

Aside from the automatic assertion of this bit due to the updating of Command Put Pointer, this bit is also usually set by FW after setting Ring Command Empty Request bit or Ring Response Empty bit.

[27:24] PeDRF_ATNPRIO - Attention Priority Encoding

This field sets the priority of Mailbox and the eight Command/Response Rings

Table 25.196: Priority Encoding

Value	Priority
4'b0000	Mailbox
4'b0001	Ring 0
4'b0010	Ring 1
4'b0011	Ring 2
4'b0100	Ring 3
4'b0101	Ring 4
4'b0110	Ring 5
4'b0111	Ring 6
4'b1000	Ring 7
others	Reserved

[29:28] RSVD - Reserved Port Attention

Can be used by FW for future "port attention" requirement.

[30] PeDRF_PMBXATN - Mailbox Attention

This is set by the Host after updating the mailbox command, signaling Port that it is the Port's turn to process the mailbox command.

[31] RSVD - Reserved Port Attention

Can be used by FW for future "port attention" requirement.

25.3.2.36 PeDRF_PSTR: EMP Port Status Register

PCIe Offset: **0x0488**

Table 25.197: PeDRF_PATR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW	RW1C	8'd0	PeDRF_PRINGERR	Ring[7:0] Error
[21:8]	RO	RO	14'd0	RSVD	Reserved
[22]	RW	RO	1'b0	PeDRF_PMBXRDY	Mailbox Ready
[23]	RS	RO	1'b0	PeDRF_PRDY	Port Ready
[24]	RW	RO	1'b0	PeDRF_PERR8	Port Error 8
[25]	RW	RO	1'b0	PeDRF_PERR7	Port Error 7
[26]	RW	RO	1'b0	PeDRF_PERR6	Port Error 6
[27]	RW	RO	1'b0	PeDRF_PERR5	Port Error 5
[28]	RW	RO	1'b0	PeDRF_PERR4	Port Error 4
[29]	RW	RO	1'b0	PeDRF_PERR3	Port Error 3
[30]	RW	RO	1'b0	PeDRF_PERR2	Port Error 2
[31]	RW	RO	1'b0	PeDRF_PERR1	Port Error 1

*RS = Readable, Settable (Can write 1 but not 0. Writing 0 has no effect)

[7:0] PeDRF_PRINGERR - Ring [7:0] Error

[21:8] RSVD - Reserved

[22] PeDRF_PMBXRDY - Mailbox Ready

[23] PeDRF_PRDY - Port Ready

[24] PeDRF_PERR8 - Port Error 8
[25] PeDRF_PERR7 - Port Error 7
[26] PeDRF_PERR6 - Port Error 6
[27] PeDRF_PERR5 - Port Error 5
[28] PeDRF_PERR4 - Port Error 4
[29] PeDRF_PERR3 - Port Error 3
[30] PeDRF_PERR2 - Port Error 2
[31] PeDRF_PERR1 - Port Error 1

25.3.2.37 PeDRF_PCLR: EMP Port Control Register

PCIe Offset: **0x048C**

Table 25.198: PeDRF_PCLR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RO	RW	8'd0	PeDRF_PRINTEN	Ring[7:0] Interrupt Enable
[13:8]	RO	RW	6'h0	PeDRF_RSVDINTEN	Reserved Host Attention Interrupt Enable
[15:14]	RO	RO	2'd0	RSVD	Reserved
[23:16]	RW	RW	8'h0	PeDRF_RAOR	Ring Attention OR-ed
[25:24]	RO	RO	2'h0	RSVD	Reserved
[26]	RO	RW	1'b0	PeDRF_INITMBX	Initialize Mailbox
[27]	RO	RW	1'b0	PeDRF_INITPORT	Initialize Port
[30:28]	RO	RO	4'd0	RSVD	Reserved
[31]	RO	RW	1'b0	PeDRF_PEINTEN	Error Interrupt Enable

[7:0] PeDRF_PRINTEN - Ring [7:0] Interrupt Enable

If this bit is set, an interrupt (MSI or INTx) is sent to the Host if the corresponding Ring Attention bit in the Host Attention Register is set.

[13:8] PeDRF_RSVDINTEN - Reserved Host Attention Interrupt Enable

If this bit is set, an interrupt is sent to the Host if the corresponding Reserved Host Attention bit in the Host Attention Register is set.

[15:14] RSVD - Reserved

[23:16] PeDRF_RAOR - Ring Attention OR-ed

If this bit is set, even if the Ring Command Empty Request bit or the Ring Response Empty bit is set and their corresponding interrupt enable is set, no interrupt will be sent to the Host for that Ring Command Empty Request bit or Ring Response Empty bit that is set. It is assumed that an interrupt (MSI or INTx) for Ring Command Empty Request and Ring Response Empty will be sent to the Host through Ring Attention bit. It can be pictured as if the Ring Command Empty Request bit, the Ring Response Empty bit, and the setting of Ring Attention bit due to the updating of Response Put Pointer are ORed as one source to Ring Attention bit of the Host Attention Register.

If this bit is cleared, an interrupt (MSI or INTx) is sent to the Host if the Ring Command Empty Request bit or the Ring Response Empty bit and their corresponding interrupt enable are set.

[25:24] RSVD - Reserved

[26] **PeDRF_INITMBX - Initialize Mailbox**

[27] **PeDRF_INITPORT - Initialize Port**

[30:28] **RSVD - Reserved**

[31] **PeDRF_PEINTEN - Error Interrupt Enable**

25.3.2.38 PeDRF_CAPD: Credit Allocation - Posted Data Register

PCIe Offset: **0x0600**

Table 25.199: PeDRF_CAPD

Field	Type (LCB)	Default	Field Name	Field Description
[11:00]	RO	12'hXXX	PeDRF_CAPD	Credit Allocation - Posted Data
[31:12]	RO	20'h0	RSVD	Reserved

[11:00] PeDRF_CAPD - Credit Allocation Posted Data

Data credit allocation for Posted request (MWr) can be read through this field.

PeCORE can hold upto 1024 data credits. These data credits are shared by Posted and Completion Data. How these data credits are partitioned between the two is determined by the strap signal FC. Refer to Table 25.200.

Table 25.200: Posted Data Credit Partition through FC

Value	Priority
3'b000	PD = 512, CPLD = 512
3'b001	PD = 768, CPLD = 256
3'b010	PD = 256, CPLD = 768
3'b011	PD = 922, CPLD = 102
3'b100	PD = 102, CPLD = 922
3'b101	PD = 512, CPLD = 512
3'b110	PD = 768, CPLD = 102
3'b111	PD = 102, CPLD = 768

[31:12] RSVD - Reserved

25.3.2.39 PeDRF_CAPH: Credit Allocation - Posted Header Register

PCIe Offset: **0x0604**

Table 25.201: PeDRF_CAPH

Field	Type (LCB)	Default	Field Name	Field Description
[07:00]	RO	8'hXX	PeDRF_CAPH	Credit Allocation - Posted Header
[31:08]	RO	24'h0	RSVD	Reserved

[07:00] PeDRF_CAPH - Credit Allocation Posted Header

Header credit allocation for Posted request (MWr) can be read through this field.

PeCORE can hold upto 64 header credits to be shared between posted request and completion TLPs. How these header credits are partitioned between the two is determined by the strap signal FC. Refer to Table 25.202.

Table 25.202: Posted Header Credit Partition through FC

Value	Priority
3'b000	PH = 32, CPLH = 32*
3'b001	PH = 32, CPLH = 32*
3'b010	PH = 32, CPLH = 32*
3'b011	PH = 32, CPLH = 32*
3'b100	PH = 32, CPLH = 32*
3'b101	PH = 32, CPLH = 32
3'b110	PH = 48, CPLH = 16
3'b111	PH = 16, CPLH = 48
	*actual is 32, but 0 (infinite) is advertised during FC Init

[31:08] RSVD - Reserved

25.3.2.40 PeDRF_CAND: Credit Allocation - Non-Posted Data Register

PCIe Offset: **0x0608**

Table 25.203: PeDRF_CAND

Field	Type (LCB)	Default	Field Name	Field Description
[11:00]	RO	12'h40	PeDRF_CAND	Credit Allocation - Non-Posted Data
[31:12]	RO	20'h0	RSVD	Reserved

[11:00] PeDRF_CAND - Credit Allocation Non-Posted Data

Data credit allocation for non-posted request can be read through this field. It is fixed to 64.

[31:12] RSVD - Reserved

25.3.2.41 PeDRF_CANH: Credit Allocation - Non-Posted Header Register

PCIe Offset: **0x060C**

Table 25.204: PeDRF_CANH

Field	Type (LCB)	Default	Field Name	Field Description
[07:00]	RO	8'h40	PeDRF_CANH	Credit Allocation - Non-Posted Header
[31:08]	RO	24'h0	RSVD	Reserved

[07:00] PeDRF_CANH - Credit Allocation Non-Posted Header

Header credit allocation for non-posted request can be read through this field. It is fixed to 64.

[31:08] RSVD - Reserved

25.3.2.42 PeDRF_CACD: Credit Allocation - Completion Data Register

PCIe Offset: **0x0610**

Table 25.205: PeDRF_CACD

Field	Type (LCB)	Default	Field Name	Field Description
[11:00]	RO	12'h000	PeDRF_CACD	Credit Allocation - Completion Data
[31:12]	RO	20'h0	RSVD	Reserved

[11:00] PeDRF_CACD - Credit Allocation Completion Data

Data credit allocation for Completion TLPs can be read through this field.

Non-Posted requests (those requiring Completions) are not issued unless the device has enough credits to receive the Completions. Because of this requirement in the transmission of Non-Posted requests, data credits for Completions are considered infinite, hence the fixed value 0.

However, since credits for Completion Data in reality are finite, some credit calculation for Completion Data is still necessary. Completion Data still shares finite data credits with Posted Data. This"actual" Credit allocation for Completion Data is set through strap signal FC. Refer to Table 25.205

Table 25.206: Completion Data Credit Partition through FC

Value	Priority
3'b000	PD = 512, CPLD = 512*
3'b001	PD = 768, CPLD = 256*
3'b010	PD = 256, CPLD = 768*
3'b011	PD = 922, CPLD = 102*
3'b100	PD = 102, CPLD = 922*
3'b101	PD = 512, CPLD = 512*
3'b110	PD = 768, CPLD = 102*
3'b111	PD = 102, CPLD = 768*
	*actual credit, but 0 (infinite) is advertised during FC Init

Note that credit advertised as 0 during FC Initialization does not need to be updated (through UpdateFC DLLP). If updated, update value should always be 0.

