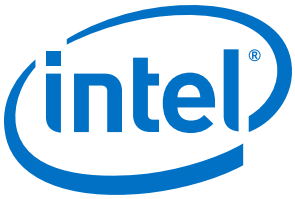


eCPRI Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.4**

IP Version: **1.0.0**

Subscribe
Send Feedback

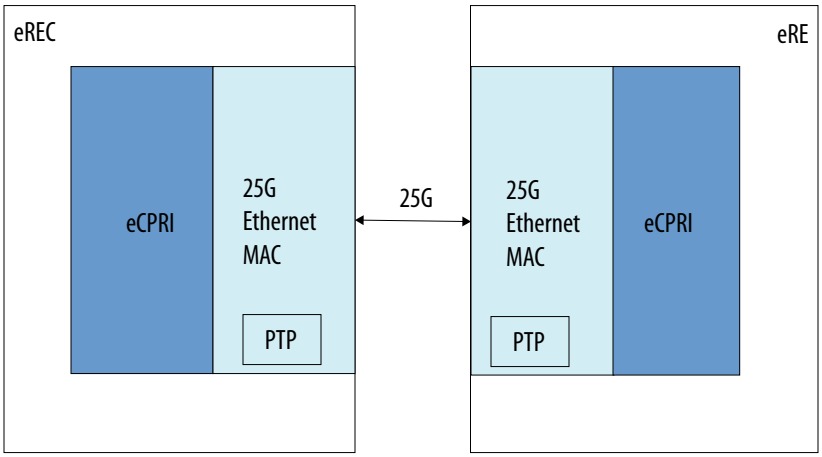
UG-20277 | 2020.04.15
Latest document on the web: **PDF** | **HTML**

Contents

1. Introduction

The Common Public Radio Interface (eCPRI) Intel® FPGA IP core implements the *eCPRI specification version 1.2*. The eCPRI IP is a front-haul interface protocol for radio base station aimed at connecting the eCPRI Radio Equipment Control (eREC) and the eCPRI Radio Equipment (eRE) via front-haul transport network.

Figure 1. Typical eCPRI Application on Intel FPGA Devices



Related Information

eCPRI Specification V1.2

1.1. Supported Features

The eCPRI Intel FPGA IP core offers the following features:

- Compliant with the *eCPRI Specification V1.2 (2018-06-25)* available on the CPRI Industry Initiative (CII) website.
- Supports eCPRI radio equipment controller (eREC) and eCPRI radio equipment (eRE) module configurations.
- Support for Ethernet headers in a variety of formats, including VLAN tag, source/destination MAC address, IPv4, UDP extraction and encapsulation.
- Supports eCPRI one-way delay measurement based on IEEE Standard 1588 Precision Clock Synchronization Protocol (1588 PTP) hardware timestamp. Full hardware support, and required 1588 PTP software stack.
- Supports 25 Gbps Ethernet port.
- Packet classifier responsible to classify eCPRI packet and send packets to eCPRI IP. All other packets are redirected to external port for user processing.

- Programmable packet queue (maximum 16 entries) to hold incoming packets when eCPRI packets transmission in progress.
- Arbitration between eCPRI packet and external incoming Ethernet frames, e.g., Control & Management (C&M) and synchronization packets.
- Offers mapping logic between eCPRI message physical channel ID to VLAN/MAC address CSR.
- Supports single DU and up to eight RU configurations using source/destination MAC address CSR.
- Support all eCPRI message types compliant to eCPRI specification v1.2
- Input/output ports compliant with Avalon-Streaming (Avalon-ST) interface .

Related Information

- CPRI Industry Initiative website
- IEEE website

1.2. Device Family Support

Table 1. Intel FPGA IP Core Device Support Levels

Device Support Level	Definition
Advance	The IP core is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (datapath width, burst depth, I/O standards tradeoffs).
Preliminary	The IP core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
Final	The IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs.

Table 2. eCPRI Intel FPGA IP Core Device Family Support

Shows the level of support offered by the eCPRI Intel FPGA IP for each Intel FPGA device family.