[31:12] RSVD - Reserved

25.3.2.43 PeDRF_CACH: Credit Allocation - Completion Header Register

PCIe Offset: **0x0614**

Table 25.207: PeDRF_CACH

Field	Type (LCB)	Default	Field Name	Field Description
[07:00]	RO	8'h00	PeDRF_CACH	Credit Allocation - Completion Header
[31:08]	RO	24'h00	RSVD	Reserved

[07:00] PeDRF_CACH - Credit Allocation Completion Header

Header credit allocation for Completion TLPs can be read through this field.

Completion header credits can be set as infinite or finite depending on the strap signal FC. Refer to Table 25.207.

Table 25.208: Completion Header Credit Partition through FC

Value	Priority
3'b000	PH = 32, CPLH = 32*
3'b001	PH = 32, CPLH = 32*
3'b010	PH = 32, CPLH = 32*
3'b011	PH = 32, CPLH = 32*
3'b100	PH = 32, CPLH = 32*
3'b101	PH = 32, CPLH = 32
3'b110	PH = 48, CPLH = 16
3'b111	PH = 16, CPLH = 48
	*actual is 32, but 0 (infinite) is advertised during FC Init

When Completion Header credit is set as infinite, Non-Posted requests (those requiring Completions) are not issued unless the device has enough credits to receive the Completions. This ensures that when Completer sends the completion, PeCORE has enough header credit to receive it.

[31:08] RSVD - Reserved

25.3.2.44 PeDRF_RXMSIBAR: MSI Base Address Register (RC)

PCIe Offset: **0x0700**

Table 25.209: PeDRF_MSIBAR

Field	Type (LCB)	Default	Field Name	Field Description
[00]	RW	1'b0	PeDRF_RXMSIBAREN	MSI Base Address Register Enable
[07:01]	RO	7'h0	RSVD	Reserved
[31:08]	RW	24'h0	PeDRF_RXMSIBAR	MSI Base Address Register

[00] PeDRF_RXMSIBAREN - MSI Base Address Register Enable

If this bit is set, MSI-detection is enabled. If this bit, however, is cleared, even if the address of the received Memory Write request falls within the range of MSI BAR, no MSI detection will be reported.

[07:01] RSVD - Reserved

[31:08] PeDRF_RXMSIBAR - MSI Base Address Register

The address range covered by this base address is used as the system-specified address assigned to devices requesting MSI address. Since reception and handling of MSI is done only by the Root Complex, this register is existent only when PeCORE is in Root Complex mode. Any Memory Write request (MSI is a Memory Write request) received that falls within the range of this register, is interpreted as a MSI. Bit 5 of PeDRF_CINT is set when a MSI is detected. This register can point to any valid AWB address.

25.3.2.45 PeDRF_RXMSIMSG: MSI Message Register (RC)

PCIe Offset: **0x0704**

Table 25.210: PeDRF_RXMSIMSG

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RW1C	32'd0	PeDRF_RXMSIMSG	MSI Message Register

[31:00] PeDRF_RXMSIMSG - MSI Message Register

EDC as Root Complex can grant upto 32 MSI messages to requesting devices. For every MSI received, EDC decodes the lower 5 bits of the Message Data. Lower 5 bits with a value of 0 sets bit 0 of this register, a value of 1 sets bit 1, and so on. When any of the bit of this register is set, bit 5 of PeDRF_CINT is set, and an interrupt is sent to the local FW.

25.3.2.46 PeDRF_MSICFG0 MSI Config Register 0 (EP)

PCIe Offset: **0x0708**

Table 25.211: PeDRF_MSICFG0 MSI Config Register 0

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR0CFG	HATR[0] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR1CFG	HATR[1] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR2CFG	HATR[2] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR3CFG	HATR[3] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR0CFG - HATR[0] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 0.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR1CFG - HATR[1] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 1.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR2CFG - HATR[2] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 2.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR3CFG - HATR[3] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 3.

[31:29] RSVD - Reserved

25.3.2.47 PeDRF_MSICFG1 MSI Config Register 1 (EP)

PCIe Offset: **0x070c**

Table 25.212: PeDRF_MSICFG1 MSI Config Register 1

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR4CFG	HATR[4] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR5CFG	HATR[5] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR6CFG	HATR[6] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR7CFG	HATR[7] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR4CFG - HATR[4] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 4.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR5CFG - HATR[5] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 5.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR6CFG - HATR[6] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 6.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR7CFG - HATR[7] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 7.

[31:29] RSVD - Reserved

25.3.2.48 PeDRF_MSICFG2 MSI Config Register 2 (EP)

PCIe Offset: **0x0710**

Table 25.213: PeDRF_MSICFG2 MSI Config Register 2

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR8CFG	HATR[8] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR9CFG	HATR[9] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR10CFG	HATR[10] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR11CFG	HATR[11] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR8CFG - HATR[8] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 8.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR9CFG - HATR[9] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 9.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR10CFG - HATR[10] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 10.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR11CFG - HATR[11] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 11.

[31:29] RSVD - Reserved

25.3.2.49 PeDRF_MSICFG3 MSI Config Register 3 (EP)

PCIe Offset: **0x0714**

Table 25.214: PeDRF_MSICFG3 MSI Config Register 3

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR12CFG	HATR[12] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR13CFG	HATR[13] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR14CFG	HATR[14] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR15CFG	HATR[15] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR12CFG - HATR[12] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 12.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR13CFG - HATR[13] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 13.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR14CFG - HATR[14] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 14.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR15CFG - HATR[15] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 15.

[31:29] RSVD - Reserved

25.3.2.50 PeDRF_MSICFG4 MSI Config Register 4 (EP)

PCIe Offset: **0x0718**

Table 25.215: PeDRF_MSICFG4 MSI Config Register 4

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR16CFG	HATR[16] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR17CFG	HATR[17] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR18CFG	HATR[18] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR19CFG	HATR[19] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR16CFG - HATR[16] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 16.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR17CFG - HATR[17] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 17.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR18CFG - HATR[18] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 18.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR19CFG - HATR[19] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 19.

[31:29] RSVD - Reserved

25.3.2.51 PeDRF_MSICFG5 MSI Config Register 5 (EP)

PCIe Offset: **0x071c**

Table 25.216: PeDRF_MSICFG5 MSI Config Register 5

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR20CFG	HATR[20] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR21CFG	HATR[21] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR22CFG	HATR[22] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR23CFG	HATR[23] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR20CFG - HATR[20] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 20.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR21CFG - HATR[21] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 21.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR22CFG - HATR[22] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 22.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR23CFG - HATR[23] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 23.

[31:29] RSVD - Reserved

25.3.2.52 PeDRF_MSICFG6 MSI Config Register 6 (EP)

PCIe Offset: **0x0720**

Table 25.217: PeDRF_MSICFG6 MSI Config Register 6

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR24CFG	HATR[24] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR25CFG	HATR[25] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR26CFG	HATR[26] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR27CFG	HATR[27] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR24CFG - HATR[24] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 24.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR25CFG - HATR[25] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 25.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR26CFG - HATR[26] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 26.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR27CFG - HATR[27] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 27.

[31:29] RSVD - Reserved

25.3.2.53 PeDRF_MSICFG7 MSI Config Register 7 (EP)

PCIe Offset: **0x0724**

Table 25.218: PeDRF_MSICFG7 MSI Config Register 7

Field	Type (LCB)	Default	Field Name	Field Description
[04:00]	RW	5'd0	PeDRF_HATR28CFG	HATR[28] MSI Msg Number
[07:05]	RO	3'd0	RSVD	Reserved
[12:08]	RW	5'd0	PeDRF_HATR29CFG	HATR[29] MSI Msg Number
[15:13]	RO	3'd0	RSVD	Reserved
[20:16]	RW	5'd0	PeDRF_HATR30CFG	HATR[30] MSI Msg Number
[23:21]	RO	3'd0	RSVD	Reserved
[28:24]	RW	5'd0	PeDRF_HATR31CFG	HATR[31] MSI Msg Number
[31:29]	RO	3'd0	RSVD	Reserved

[04:00] PeDRF_HATR28CFG - HATR[28] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 28.

[07:05] RSVD - Reserved

[12:08] PeDRF_HATR29CFG - HATR[29] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 29.

[15:13] RSVD - Reserved

[20:16] PeDRF_HATR30CFG - HATR[30] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 30.

[23:21] RSVD - Reserved

[28:24] PeDRF_HATR31CFG - HATR[31] MSI Config Register

FW can use this to assign a MSI message number to Host Attention Register bit 31.

[31:29] RSVD - Reserved

25.3.2.54 PeDRF_HAACLR Register (EP)

PCIe Offset: **0x0728**

Table 25.219: PeDRF_HAACLR

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RW	32'h0	PeDRF_HAACLR	Host Attention AutoClear

[31:00] PeDRF_HAACLR - Host Attention AutoClear

If set, this register enables auto-clearing any bit in the Host Attention Register (PeDRF_HATR). A bit of the Host Attention Register is set when its corresponding MSI is already sent to the Host. That way, the Host does not need to access this register to check and clear the source of the interrupt.

25.3.2.55 PeDRF_PMDATA0~7 Register

PCIe Offset: **0x0780** ~ **0x079C**

Table 25.220: PeDRF_PMDATA

Field	Type (LCB)	Default	Field Name	Field Description
[07:00]	RW	8'h0	PeDRF_Data	
[09:08]	RW	2'h0	PeDRF_DataSc	
[31:10]	RW	22'h0	RSVD	Reserved

[07:00] PeDRF_DATA - Power Management Data

[09:08] PeDRF_DATASC -

[31:10] Reserved

25.3.2.56 PeDRF_PBDATA0~5 Register

PCIe Offset: **0x07A0** ~ **0x07B4**

Table 25.221: PeDRF_PBDATA

Field	Type (LCB)	Default	Field Name	Field Description
[20:00]	RW	21'h0	PeDRF_Data	
[31:10]	RW	11'h0	RSVD	Reserved

[20:00] PeDRF_DATA - Power Budgeting Data

[31:10] Reserved

25.3.2.57 PeDRF_LTSR: Link Training Status Register

PCIe Offset: **0x8800**

Table 25.222: PeDRF_LTSR

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RO	RO	4'h0	PeDRF_LTS	Link Training Status
[7:4]	RO	RO	4'h0	PeDRF_PLTS	Previous Link Training Status
[8]	RO	RO	1'b0	PeDRF_LNKUP	LinkUp Status
[16:9]	RO	RO	8'h0	PeDRF_LNKNEG	Link Number Negotiated
[31:17]	RO	RO	15'h0	RSVD	Reserved

[03:00] PeDRF_LTSR_LTS - Current Link Training Status

[07:04] PeDRF_LTSR_PLTS - Previous Link Training Status

Table 25.223: Link Training Status

Value	Link State
4'b0000	Detect State
4'b0001	Polling State
4'b0010	Configuration State
4'b0101	L1 State
4'b0110	L2 State
4'b0111	Recovery State
4'b1000	Loopback State
4'b1001	Hot Reset
4'b1010	Disable State
4'b1100	Tx - L0 State, Rx - L0 State
4'b1101	Tx - L0 State, Rx - L0s State
4'b1110	Tx - L0s State, Rx - L0 State
4'b1111	Tx - L0s State, Rx - L0s State

[08] PeDRF_LTSR_LNKUP - LinkUp Status

This bit indicates the status of the LinkUp signal.

[16:09] PeDRF_LTSR_LNKNEG - Link Number Negotiated

This field indicates the negotiated Link Number.

[31:09] RSVD - Reserved

25.3.2.58 PeDRF_LTCTL Link Training Controller Register

PCIe Offset: **0x0804**

Table 25.224: PeDRF_LTCTL

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[00]	RW	RO	1'b0	PeDRF_RECGoCFG	Recovery to ConfigurationState
[01]	RW	RO	1'b0	PeDRF_POLGoDET	Polling to Detect State
[02]	RW	RO	1'b0	PeDRF_DISGoDET	Disable to Detect State
[03]	RW	RO	1'b0	PeDRF_CFGGoDIS	Configuration to DisableState
[04]	RW	RO	1'b0	PeDRF_RECGoDIS	Recovery to Disable State
[05]	RW	RO	1'b0	PeDRF_RECGoHOT	Recovery to Hot Reset State
[06]	RW	RO	1'b0	PeDRF_CFGGoLBK	Configuration to Loopback
[07]	RW	RO	1'b0	PeDRF_RECGoLBK	Recovery to Loopback
[08]	RW	RO	1'b0	PeDRF_LXGoREC	LX to Recovery
[09]	RW	RO	1'b0	PeDRF_LBKExit	Loopback Exit Enable
[10]	RW	RO	1'b1	PeDRF_LNKL0INTE	L0 State Interrupt Enable
[11]	RW	RO	1'b0	PeDRF_LNKSCINTE	Link State Change Interrupt Enable
[12]	RW	RO	1'b1	PeDRF_LNKL0INTM	Link L0 StateInterrupt Mask
[13]	RW	RO	1'b0	PeDRF_LNKSCINTM	Link State Change Interrupt Mask
[14]	RW	RO	1'b0	PeDRF_LNKDETINTE	Link Detect State Interrupt Enable
[15]	RW	RO	1'b0	PeDRF_LNKDETINTM	Link Detect State Interrupt Mask
[16]	RW	RO	1'b0	PeDRF_CMPRECBITE	Compliance Receive Bit Enable
[17]	RW	RO	1'b0	PeDRF_REC2DETNOFC	Recovery to Detect No FC Init
[31:18]	RO	RO	14'h0	RSVD	Reserved

[00] RECGoCFG- Recovery to Configuration State

This bit, when asserted during Recovery State, will force PeCORE to transition from Recovery State to Configuration State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC to ensure the transition L0 -> REC-> CFG will take place.

[01] POLGoDET - Polling to Detect State

This bit, when asserted during Polling.Compliance state, will force PeCORE to transition from Polling State to Detect State.

[02] DISGoDET - Disable to Detect State

This bit, when asserted during Disable state, will force PeCORE to transition from Disable State to Detect State.

[03] CFGGoDIS - Configuration to Disable State

This bit, when asserted during Configuration.LinkwidthStart (RC), will force PeCORE to transition from Configuration State to Disable State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC and

RECGoCFG to ensure the transition L0 -> REC-> CFG -> DIS will take place. This bit is reserved when the device is in Endpoint Mode.

[04] RECGoDIS - Recovery to Disable State

This bit, when asserted during Recovery.Idle state (RC), will force PeCORE to transition from Recovery State to Disable State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC to ensure the transition L0 -> REC-> DIS will take place. This bit is reserved when the device is in Endpoint Mode.

[05] RECGoHOT - Recovery to Hot Reset State

This bit, when asserted during Recovery.Idle State(RC), will force PeCORE to transition from Recovery State to Hot Reset State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC to ensure the transition L0 -> REC-> HOT will take place. This bit is reserved when the device is in Endpoint Mode.

[06] CFGGoLBK - Configuration to Loopback State

This bit, when asserted during Configuration.LinkwidthStart, will force PeCORE to transition to Loopback State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC and RECGoCFG to ensure the transition L0 -> REC-> CFG -> LBK will take place.

[07] RECGoLBK - Recovery to Loopback State

This bit, when asserted during Recovery.Idle State, will force PeCORE to transition to Loopback State. FW must assert this bit in L0 along with Retrain Link Bit or L0GoREC to ensure the transition L0 -> REC-> LBK will take place.

[08] LXGoREC - LX to Recovery State

This bit, when asserted during L0/L1 State and the device is an Endpoint, will force PeCORE to transition to Recovery State. This bit will also signify that the Endpoint is autonomously changing link bandwidth. This bit would be auto-deasserted once link exits REC state.

[09] LBKExit - Loopback Exit Enable

This bit, when asserted during Loopback Active state will force PeCORE to exit loopback mode.

[10] LNKL0INTE- L0 State Interrupt Enable

This bit, when asserted enables generation of an interrupt during L0 state transition.

[11] **LNKSCINTE - Link State Change Interrupt Enable**

This bit, when asserted enables generation of an interrupt every change of state during LTSSM.

[12] **LNKL0INTM - Link L0 State Interrupt Status Mask**

If set, this bit masks the Link L0 State Interrupt.

[13] **LNKSCINTM - Link State Change Interrupt Status Mask**

If set, this bit masks the Link State Change Interrupt.

[14] **LNKDETINTE- Detect State Interrupt Enable**

This bit, when asserted enables generation of an interrupt during Detect state transition.

[15] **LNKDETINTM - Detect State Interrupt Status Mask**

If set, this bit masks the Link Detect State Interrupt.

[16] **CMPRECBITE - Compliance Receive Bit Enable**

This bit, when asserted, will set Compliance Receive Bit (Symbol 5 : Bit 4) of the transmitted TS1 Ordered Set to 1'b1. This bit is usually used during Polling.Active to order the other device to enter Polling.Compliance.

[17] **REC2DETNOFC - Recovery to Detect Enable: Unsuccessfull FC INIT**

This bit, when asserted, will enable Recovery.Idle to Detect transition when Flow Control Initialization failed during L0. Directing the link to Detect state will somehow repeat the link training up to L0 state thus performing Flow Control Initialization again.

[31:18] **RSVD - Reserved**

25.3.2.59 PeDRF_LASCC: LTSSM Auxiliary Status, Control, and Capability Register

PCIe Offset: **0x0808**

Table 25.225: PeDRF_LASCC

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW	RO	8'd64	PeDRF_NFTS	Number of FTS
[8]	RW	RO	1'b0	PeDRF_DSE	Disable Scramble Enable
[9]	RW	RO	1'b1	PeDRF_LUCC	Link Upconfiguration Capability
[10]	RW	RO	1'b0	PeDRF_UCLWI	Upconfiguration Link Width Initiator
[12:11]	RW	RO	2'b00	PeDRF_UCLW	Upconfigure Link Width
[13]	RW	RO	1'b0	PeDRF_EDDEC	External Device De-emphasis Control
[15:14]	RW	RO	2'b10	PeDRF_LNKIDLTML0S	Link Idle Time Before L0s
[18:16]	RW	RO	3'b010	PeDRF_LNKIDLTML1	Link Idle Time Before L1 ASPM
[26:19]	RO	RO	8'h8	peDRF_EDNFTS	External Device Number of FTS
[30:27]	RO	RO	4'h0	PeDRF_EDSLS	External Device Supported Link Speed
[31]	RO	RO	1'b0	PeDRF_EDDEL	External Device De-emphasis Level

[07:00] NFTS- Number of Fast Training Sequence

This field contains the number of Fast Training Sequences required to achieved Bit Lock and Symbol Lock during Link training and Link initialization.

[08] DSE- Disable Scramble Enable

This bit, when set to 1, is used to disable scrambling of transmitted data. During TS1/TS2 transmission, the Disable Scrambling bit will be asserted.

[09] LUCC - Link Upconfigure Capability

This bit indicates that the device is capable of changing link width without going back to Detect State. This means that the link is capable of either downsizing or upconfiguration of the link width after the initial link width negotiation. FW must see to it that this bit was set before the last entry to Configuration State before it can initiate any link width change.

[10] UCLWI - Upconfigure Link Width Iniator

This bit, when set to 1, is used to indicate that PeCORE was the initiator of the link width change. This bit will be used to direct PeCORE from Recovery to Configuration during link width change.

[12:11] UCLW - Upconfigure Link Width

This field indicates the target link width during downsizing/upconfiguration.

Table 25.226: Target Link Width

Value	Target Link Width
2'b00	x1
2'b01	x2
2'b10	x4
2'b11	x8

[13] EDDEC - External Device De-emphasis Control

This bit, when set to 1 and the device is in Root Complex Mode, indicates that the device will use the de-emphasis level advertised by the Endpoint during Recovery state. When the device is in Endpoint Mode, this bit indicates the de-emphasis level preference of the device. This bit will dictate the advertised value of Selectable De-emphasis Bit in TS1 and TS2 Ordered Sets.

[15:14] LNKIDLTML0S - Link Idle Time Before L0s

This field indicates the amount of link idle time before LTSSM decides to transition the transmit side of the link from L0 to L0s.

Table 25.227: Idle Time

Value	Idle Time
2'b00	0.5us
2'b01	1us
2'b10	2us
2'b11	4us

[18:16] LNKIDLTML1 - Link Idle Time Before L1 ASPM

When L0s entry is disabled, this field indicates the amount of link idle time before LTSSM decides to request for link transition to L1 ASPM. When L0s entry is enabled, this field indicates the amount of time for both directions to be in L0s before LTSSM decides to request for link transition to L1 ASPM.

In case the other device doesn't support L0s, the second scenario wouldn't happen. Hence, it is suggested that L0s functionality to be disabled. Link would then enter L1 via the first scenario.

Table 25.228: Idle Time

Value	Idle Time
3'b000	0.5us
3'b001	1us
3'b010	2us
3'b011	4us
3'b100	6us

Value	Idle Time
3'b101	8us
3'b110	12us
3'b111	16us

[26:19] EDNFTS - External Device Number of FTS

This field indicates the NFTS value advertised, through TS1/TS2, by the device at the opposite side of the link. This is determined during the Configuration and Recovery state of the LTSSM.