Device Family	Support
Intel Stratix® 10 (H- and E-tile devices only)	Final
Other device families	No support

1.3. Resource Utilization

The resources for the eCPRI Intel FPGA IP core were obtained form the Intel Quartus® Prime Pro Edition software version 19.4 with advance mapping enabled:

eCPRI Line Rate	ALMs	Logic Registers		Memory 20K
		Primary	Secondary	
25 Gbps	8016	17000	2500	86

1.4. Release Information

Table 3. eCPRI Intel FPGA IP Core Release Information

Item	Description
IP Version	1.0.0
Intel Quartus Prime Version	19.4
Release Date	2020.02.10
Ordering Code	IP-eCPRI

1.5. eCPRI Intel FPGA IP Device Speed Grade Support

The eCPRI Intel FPGA IP core supports Intel Stratix 10 H- and E-tile devices with these speed grade properties:

- Transceiver speed grade: –2
- Core speed grade: –2

1.6. Intel FPGA IP Core Verification

To ensure functional correctness of the eCPRI Intel FPGA IP core, Intel performs validation through both simulation and hardware testing. Before releasing a version of the eCPRI Intel FPGA IP core, Intel runs regression tests in the associated version of the Intel Quartus Prime software.

2. Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the eCPRI Intel FPGA IP IP core:

2.1. Installing and Licensing

The eCPRI Intel FPGA IP core is an extended FPGA IP core which is not included with the Intel Quartus Prime release. This section provides a general overview of the Intel extended FPGA IP core installation process to help you quickly get started with any Intel extended FPGA IP core.

The Intel extended FPGA IP cores are available from the Intel Self-Service Licensing Center (SSLC). Refer to Related Information below for the correct link for this IP core.

Figure 2. eCPRI Intel FPGA IP Core Installation Directory Structure

Directory structure after you install the eCPRI IP core.

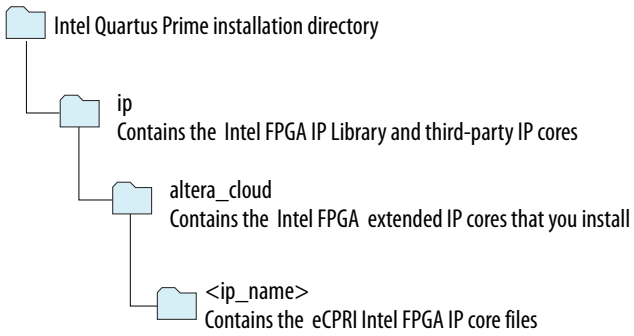


Table 4. Intel FPGA IPCore Installation Locations

Location	Software	Platform
<*drive*>:\intelFPGA_pro\<*version*>\quartus\ip\altera_cloud	Intel Quartus Prime Pro Edition	Windows*
<*home directory*>:/intelFPGA_pro/<*version*>/quartus/ip/altera_cloud	Intel Quartus Prime Pro Edition	Linux*

Related Information

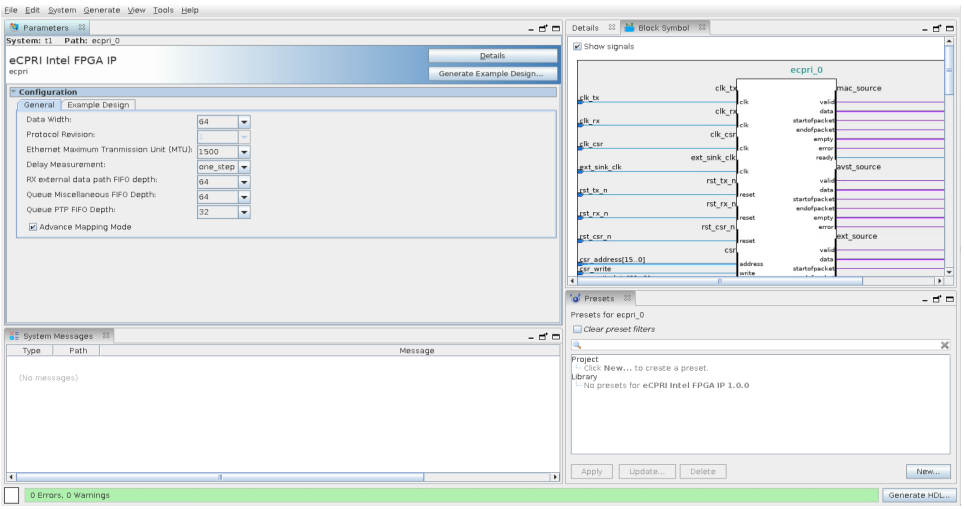
- Intel FPGA website
- Self-Service Licensing Center (SSLC)
  After you purchase the eCPRI Intel FPGA IP core, the IP core is available for download from the SSLC page in your My Intel account. You must create a My Intel account if you do not have one already, and log in to access the SSLC. On the SSLC page, click Run for this IP core. The SSLC provides an installation dialog box to guide your installation of the IP core.

2.2. Specifying the IP Core Parameters and Options

The IP parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Intel Quartus Prime Pro Edition software.