[30:27] EDSLS - External Device Supported Link Speed

This field indicates the supported link speed of the device at the opposite side of the link. This is determined during the Configuration state of the LTSSM.

Table 25.229: Ext Device Supported Link Speed

Value	Speed
4'b0001	2.5 GT/s
4'b0010	5.0 GT/s
Others	Reserved

[31] EDDEL - External Device De-emphasis Level

This field indicates the de-emphasis level of the device at the opposite side of the link. This is determined during the Recovery state of the LTSSM.

Table 25.230: Ext Device De-emphasis Level

Value	De-emphasis Level
1'b0	-6.0 dB
1'b1	-3.5 dB

25.3.2.60 PeDRF_SPMC: System Power Management Control Register

PCIe Offset: **0x080C**

Table 25.231: PeDRF_SPMC

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'b0	PeDRF_HPO	PCIe Hierarchy Power Off Control
[1]	RW	RO	1'b0	PeDRF_PRTLCLKSTPEN	Partial Clock Stopping Enable
[15:2]	RO	RO	14'h0	RSVD	Reserved
[23:16]	RW	RO	8'h7F	PeDRF_INACTCTR	PMMaster Inactivity Counter
[31:24]	RO	RO	8'h0	RSVD	Reserved

[00] HPO - PCIe Hierarchy Power Off Control

This bit, when set to '1' by firmware, would initiate PME_Turn_Off TLP message transmission. The PME_Turn_Off TLP message would propagate to all devices within the PCIe hierarchy. Upon receiving PME_TO_ACK, this bit would be reset to '0' by PeCORE control logic. Note that this is applicable only when device is in Root Complex mode.

[01] PRTLCLKSTPEN - Partial Clock Stopping Enable

This bit, when set to '1' by firmware, would also enable clock stopping of all stoppable clocks within CPMU3 except Cr_PeCOREPIPEClk_S during L0/L0s. Note that, by default, automatic clock stopping via hardware is enabled only during L1/L2/L3 states.

[15:2] RSVD - Reserved

[23:16] INACTCTR - PMMaster Inactivity Counter

This field sets the number of clock cycles for CPMU3 Active signal to be deasserted before PMMaster decides that PMU3 is deemed inactive. When this timeout value is reached, PMMaster would continue with its clock stopping process.

[31:24] RSVD - Reserved

25.3.2.61 PeDRF_SRAMTSTCFG1 - SRAM Test Configuration Register

PCIe Offset: **0x0900**

Table 25.232: PeDRF_SRAMTSTCFG1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'b0	TST_HSRAM0_EN	Header Buffer 0 test mode enable
[1]	RW	RO	1'b0	TST_HSRAM1_EN	Header Buffer 1 test mode enable
[2]	RW	RO	1'b0	TST_HSRAM2_EN	Header Buffer 2 test mode enable
[3]	RW	RO	1'b0	TST_HSRAM3_EN	Header Buffer 3 test mode enable
[4]	RW	RO	1'b0	TST_RSRAM0_EN	Retry Buffer 0 test mode enable
[5]	RW	RO	1'b0	TST_RSRAM1_EN	Retry Buffer 1 test mode enable
[6]	RW	RO	1'b0	TST_RSRAM2_EN	Retry Buffer 2 test mode enable
[7]	RW	RO	1'b0	TST_RSRAM3_EN	Retry Buffer 3 test mode enable
[15:8]	RO	RO	8'd0	RSVD	Reserved
[16]	RW	RO	1'b0	TST_IOB0_EN	IO Buffer 0 test mode enable
[17]	RW	RO	1'b0	TST_IOB1_EN	IO Buffer 1 test mode enable
[18]	RW	RO	1'b0	TST_IOB2_EN	IO Buffer 2 test mode enable
[19]	RW	RO	1'b0	TST_IOB3_EN	IO Buffer 3 test mode enable
[31:20]	RO	RO	12'd0	RSVD	Reserved

[00] TST_HSRAM0_EN - Header Buffer 0 test mode enable

[01] TST_HSRAM1_EN - Header Buffer 1 test mode enable

[02] TST_HSRAM2_EN - Header Buffer 2 test mode enable

[03] TST_HSRAM3_EN - Header Buffer 3 test mode enable

[04] TST_RSRAM0_EN - Retry Buffer 0 test mode enable

[05] TST_RSRAM1_EN - Retry Buffer 1 test mode enable

[06] TST_RSRAM2_EN - Retry Buffer 2 test mode enable

[07] TST_RSRAM3_EN - Retry Buffer 3 test mode enable

Each bit field enables test mode for the corresponding Header and Retry buffer SRAMs. The address and data input lines of the SRAMs are connected to the control registers.

[15:08] RSVD - Reserved

[16] TST_IOB0_EN - IO Buffer0 test mode enable

[17] TST_IOB1_EN - IO Buffer1 test mode enable

[18] TST_IOB2_EN - IO Buffer2 test mode enable

[19] TST_IOB3_EN - IO Buffer3 test mode enable

Each fit field enables test mode for the corresponding IO buffer SRAMs. The address and data input lines of the SRAMs are connected to the control registers.

[31:20] RSVD - Reserved

25.3.2.62 PeDRF_TSRAM_PE1 - SRAM Test Parity Error Injection (for RX Header and TX Retry Buffers)

PCIe Offset: **0x0904**

Table 25.233: PeDRF_TSRAM_RHBPE

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'b0	TSRAM_PE1_0_0	Buffer 0, byte-0
[1]	RW	RO	1'b0	TSRAM_PE1_0_1	Buffer 0, byte-1
[2]	RW	RO	1'b0	TSRAM_PE1_0_2	Buffer 0, byte-2
[3]	RW	RO	1'b0	TSRAM_PE1_0_3	Buffer 0, byte-3
[4]	RW	RO	1'b0	TSRAM_PE1_1_0	Buffer 1, byte-0
[5]	RW	RO	1'b0	TSRAM_PE1_1_1	Buffer 1, byte-1
[6]	RW	RO	1'b0	TSRAM_PE1_1_2	Buffer 1, byte-2
[7]	RW	RO	1'b0	TSRAM_PE1_1_3	Buffer 1, byte-3
[8]	RW	RO	1'b0	TSRAM_PE1_2_0	Buffer 2, byte-0
[9]	RW	RO	1'b0	TSRAM_PE1_2_1	Buffer 2, byte-1
[10]	RW	RO	1'b0	TSRAM_PE1_2_2	Buffer 2, byte-2
[11]	RW	RO	1'b0	TSRAM_PE1_2_3	Buffer 2, byte-3
[12]	RW	RO	1'b0	TSRAM_PE1_3_0	Buffer 3, byte-0
[13]	RW	RO	1'b0	TSRAM_PE1_3_1	Buffer 3, byte-1
[14]	RW	RO	1'b0	TSRAM_PE1_3_2	Buffer 3, byte-2
[15]	RW	RO	1'b0	TSRAM_PE1_3_3	Buffer 3, byte-3
[16]	RW	RO	1'b0	IOBSRAM_PE_0_0	IOB Buffer, byte-0
[17]	RW	RO	1'b0	IOBSRAM_PE_0_1	IOB Buffer, byte-1
[18]	RW	RO	1'b0	IOBSRAM_PE_0_2	IOB Buffer, byte-2
[19]	RW	RO	1'b0	IOBSRAM_PE_0_3	IOB Buffer, byte-3
[31:20]	RO	RO	12'd0	RSVD	Reserved

[15:00] TSRAM_PE1_x_y - Buffer x, byte-y

Each of the four RX Header buffers (x = 3, 2, 1, 0) is a 64 x 36 SRAM.

Each of the four TX Retry buffers (x = 3, 2, 1, 0) is a 512 x 36 SRAM.

The upper 4 bits of the 36-bit data corresponds to the parity bit for each byte of the lower 32 bits (bytes-y = 3, 2, 1, 0), as:

BUFFER_x[35:0] = {P3, P2, P1, P0, byte-3, byte-2, byte-1, byte-0}

Set the corresponding TSRAM_PE1_x_y bit of this control register to force the wrong parity bit for each byte.

[19:16] IOBSRAM_PE_0_y - Buffer x, byte-y

The IOB buffer is a 128 x 36 SRAM. The upper 4 bits of the 36-bit data corresponds to the parity bit for each byte of the lower 32 bits:

BUFFER_x[35:0] = {P3, P2, P1, P0, byte-3, byte-2, byte-1, byte-0}

Set the corresponding TSRAM_RHx_y bit of this control register to force the wrong parity bit for each byte.

[31:16] RSVD - Reserved

25.3.2.63 PeDRF_HDRBUF_Addr - RX Header Buffer SRAM Test Address

PCIe Offset: **0x0908**

Table 25.234: PeDRF_HDRBUF_Addr

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[5:0]	RW	RO	6'h00	TRXHEAD_ADDR	Header Buffer Test Address
[7:6]	RW	RO	2'h0	TRXHEAD_SEL	Header Buffer Select
[30:8]	RO	RO	23'd0	RSVD	Reserved
[31]	RW	RO	1'b0	TRXHEAD_W1R0	Write (1) or Read (0) Operation

[05:00] TRXHEAD_ADDR[5:0] - Header Buffer Test Address

Address one of 64 memory locations of header buffer SRAM.

[07:06] TRXHEAD_SEL[1:0] - Header Buffer Select

Selects one of the four header buffer SRAMs (00 = Header Buffer 0, 01 = Header Buffer 1, etc.)

[30:08] RSVD - Reserved

[31] TRXHEAD_W1R0 - Write (1) or Read (0) Operation

When 1, data from PeDRF_HSRAM_Data register is stored in addressed SRAM. When 0, data from addressed SRAM is read into PeDRF_HSRAM_Data register. (See section on "SRAM Test Procedures")

25.3.2.64 PeDRF_HDRBUF_Data0 - RX Header Buffer SRAM Test Data W0

PCIe Offset: **0x090C**

Table 25.235: PeDRF_HDRBUF_Data0

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RW	RO	32'd0	TRXHEAD_DATA_W0	Header Buffer Data[31:0]

[31:00] TRXHEAD_DATA_W0[31:0] - Header Buffer Data[31:0]

Lower 32 bits of Header Buffer Data[35:0].

25.3.2.65 PeDRF_HDRBUF_Data1 - RX Header Buffer SRAM Test Data W1

PCIe Offset: **0x0910**

Table 25.236: PeDRF_HDRBUF_Data1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RO	4'd0	TRXHEAD_DATA_W1	Header Buffer Data[35:32]
[31:4]	RO	RO	28'd0	RSVD	Reserved

[03:00] TRXHEAD_DATA_W0[3:0] - Header Buffer Data[35:32]

Upper 4 bits of Header Buffer Data[35:0].

[31:04] RSVD - Reserved

25.3.2.66 PeDRF_RTRYBUF_Addr - TX Retry Buffer SRAM Test Address

PCIe Offset: **0x0914**

Table 25.237: PeDRF_RTRYBUF_Addr

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[8:0]	RW	RO	9'h00	TTXRETRY_ADDR	Retry Buffer Test Address
[10:9]	RW	RO	2'h0	TTXRETRY_SEL	Retry Buffer Select
[30:11]	RO	RO	20'd0	RSVD	Reserved
[31]	RW	RO	1'b0	TTXRETRY_W1R0	Write (1) or Read (0) Operation

[08:00] TTXRETRY_ADDR[8:0] - Retry Buffer Test Address

Address one of 512 memory locations of retry buffer SRAM.

[10:09] TTXRETRY_SEL[1:0] - Retry Buffer Select

Selects one of the four retry buffer SRAMs (00 = Retry Buffer 0, 01 = Retry Buffer 1, etc.)

[30:11] RSVD - Reserved

[31] TTXRETRY_W1R0 - Write (1) or Read (0) Operation

When 1, data from PeDRF_RSRAM_Data register is stored in addressed SRAM. When 0, data from addressed SRAM is read into PeDRF_RSRAM_Data register. (See section on "SRAM Test Procedures")

25.3.2.67 PeDRF_RTRYBUF_Data0 - TX Retry Buffer SRAM Test Data W0

PCIe Offset: **0x0918**

Table 25.238: PeDRF_RTRYBUF_Data0

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RW	RO	32'd0	TTXRETRY_DATA_W0	Retry Buffer Data[31:0]

[31:00] TTXRETRY_DATA_W0[31:0] - Retry Buffer Data[31:0]

Lower 32 bits of Retry Buffer Data[35:0].

25.3.2.68 PeDRF_RTRYBUF_Data1 - TX Retry Buffer SRAM Test Data W1

PCIe Offset: **0x091C**

Table 25.239: PeDRF_RTRYBUF_Data1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RO	4'd0	TTXRETRY_DATA_W1	Retry Buffer Data[35:32]
[31:4]	RO	RO	28'd0	RSVD	Reserved

[03:00] TTXRETRY_DATA_W0[3:0] - Retry Buffer Data[35:32]

Upper 4 bits of Retry Buffer Data[35:0].

[31:4] RSVD - Reserved

25.3.2.69 PeDRF_IOBUF0_Addr - IO Buffer SRAM Test Address

PCIe Offset: **0x0920**

Table 25.240: PeDRF_IOBUF0_Addr

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW	RO	8'h00	TIOBUF_ADDR	IO Buffer Test Address
[9:8]	RW	RO	2'h0	TIOBUF_SEL	IO Buffer Select
[30:10]	RO	RO	20'd0	RSVD	Reserved
[31]	RW	RO	1'b0	TIOBUF_W1R0	Write (1) or Read (0) Operation

[07:00] TIOBUF_ADDR[7:0] - IO Buffer Test Address

Address one of 128 memory locations of IO buffer SRAM.

[09:08] TIOBUF_SEL[1:0] - IO Buffer Select

Selects IO buffer SRAMs (Always set to 00.)

[30:10] RSVD - Reserved

[31] TIOBUF_W1R0 - Write (1) or Read (0) Operation

When 1, data from PeDRF_IOBUF_Data register is stored in addressed SRAM. When 0, data from addressed SRAM is read into PeDRF_IOBUF_Data register. (See section on "SRAM Test Procedures")

25.3.2.70 PeDRF_IOBUF0_Data0 - IO Buffer SRAM Test Data W0

PCIe Offset: **0x0924**

Table 25.241: PeDRF_IOBUF0_Data0

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RW	RO	32'd0	TIOBUF_DATA_W0	IO Buffer Data[31:0]

[31:00] TIOBUF_DATA_W0[31:0] - IO Buffer Data[31:0]

Lower 32 bits of IO Buffer Data[35:0].

25.3.2.71 PeDRF_IOBUF0_Data1 - IO Buffer SRAM Test Data W1

PCIe Offset: **0x0928**

Table 25.242: PeDRF_IOBUF0_Data1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RO	4'd0	TIOBUF_DATA_W1	IO Buffer Data[35:32]
[31:4]	RO	RO	28'd0	RSVD	Reserved

[03:00] TIOBUF_DATA_W0[3:0] - IO Buffer Data[35:32]

Upper 4 bits of IO Buffer Data[35:0].

[31:04] RSVD - Reserved

25.3.2.72 PeDRF_BISTCTL - BIST Control Register

PCIe Offset: **0x092C**

Table 25.243: PeDRF_BISTCTL

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[2:0]	RW	RO	3'd0	PCS_NUM	PCS Number
[3]	RW	RO	1'd0	BIST_EN	BIST Enable
[5:4]	RW	RO	2'd00	BIST_MOD	BIST Mode
[6]	RW	RO	1'd0	BIST_INJERR	BIST Inject Error
[31:7]	RO	RO	25'd0	RSVD	Reserved

[02:00] PCS_NUM - PCS Number

Selects which PCS would be tested. PCS_NUM could be any value from 0 to 7. This field implies that testing of each PCS would be done serially. This is done in order to minimize register count.

[03] BIST_EN - BIST Enable

When set to '1', BIST test is enabled. Otherwise, BIST test is disabled.

[05:04] BIST_MOD - BIST Mode

Determines type of BIST test. Description of PRBS7 is TBD.

Table 25.244: BIST Modes

Value	BIST Mode Type
2'b00	PRBS7
2'b01	User-Defined Pattern
2'b10	User-Defined Pattern with-out lock state
2'b11	PRBS7 with lock state

[06] BIST_INJERR - BIST Inject Error

When set to '1', false pattern would be sent to the PCS. This is used to test BIST error detection circuitry.

[31:07] RSVD - Reserved

25.3.2.73 PeDRF_BISTPTRN1 - BIST Pattern Register 1

PCIe Offset: **0x0930**

Table 25.245: PeDRF_BISTPTRN1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RW	RO	32'd0	BIST_PTRN1	BIST Pattern Part 1

[31:00] BIST_PTRN1 - BIST Pattern Part 1

Lower 32 bits of BIST test pattern that would be sent to the PCS.

25.3.2.74 PeDRF_BISTPTRN2 - BIST Pattern Register 2

PCIe Offset: **0x0934**

Table 25.246: PeDRF_BISTPTRN2

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW	RO	8'd0	BIST_PTRN2	BIST Pattern Part 2
[31:8]	RO	RO	24'd0	RSVD	Reserved

[07:00] BIST_PTRN2 - BIST Pattern Part 2

Upper 8 bits of BIST test pattern that would be sent to the PCS.

[31:08] RSVD - Reserved

25.3.2.75 PeDRF_BISTSTAT - BIST Status Register

PCIe Offset: **0x0938**

Table 25.247: PeDRF_BISTSTAT

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RO	RO	1'd0	BIST_ERR	BIST Error
[1]	RO	RO	1'd0	BIST_RUN	BIST Run
[2]	RO	RO	1'd0	BIST_OK	BIST Ok
[18:3]	RO	RO	16'd0	BIST_ERR_CNT	BIST Error Count
[31:19]	RO	RO	13'd0	RSVD	Reserved

[00] BIST_ERR - BIST Error

If an error occurs during BIST run, this is set to '1'. This bit would be set to its default value when BIST_EN is deasserted.

[01] BIST_RUN - BIST Run

If BIST run is still ongoing, this is set to '1'. This bit would be set to its default value when BIST_EN is deasserted.

[02] BIST_OK - BIST Ok

If BIST test passed, this is set to '1'. This bit would be set to its default value when BIST_EN is deasserted.

[18:03] BIST_ERR_CNT - BIST Error Count

This indicates the number of BIST errors encountered. This field would be set to its default value when BIST_EN is deasserted.

[31:19] RSVD - Reserved

25.3.2.76 PeDRF_PCS0DBG - GUC PCS0 Debug Register

PCIe Offset: **0x0940**

Table 25.248: PeDRF_PCS0DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS0DBG	GUC PCS0 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS0DBG - GUC PCS0 Debug Field

This field contains current GUC PCS0 debug output port value.

[31:24] RSVD - Reserved

25.3.2.77 PeDRF_PCS1DBG - GUC PCS1 Debug Register

PCIe Offset: **0x0944**

Table 25.249: PeDRF_PCS1DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS1DBG	GUC PCS1 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS1DBG - GUC PCS1 Debug Field

This field contains current GUC PCS1 debug output port value.

[31:24] RSVD - Reserved

25.3.2.78 PeDRF_PCS2DBG - GUC PCS2 Debug Register

PCIe Offset: **0x0948**

Table 25.250: PeDRF_PCS2DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS2DBG	GUC PCS2 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS2DBG - GUC PCS2 Debug Field

This field contains current GUC PCS2 debug output port value.

[31:24] RSVD - Reserved

25.3.2.79 PeDRF_PCS3DBG - GUC PCS3 Debug Register

PCIe Offset: **0x094C**

Table 25.251: PeDRF_PCS3DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS3DBG	GUC PCS3 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS3DBG - GUC PCS3 Debug Field

This field contains current GUC PCS3 debug output port value.

[31:24] RSVD - Reserved

25.3.2.80 PeDRF_PCS4DBG - GUC PCS4 Debug Register

PCIe Offset: **0x0950**

Table 25.252: PeDRF_PCS4DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS4DBG	GUC PCS4 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS4DBG - GUC PCS4 Debug Field

This field contains current GUC PCS4 debug output port value.

[31:24] RSVD - Reserved

25.3.2.81 PeDRF_PCS5DBG - GUC PCS5 Debug Register

PCIe Offset: **0x0954**

Table 25.253: PeDRF_PCS5DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS5DBG	GUC PCS5 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS5DBG - GUC PCS5 Debug Field

This field contains current GUC PCS5 debug output port value.

[31:24] RSVD - Reserved

25.3.2.82 PeDRF_PCS6DBG - GUC PCS6 Debug Register

PCIe Offset: **0x0958**

Table 25.254: PeDRF_PCS6DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS6DBG	GUC PCS6 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS6DBG - GUC PCS6 Debug Field

This field contains current GUC PCS6 debug output port value.

[31:24] RSVD - Reserved

25.3.2.83 PeDRF_PCS7DBG - GUC PCS7 Debug Register

PCIe Offset: **0x095C**

Table 25.255: PeDRF_PCS7DBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PCS7DBG	GUC PCS7 Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PCS7DBG - GUC PCS7 Debug Field

This field contains current GUC PCS7 debug output port value.