Prerequisite: Once you receive the eCPRI web-core IP, save the web-core installer to the local area. Run the installer with Windows/Linux. When prompt, install to the same location as Intel Quartus Prime folder. The eCPRI Intel FPGA IP now appears in the IP Catalog.

Figure 3. eCPRI IP Parameter Editor



1. If you do not already have an Intel Quartus Prime Pro Edition project in which to integrate your eCPRI IP core, you must create one.
   a. In the Intel Quartus Prime Pro Edition, click **File ➤ New Project Wizard** to create a new Quartus Prime project, or **File ➤ Open Project** to open an existing Quartus Prime project. The wizard prompts you to specify a device.
   b. Specify the device family **Intel Stratix 10 (GX/SX/MX/TX/DX)** and select a production H-tile or E-tile device that meets the speed grade requirements for the IP core.
   c. Click **Finish**.
2. In the IP Catalog, locate and select **eCPRI Intel FPGA IP**. The **New IP Variation** window appears.
3. Specify a top-level name for your new custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>*.ip.
4. Click **OK**. The parameter editor appears.
5. Specify the parameters for your IP core variation. Refer to Parameter Settings on page 13 for information about specific IP core parameters.
6. Optionally, to generate a simulation testbench or compilation and hardware design example, follow the instructions in the *Design Example User Guide*.
7. Click **Generate HDL**. The **Generation** dialog box appears.

8. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
9. Click **Finish**. The parameter editor adds the top-level `.ip` file to the current project automatically. If you are prompted to manually add the `.ip` file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports and set any appropriate per-instance RTL parameters.

Related Information

eCPRI Intel Stratix 10 FPGA Design Example User Guide

2.3. Generated File Structure

The Intel Quartus Prime Pro Edition software generates the following IP core output file structure.

Figure 4. eCPRI IP Core Generated Files

For more information about the file structure of the design example, refer to the *eCPRI Intel Stratix 10 FPGA Design Example User Guide*.