[31:24] RSVD - Reserved

25.3.2.84 PeDRF_PMADDBG - GUC PMAD Debug Register

PCIe Offset: **0x0960**

Table 25.256: PeDRF_PMADDBG

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[23:0]	RO	RO	24'd0	PMADDBG	GUC PMA Debug Field
[31:24]	RO	RO	8'd0	RSVD	Reserved

[23:00] PMADDBG - GUC PMA Debug Field

This field contains current GUC PMA debug output port value.

[31:24] RSVD - Reserved

25.3.2.85 PeDRF_GUCPMABO - GUC PHY PMA BIST Observation Register

PCIe Offset: **0x893C**

Table 25.257: PeDRF_GUCPMABO

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RO	RO	32'h0	PeDRF_PMASO	PMA Set Observation

[31:00] PeDRF_PMASO - PMA Set Observation

For GUC internal PMA test observation.

25.3.2.86 PeDRF_GUCPMABC1 - GUC PHY PMA BIST Controller 1 Register

PCIe Offset: **0x8940**

Table 25.258: PeDRF_GUCPMABC1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[31:0]	RW	RW	32'h0	PeDRF_PMASC1	PMA Set Control 1

[31:00] PeDRF_PMASC1 - PMA Set Controller 1

For GUC internal PMA test purpose. Bit[31:0] of [61:0]

25.3.2.87 PeDRF_GUCPMABC2 - GUC PHY PMA BIST Controller 2 Register

PCIe Offset: **0x8944**

Table 25.259: PeDRF_GUCPMABC2

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[29:0]	RW	RW	30'h0	PeDRF_PMASC2	PMA Set Control 2
[31:30]	RO	RO	2'h0	RSVD	Reserved

[29:00] PeDRF_PMASC1 - PMA Set Controller 2

For GUC internal PMA test purpose. Bit[61:32] of [61:0]

[31:30] RSVD - Reserved

25.3.2.88 PeDRF_GUCPMARXC - GUC PMA Rx Controller Register

PCIe Offset: **0x8948**

Table 25.260: PeDRF_GUCPMARXC

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[1:0]	RW	RO	2'b11	PeDRF_RxIchgSel	Squelch Ichgh Control
[9:2]	RW	RO	8'h0	PeDRF_RxCmCtlN	Common Mode Control for rxn
[13:10]	RW	RO	4'b0011	PeDRF_RxCmCtlP	Common Mode Control for rxp
[15:14]	RW	RO	2'b01	PeDRF_RxDataBias	Rx Data Bias Control
[18:16]	RW	RO	3'b010	PeDRF_RxPiBiasCtl	PI Bias Control
[21:19]	RW	RO	3'b010	PeDRF_RxPiBiasCtl2	PI Bias Control 2
[29:22]	RW	RO	8'h77	PeDRF_RxFltrCtl	Threshold for Counter
[30]	RW	RO	1'b0	PeDRF_RxCdrSqSel	CDR SQ Control
[31]	RO	RO	1'b0	RSVD	Reserved

[01:00] PeDRF_RxIchgSel - Squelch Ichgh Control

[09:02] PeDRF_RxCmCtlN - Common Mode Control for rxn

[13:10] PeDRF_RxCmCtlP - Common Mode Control for rxp

[15:14] PeDRF_RxDataBias - Rx Data Bias Control

[18:16] PeDRF_RxPiBiasCtl - PI Bias Control

[21:19] PeDRF_RxPiBiasCtl2 - PI Bias Control 2

[29:22] PeDRF_RxFltrCtl - Threshold for Counter

[30] PeDRF_RcCdrSqSel - CDR SQ Control

[31] RSVD - Reserved

25.3.2.89 PeDRF_GUCPMAPLLC - GUC PMA PLL Controller Register

PCIe Offset: **0x894c**

Table 25.261: PeDRF_GUCPMAPLLC

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RW	RO	4'b0010	PeDRF_PllVc0SpSet	VC0 Speed Setting
[8:4]	RW	RO	5'h0C	PeDRF_PllSiPpathSet	Ppath Current Setting
[13:9]	RW	RO	5'h1C	PeDRF_PllSiCpSet	Charge Pump Current Setting
[16:14]	RW	RO	3'b000	PeDRF_PllTpSet	PLL Internal refclk
[17]	RW	RO	1'b0	PeDRF_CalForce	Calibration Reference
[19:18]	RW	RO	2'b11	PeDRF_CalSelNbc	Tx Driver Bias Control
[23:20]	RW	RO	4'b1000	PeDRF_CalVc500	Voltage Reference Fine Tuner
[24]	RW	RO	1'b0	PeDRF_PipeRateCtl	Pipe Rate External Control
[31:25]	RO	RO	7'h0	RSVD	Reserved

[03:00] PeDRF_PllVc0SpSet - VC0 Speed Setting

[08:04] PeDRF_PllSiPpathSet - Ppath Current Setting

[13:09] PeDRF_PllSiCpSet - Charge Pump Current Setting

[16:14] PeDRF_PllTpSet - Pll Internal refclk

[17] PeDRF_CalForce - Calibration Reference

[19:18] PeDRF_CalSelNbc - Tx Driver Bias Control

[23:20] PeDRF_CalVc500 - Voltage Reference Fine Tuner

[24] PeDRF_PipeRateCtl - Pipe Rate External Control

For GUC Testing on PHY's pipe_rate. For this field to take effect, LTSSM must be disabled first through setting PeDRF_LOOPBACK[1] to 1.

[31:25] RSVD - Reserved

25.3.2.90 PeDRF_GUCPMATXC0/7 - GUC PHY PMA Tx Controller Lane 0/7 Register

PCIe Offset: **0x8950 - 0x896C**

Table 25.262: PeDRF_GUCPMATXC0/7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[7:0]	RW	RO	8'hC4	PeDRF_Tx0PreSkw	Lane 0/7 Tx Cross Point Control
[13:8]	RW	RO	6'h24	PeDRF_Tx0PreSpd	Lane 0/7 Tx Speed Control
[19:14]	RW	RO	6'h01	PeDRF_Tx0PreDly	Lane 0/7 Tx Slew Rate Control
[27:20]	RW	RO	8'hB4	PeDRF_Tx0DeEmpCtl	Lane 0/7 Tx De-emphasis Control
[29:28]	RW	RO	2'b10	PeDRF_Tx0RxDetVthSel	Lane 0/7 Rx Detection Threshold
[31:30]	RO	RO	2'h0	RSVD	Reserved

[07:00] PeDRF_Tx0PreSkw - Lane 0/7 Tx Cross Point Control

[13:08] PeDRF_Tx0PreSpd - Lane 0/7 Tx Speed Control

[19:14] PeDRF_Tx0PreDly - Lane 0/7 Tx Slew RateControl

[27:20] PeDRF_Tx0DeEmpCtl - Lane 0/7 Tx De-emphasis Control

[29:28] RSVD - Tx0RxDetVthSel - Lane 0/7 Rx Detection Threshold

[31:30] RSVD - Reserved

25.3.2.91 PeDRF_GUCPMARXC0/7 - GUC PHY PMA Rx Controller Lane 0/7 Register

PCIe Offset: **0x8970 - 0x898C**

Table 25.263: PeDRF_GUCPMARXC0/7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[15:0]	RW	RO	16'h58A8	PeDRF_Rx0DataEqCtl	Lane 0/7 Rx Equalization Control
[16]	RW	RO	1'b0	PeDRF_Rx0SqVthSel	Lane 0/7 Rx Squelch Vth Control
[17]	RW	RO	1'b1	PeDRF_Rx0EnIntFlt	Lane 0/7 Rx Disable Integral Filter
[23:18]	RW	RO	6'h1B	PeDRF_Rx0VrmSel	Lane 0/7 Rx Voltage Regulator
[31:24]	RO	RO	8'h0	RSVD	Reserved

[15:00] PeDRF_Rx0DataEqCtl - Lane 0/7 Rx Equalization Control

[16] PeDRF_Rx0SqVthSel - Lane 0/7 Rx Squelch vth Control

[17] PeDRF_Rx0EnIntFlt - Lane 0/7 Rx Disable Integral Filter

[23:18] PeDRF_Rx0VrmSel - Lane 0/7 Rx Voltage Regulator

[31:24] RSVD - Reserved

25.3.2.92 PeDRF_GUCPMARXDC0/7 - GUC PHY PMA Rx DFE Controller Lane 0/7 Register

PCIe Offset: **0x8990 - 0x89AC**

Table 25.264: PeDRF_GUCPMARXDC0/7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[17:0]	RW	RO	18'h0	PeDRF_Rx0DfeIdac	Lane 0/7 IDAC Control
[29:18]	RW	RO	12'h0	PeDRF_Rx0DfeIdacMain	Lane 0/7 IDAC LSB Magnitude
[30]	RW	RO	1'b0	PeDRF_Rx0DfeEn	Lane 0/7 DFE Enable
[31]	RO	RO	1'b0	RSVD	Reserved

[17:00] PeDRF_Rx0DfeIdac - Lane 0/7 IDAC Control

[29:18] PeDRF_Rx0DfeIdacMain - Lane 0/7 IDAC LSB Magnitude

[30] PeDRF_Rx0DfeEn - Lane 0/7 DFE Enable

[31] RSVD - Reserved

25.3.2.93 PeDRF_PHY_STATUS PCIe PHY Status

PCIe Offset: **0x8000**

Table 25.265: PeDRF_PHY_STATUS

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RO	RO	1'h0	PWR_RST_STAT	PHY power and reset status
[1]	RO	RO	1'h0	PHY_READY	PHY is ready to run
[3:2]	RO	RO	1'h2	PHY_PWR_MODE	PHY powerdown mode
[31:4]	RO	RO	28'd0	RSVD	Reserved

[00] PWR_RST_STAT - PHY power and reset status

Flag asserts when PHY detects stable power supply and the reset to the PHY itself is deasserted.

[01] PHY_READY - PHY ready to run

Flag asserts when PHY PLLs are stable, and PHY output clock of 125MHz to the PCIe core is stable.

[03:02] PHY_PWR_MODE - PHY power mode

Indicates the powerdown mode state of the PHY as defined by based specification:

Table 25.266: PHY Power Modes

PHY_PWR_MODE[1:0]	Power Mode Description
2'b00	P0, normal operation
2'b01	P0s, not transmitting
2'b10	P1, lower power state
2'b11	P2, lowest power state

[31:04] RSVD - Reserved

25.3.2.94 PeDRF_PCIE_AUX - PCIe Auxiliary Signals Control

PCIe Offset: **0x8004**

Table 25.267: PeDRF_PCIE_AUX

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'b1	PeDRF_PERSTNOUT	PCIe Auxiliary Signal PERST#
[1]	RW	RO	1'b0	PeDRF_PERSTNDIR	XbPe_PERST Pin Direction
[2]	RW	RO	1'b0	PeDRF_LNKWAKEUP	Link Wake Up
[31:3]	RO	RO	29'h0	RSVD	Reserved

[00] PeDRF_PERSTNOUT - PCIe PERSTN Output

Set this field to the desired state of PERST# I/O pin (EDC pin XbPe_PERST). PERST# is an active low reset signal to the slot.

[01] PeDRF_PERSTNDIR - PCIe PERSTN Pin Direction

This field sets the direction of the I/O pin XbPe_PERST. XbPe_PERST direction by default is input. Set this field to 1 to set the direction to output.

[02] PeDRF_LNKWAKEUP - Link Wake Up

During Endpoint mode, FW can trigger link wake up by setting this bit to '1'. Doing this would both trigger Beacon transmission and assert WAKE# output pin (EDC pin XbPe_WAKE). Note that WAKE# is an active low signal driving an open-drain pin.

[31:03] RSVD - Reserved

25.3.2.95 PeDRF_LOOPBACK - PCIe Loopback Control

PCIe Offset: **0x8008**

Table 25.268: PeDRF_LOOPBACK

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'b0	PeDRF_LBEN	Self-Loopback Test Enable
[1]	RW	RO	1'b0	PeDRF_LTSSMDIS	LTSSM Disable
[3:2]	RW	RO	2'b00	PeDRF_LBMSTRMOD	Loopback Master Mode
[5:4]	RW	RO	2'b01	PeDRF_LBSLVMOD	Loopback Slave Mode
[31:6]	RO	RO	26'd0	RSVD	Reserved

[00] PeDRF_LBEN - Self Loopback Test Enable

When this field is set to 1, loopback test is enabled. Each of the lane 16-bit TX data lines is loopback to the RX data lines, thereby bypassing the PCIe physical layer.

[01] PeDRF_LTSSMDIS - LTSSM Disable

When this field is set to 1, LTSSM is disabled. This is used for loopback test.

[03:02] PeDRF_LBMSTRMOD - Loopback Master Mode

Selects near end loopback path. Note that this functional enhancement is applicable to GUC's PCIe PIPE. This field can be configured at run-time as long as the firmware will set this field first before directing the link to Loopback Master transition.

Table 25.269: PeDRF_LBMSTRMOD

Value	Control
2'b00	normal mode
2'b01	parallel Tx to Rx loopback from PCS
2'b10	serial Tx to Rx loopback from PMA
2'b11	reserved

[05:04] PeDRF_LBSLVMOD - Loopback Slave Mode

Selects far end loopback path. Note that this functional enhancement is applicable to GUC's PCIe PIPE. This field must only be configured at start up. Due to GUC PHY implementations, this was hardwired to 2'b01.

Table 25.270: PeDRF_LBSLVMOD

Value	Control
2'b00	far end loopback from pcs slave fifo
2'b01	far end loopback from pcs elastic buffer
2'b10	far end loopback from PMA hard macro in front of CDR
2'b11	far end loopback from PMA hard macro behind CDR

[31:06] RSVD - Reserved

25.3.2.96 PeDRF_CINT2 - PeCORE Interrupt Register 2

PCIe Offset: **0x9000**

Table 25.271: PeDRF_CINT2

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RCW1	RO	1'b0	PeDRF_LNKUPREQINT	Link Up Request Interrupt
[1]	RCW1	RO	1'b0	PeDRF_HOTRSTINT	Hot-Reset Interrupt
[2]	RCW1	RO	1'b0	PeDRF_PERSTINT	PERST# Interrupt
[3]	RCW1	RO	1'b0	PeDRF_LNKL0INT	Link L0 State Interrupt
[4]	RCW1	RO	1'b0	PeDRF_LNKSCINT	Link State Change Interrupt
[5]	RCW1	RO	1'b0	PeDRF_D3HD0RSTINT	D3 Hot to D0 Reset Interrupt
[6]	RCW1	RO	1'b0	PeDRF_LNKDETINT	Link Detect State Interrupt
[15:7]	RO	RO	9'h0	RSVD	Reserved
[16]	RW	RO	1'b0	PeDRF_LNKUPREQMSK	Link Up Request Interrupt Mask
[17]	RW	RO	1'b0	PeDRF_HOTRSTMSK	Hot-Reset Interrupt Mask
[18]	RW	RO	1'b1	PeDRF_PERSTMSK	PERST# Interrupt Mask
[20:19]	RO	RO	2'h0	RSVD	Reserved
[21]	RW	RO	1'b0	PeDRF_D3HD0RSTMSK	D3 Hot to D0 Reset Interrupt Mask
[22]	RW	RO	1'b1	PeDRF_HWSYSINTMSK	Hardware System Interrupt Mask
[31:23]	RO	RO	9'h0	RSVD	Reserved

This register lists the summary of all the interrupt sources of interrupt line #60 and #188 of PeCORE. Once interrupt line number #60 or #188 from PeCORE is received, this register is the first interrupt register to read. From here, all other sources of interrupt can be traced.

[00] PeDRF_LNKUPREQINT - Link Up Request Interrupt

During RC, this bit is set to 1 when a beacon or WAKE# is detected. A critical interrupt is generated.

[01] PeDRF_HOTRSTINT - Hot-Reset Interrupt

This bit is set to 1 when a Hot-Reset event is detected in LTSSM. A critical interrupt is generated.

[02] PeDRF_PERSTINT - PERST# Interrupt

This bit is set to 1 when a valid Soft-Reset event is detected from PERST# input pin XbPe_PERST. A critical interrupt is generated. Note that the Soft-Reset event is considered valid if PERST# is asserted for at least 100us.

[03] PeDRF_LNKL0INT - Link L0 State Interrupt

This bit is set for any state transition of LTSSM to L0. A normal interrupt is generated.

[04] **PeDRF_LNKSCINT - Link State Change Interrupt**

This bit is set for any state transition of LTSSM. A normal interrupt is generated.

[05] **PeDRF_D3HD0RSTINT - D3 Hot to D0 Reset Interrupt**

This bit is set to 1 if a Soft-Reset is required upon transition from D3 hot to D0 when No_Soft_Reset (PeDRF_PMNSRES) bit is deasserted. A critical interrupt is generated.

[06] **PeDRF_LNKDETINT - Link Detect State Interrupt**

This bit is set for any state transition of LTSSM back to Detect. A normal interrupt is generated.

[15:07] **RSVD - Reserved**

[16] **PeDRF_LNKUPREQMSK - Link Up Request Interrupt Mask**

Set this to 1 to mask Link Up Request interrupt.

[17] **PeDRF_HOTRSTMSK - Hot-Reset Interrupt Mask**

Set this to 1 to mask Hot-Reset interrupt.

[18] **PeDRF_PERSTMSK - Soft-Reset Interrupt Mask**

Set this to 1 to mask Soft-Reset interrupt.

[20:19] **RSVD - Reserved**

[21] **PeDRF_D3HD0RSTMSK - D3 Hot to D0 Interrupt Mask**

Set this to 1 to mask D3 Hot to D0 Reset interrupt.

[22] **PeDRF_HWSYSINTMSK - Hardware System Interrupt Mask**

Set this to 1 to mask Hardware System interrupt.

[31:23] **RSVD - Reserved**

25.3.2.97 PeDRF_HOTRST - Hot Reset Status and Control

PCIe Offset: **0x9004**

Table 25.272: PeDRF_HOTRST

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[1:0]	RO	RO	2'b0	RSVD	Reserved
[2]	RW	RO	1'b0	PeDRF_HOTRSTRDY	Hot-Reset Ready
[15:3]	RO	RO	13'h0	RSVD	Reserved
[16]	RO	RO	1'b0	PeDRF_HOTRSTST	Hot-Reset Status
[31:17]	RO	RO	15'h0	RSVD	Reserved

[01:00] RSVD - Reserved

[02] PeDRF_HOTRSTRDY - Hot-Reset Ready

After a Hot-Reset is detected, FW determines when the device is ready to reset. Set this field to 1 to indicate that the device is ready to reset. The LTSSM will then acknowledge Hot-Reset by transmitting TS1 with Hot Reset bit asserted and after 100 us, the FW will start reset sequence for the device.

[15:03] RSVD - Reserved

[16] PeDRF_HOTRSTST - Hot-Reset Status

This bit is set to 1 to indicate that the LTSSM is detecting Hot-Reset request (in-band signal TS1) from the other device.

[31:17] RSVD - Reserved

25.3.2.98 PeDRF_PERST - PERST Timeout Control Register

PCIe Offset: **0x907C**

Table 25.273: PeDRF_PERST_TO

Bits	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[03:00]	RW	RO	4'h2	PeDRF_PERST_TO	PERST Timeout
[07:04]	RW	RO	4'h3	PeDRF_PERST_READY	PERST Ready
[31:08]	RO	RO	24'h0	RSVD	Reserved

[03:00] PeDRF_PERST_TO - PERST Timeout

This field determines the timeout value for PERST# to be serviced before Hardware System Interrupt automatically asserts.

Table 25.274: PERST Timeout Value

Value	Timeout Value
4'h0	1ms
4'h1	4ms
4'h2	8ms
4'h3	10ms
4'h4	100ms
4'h5	1s
4'h6	8ms
4'h7	8ms

[07:04] PeDRF_PERST_READY - PERST Ready

This field determines the duration for PERST# to be considered valid/ ready. This timeout is used to ignore invalid PERST# glitches.

Table 25.275: PERST

Value	PERST duration
4'h0	1us
4'h1	10us
4'h2	50us
4'h3	100us
4'h4	1ms
4'h5	10ms
4'h6	100us
4'h7	100us

[31:08] Reserved

25.3.2.99 PeDRF_DBG0 - Debug CSR 0

PCIe Offset: **0x7FFC**

Table 25.276: PeDRF_DBG0

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RO	RO	1'b1	PeDRF_PIPE0TXEI	pipe0_tx_elecidle
[1]	RO	RO	1'b0	PeDRF_PIPE0RXV	pipe0_rx_valid
[2]	RO	RO	1'b0	PeDRF_PIPE0RXEI	pipe0_rx_elecidle
[5:3]	RO	RO	3'h0	PeDRF_PIPE0RXST	pipe0_rx_status
[6]	RO	RO	1'b0	PeDRF_PIPENTXDR	pipeN_tx_detectrx
[7]	RO	RO	1'b0	RSVD	Reserved
[8]	RO	RO	1'b1	PeDRF_PIPE1TXEI	pipe1_tx_elecidle
[9]	RO	RO	1'b0	PeDRF_PIPE1RXV	pipe1_rx_valid
[10]	RO	RO	1'b0	PeDRF_PIPE1RXEI	pipe1_rx_elecidle
[13:11]	RO	RO	3'h0	PeDRF_PIPE1RXST	pipe1_rx_status
[15:14]	RO	RO	2'b10	PeDRF_PIPENPWRDN	pipeN_powerdown
[16]	RO	RO	1'b1	PeDRF_PIPE2TXEI	pipe2_tx_elecidle
[17]	RO	RO	1'b0	PeDRF_PIPE2RXV	pipe2_rx_valid
[18]	RO	RO	1'b0	PeDRF_PIPE2RXEI	pipe2_rx_elecidle
[21:19]	RO	RO	3'h0	PeDRF_PIPE2RXST	pipe2_rx_status
[23:22]	RO	RO	2'b00	RSVD	Reserved
[24]	RO	RO	1'b1	PeDRF_PIPE3TXEI	pipe3_tx_elecidle
[25]	RO	RO	1'b0	PeDRF_PIPE3RXV	pipe3_rx_valid
[26]	RO	RO	1'b0	PeDRF_PIPE3RXEI	pipe3_rx_elecidle
[29:27]	RO	RO	3'h0	PeDRF_PIPE3RXST	pipe3_rx_status
[31:30]	RO	RO	2'b00	RSVD	Reserved

This register lists some of the PIPE signals for debugging purposes.

25.3.2.100 PeDRF_DBG1 - Debug CSR 1

PCIe Offset: **0x7FF8**

Table 25.277: PeDRF_DBG1

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RO	RO	1'b1	PeDRF_PIPE4TXEI	pipe4_tx_elecidle
[1]	RO	RO	1'b0	PeDRF_PIPE4RXV	pipe4_rx_valid
[2]	RO	RO	1'b0	PeDRF_PIPE4RXEI	pipe4_rx_elecidle
[5:3]	RO	RO	3'h0	PeDRF_PIPE4RXST	pipe4_rx_status
[7:6]	RO	RO	2'b00	RSVD	Reserved
[8]	RO	RO	1'b1	PeDRF_PIPE5TXEI	pipe5_tx_elecidle
[9]	RO	RO	1'b0	PeDRF_PIPE5RXV	pipe5_rx_valid
[10]	RO	RO	1'b0	PeDRF_PIPE5RXEI	pipe5_rx_elecidle
[13:11]	RO	RO	3'h0	PeDRF_PIPE5RXST	pipe5_rx_status
[15:14]	RO	RO	2'b00	RSVD	Reserved
[16]	RO	RO	1'b1	PeDRF_PIPE6TXEI	pipe6_tx_elecidle
[17]	RO	RO	1'b0	PeDRF_PIPE6RXV	pipe6_rx_valid
[18]	RO	RO	1'b0	PeDRF_PIPE6RXEI	pipe6_rx_elecidle
[21:19]	RO	RO	3'h0	PeDRF_PIPE6RXST	pipe6_rx_status
[23:22]	RO	RO	2'b00	RSVD	Reserved
[24]	RO	RO	1'b1	PeDRF_PIPE7TXEI	pipe7_tx_elecidle
[25]	RO	RO	1'b0	PeDRF_PIPE7RXV	pipe7_rx_valid
[26]	RO	RO	1'b0	PeDRF_PIPE7RXEI	pipe7_rx_elecidle
[29:27]	RO	RO	3'h0	PeDRF_PIPE7RXST	pipe7_rx_status
[31:30]	RO	RO	2'b00	RSVD	Reserved

This register lists some of the PIPE signals for debugging purposes.

25.3.2.101 PeDRF_DBG2 - Debug CSR 2

PCIe Offset: **0x7FF4**

Table 25.278: PeDRF_DBG2

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[5:0]	RO	RO	6'h0	PeDRF_TXREQSM	Tx Request State Machine
[7:6]	RO	RO	2'h0	RSVD	Reserved
[12:8]	RO	RO	5'h0	PeDRF_TXFCCHKSM	Tx FC Check State Machine
[15:13]	RO	RO	3'h0	RSVD	Reserved
[20:16]	RO	RO	2'b00	PeDRF_TXBUFCTLSM	Tx Buffer Controller State Machine
[23:21]	RO	RO	3'h0	RSVD	Reserved
[24]	RO	RO	1'b0	PeDRF_Tran_TlpSent	Transaction Layer Transmitted TLP
[25]	RO	RO	1'b0	PeDRF_Link_TlpSent	Link Layer Transmitted TLP
[26]	RO	RO	1'b0	PeDRF_Tran_TlpRcvd	Transaction Layer Received TLP
[27]	RO	RO	1'b0	PeDRF_Link_GoodTlpRcvd	Link Layer Received Good TLP
[28]	RO	RO	1'b0	PeDRF_Link_BadTlpRcvd	Link Layer Received Bad TLP
[29]	RO	RO	1'b0	PeDRF_Tran_CplReqSent	Transaction Layer Transmitted Cpl
[31:30]	RO	RO	2'h0	RSVD	Reserved

25.3.2.102 PeDRF_DBG3 - Debug CSR 3

PCIe Offset: **0x7FF0**

Table 25.279: PeDRF_DBG3

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[3:0]	RO	RO	4'h0	PeDRF_RXMWRSM	Rx Memory Write State Machine
[7:4]	RO	RO	4'h0	RSVD	Reserved
[13:8]	RO	RO	6'h0	PeDRF_RXMRDSM	Rx Memory Rd Request State Machine
[15:14]	RO	RO	2'h0	RSVD	Reserved
[17:16]	RO	RO	2'h0	PeDRF_RXCSRSM	Rx CSR (Cfg/IO) State Machine
[23:18]	RO	RO	6'h0	RSVD	Reserved
[27:24]	RO	RO	4'h0	PeDRF_RXCPLSM	Rx Cpl State Machine
[31:28]	RO	RO	4'h0	RSVD	Reserved

25.3.2.103 PeDRF_DBG4 - Debug CSR 4

PCIe Offset: **0x7FEC**

Table 25.280: PeDRF_DBG4

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[2:0]	RO	RO	3'h0	PeDRF_DLCMSM	Data Link Control & Mgmt State Machine
[4:3]	RO	RO	2'h0	RSVD	Reserved
[5]	RO	RO	1'h0	PeDRF_TXINITFCDLDN	Tx InitFC DL Down
[6]	RO	RO	1'h0	PeDRF_TXINITFCDLUP	Tx InitFC DL Up
[7]	RO	RO	1'h0	PeDRF_DLCMDLAC	Data Link Control & Mgmt DL_Active
[8]	RO	RO	1'h0	PeDRF_LNKUP	LTSSM Link Up
[15:9]	RO	RO	7'h0	RSVD	Reserved
[18:16]	RO	RO	2'h0	PeDRF_RXINITFCSM	Rx InitFC State Machine
[21:19]	RO	RO	3'h0	RSVD	Reserved
[22]	RO	RO	1'h0	PeDRF_RXINITFCFI1	Rx InitFC Support FI1
[23]	RO	RO	1'h0	PeDRF_RXINITFCFI2	Rx InitFC Support FI2
[28:24]	RO	RO	5'h0	PeDRF_TXINITFCSM	Tx InitFC State Machine
[31:29]	RO	RO	3'h0	RSVD	Reserved

25.3.2.104 PeDRF_DBG5 - Debug CSR 5

PCIe Offset: **0x7FE8**

Table 25.281: PeDRF_DBG5

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[0]	RW	RO	1'h0	PeDRF_DUPFCTX	
[1]	RW	RO	1'h0	PeDRF_DCRDRETICBM	
[31:29]	RO	RO	30'h0	RSVD	Reserved

25.3.2.105 PeDRF_DBG6 - Debug CSR 6

PCIe Offset: **0x7FE4**

Table 25.282: PeDRF_DBG6

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[17:0]	RO	RO	18'h0	PeDRF_LINKSTATE	LTSSM Link State
[19:18]	RO	RO	2'h0	RSVD	Reserved
[20]	RO	RO	1'b0	PeDRF_Pol24msTO	24 ms timeout during Polling
[21]	RO	RO	1'b0	PeDRF_CfgIdle2msTO	2 ms timeout during Configuration Idle
[23:22]	RO	RO	2'h0	PeDRF_NUMRXDET	Number of Receiver detected
[25:24]	RO	RO	2'h0	PeDRF_CfgLk	Configured Link
[26]	RO	RO	1'b0	PeDRF_LinkUp	Link Up
[27]	RO	RO	1'b0	PeDRF_TXINITFCDLDN	Tx InitFC DL Down
[28]	RO	RO	1'b0	PeDRF_TXINITFCDLUP	Tx InitFC DL Up
[29]	RO	RO	1'b0	PeDRF_DLCMDLAC	Data Link Control & Mgmt DL_Active
[30]	RO	RO	1'b0	PeDRF_RXINITFCFI1	Rx InitFC Support FI1
[30]	RO	RO	1'b0	PeDRF_RXINITFCFI2	Rx InitFC Support FI2

25.3.2.106 PeDRF_DBG7 - Debug CSR 7

PCIe Offset: **0x7FE0**

Table 25.283: PeDRF_DBG7

Field	Type (LCB)	Type (PCIe)	Default	Field Name	Field Description
[17:0]	RO	RO	18'h0	PeDRF_LINKSTATE	LTSSM Link State
[19:18]	RO	RO	2'h0	RSVD	Reserved
[26:20]	RO	RO	7'h0	PeDRF_PhyErr	PHY Error
[27]	RO	RO	1'b0	RSVD	Reserved
[31;28]	RO	RO	4'b0	PeDRF_L0XGo2Rec	L0 to Recovery Transition

25.3.2.107 PeDRF_DBG8 - Debug CSR 8

PCIe Offset: **0x7FDC**

Table 25.284: PeDRF_DBG8

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RO	4'b0	PeDRF_MWRDAT	MWr Data

[31:00] PeDRF_MWRDAT - MWr Data

When PeCORE receives MWr request from the other device, the last word at wordlane 0 of the data payload can be read through this register.

25.3.2.108 PeDRF_DBG9 - Debug CSR 9

PCIe Offset: **0x7FD8**

Table 25.285: PeDRF_DBG8

Field	Type (LCB)	Default	Field Name	Field Description
[31:00]	RO	4'b0	PeDRF_MWRADR	MWr Address

[31:00] PeDRF_MWRDAT - MWr Data

When PeCORE receives MWr request from the other device, the lower 32 bits of the translated MWr address can be read through this register.