Figure 5. eCPRI IP Core Generated Files

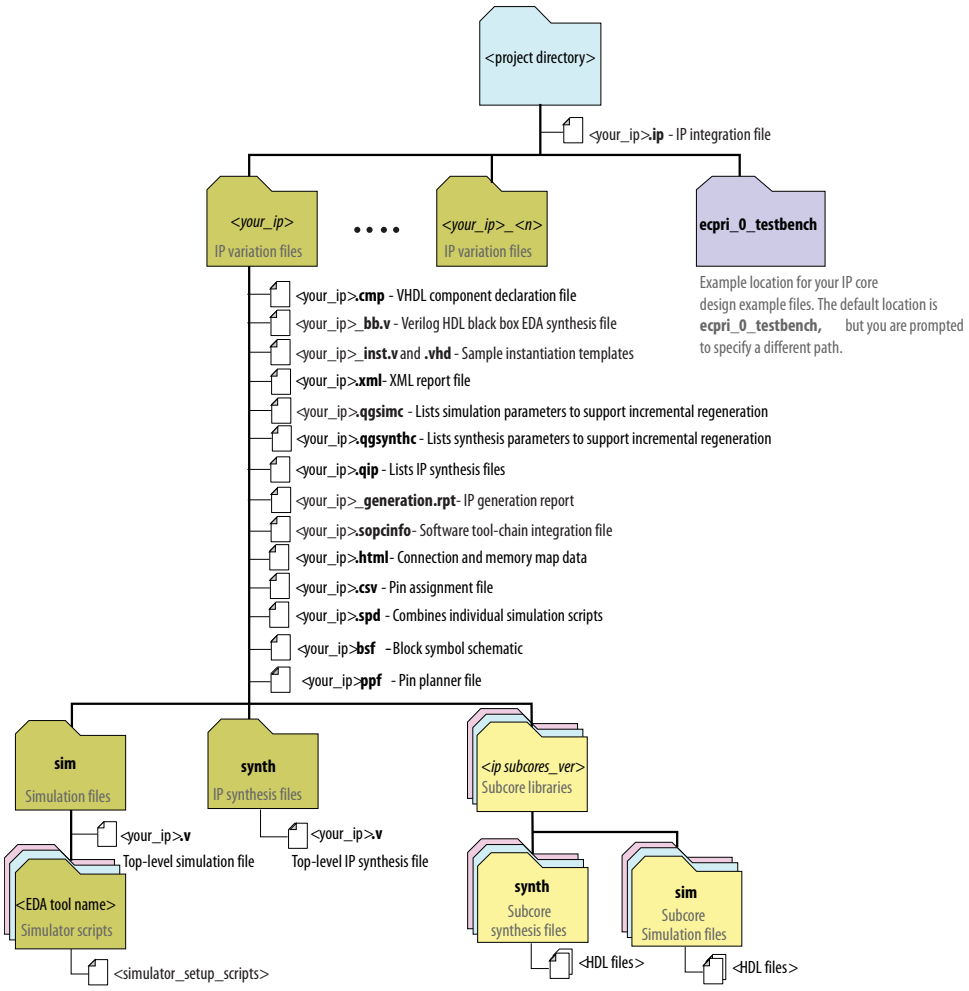


Table 5. eCPRI IP Core Generated Files

File Name	Description
<*your_ip*>.ip	The Platform Designer system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*your_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.
<*your_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*your_ip*>.qgsimc	Lists simulation parameters to support incremental regeneration.
<*your_ip*>.qgsynthc	Lists synthesis parameters to support incremental regeneration.
<*your_ip*>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime software.

continued...

File Name	Description
<*your_ip*>.sopcinfo	Describes the connections and IP component parameterizations in your Platform Designer system. You can parse its contents to get requirements when you develop software drivers for IP components. Downstream tools such as the Nios® II tool chain use this file. The .sopcinfo file and the system.h file generated for the Nios II tool chain include address map information for each slave relative to each master that accesses the slave. Different masters may have a different address map to access a particular slave component.
<*your_ip*>.csv	Contains information about the upgrade status of the IP component.
<*your_ip*>.bsf	A Block Symbol File (.bsf) representation of the IP variation for use in Intel Quartus Prime Block Diagram Files (.bdf).
<*your_ip*>.spd	Required input file for ip-make-simscript to generate simulation scripts for supported simulators. The .spd file contains a list of files generated for simulation, along with information about memories that you can initialize.
<*your_ip*>.ppf	The Pin Planner File (.ppf) stores the port and node assignments for IP components created for use with the Pin Planner
<*your_ip*>_bb.v	You can use the Verilog black-box (_bb.v) file as an empty module declaration for use as a black box.
<*your_ip*>_inst.v or _inst.vhd	HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*your_ip*>.v or <*your_ip*>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim* script msim_setup.tcl to set up and run a simulation.
synopsys/vcs/ synopsys/vcsmx/	Contains a shell script vcs_setup.sh to set up and run a VCS* simulation. Contains a shell script vcsmx_setup.sh and synopsys_ sim.setup file to set up and run a VCS MX* simulation.
cadence/	Contains a shell script ncsim_setup.sh and other setup files to set up and run an NCSIM* simulation.
aldec/	Contains a shell script rivierapro_setup.sh to setup and run an Aldec* simulation.
xcelium/	Contains a shell script xcelium_setup.sh and other setup files to set up and run an Xcelium* simulation.
submodules/	Contains HDL files for the IP core submodules.
<*child IP cores*>/	For each generated child IP core directory, Platform Designer generates synth/ andsim/ sub-directories.

Related Information

eCPRI Intel Stratix 10 FPGA Design Example User Guide

2.4. Simulating the IP Core

You can simulate your eCPRI IP variation using any of the vendor-specific IEEE encrypted functional simulation models which are available in the <instance_name>/sim subdirectory of your project directory.

The eCPRI IP core supports the Synopsys VCS, Synopsys VCS MX, Mentor Graphics Modelsim-SE, Cadence NCSim, Aldec Riviera and Xcelium Parallel simulators. The eCPRI IP core generates a Verilog HDL and VHDL simulation model. The IP core

parameter editor offers you the option of generating a Verilog HDL or VHDL simulation model for the IP core. The IP core design example also supports Verilog HDL/VHDL simulation model or testbench.

For more information about functional simulation models for Intel FPGA IP cores, refer to the *Simulating Intel FPGA Designs chapter in Quartus Prime Pro Edition User Guide: Third-party Simulation*.

Related Information

- Simulating Intel FPGA Designs
- eCPRI Intel Stratix 10 FPGA Design Example User Guide

2.5. Compiling the Full Design and Programming the FPGA

You can use the **Start Compilation** command on the **Processing** menu in the Intel Quartus Prime software to compile your design. After successfully compiling your design, program the targeted Intel device with the Programmer and verify the design in hardware.

3. Parameter Settings

3.1. Parameter Settings

You customize the IP core by specifying parameters in the IP parameter editor.

Table 6. Parameters: Configuration Tab

Parameter	Supported Values	Default Setting	Description
Data Width	64	64	Primary data bus width.
Protocol Revision	1	1	Specifies eCPRI protocol revision used in eCPRI common header. This option is grayed out in the current version of the Intel Quartus Prime software.
Ethernet Maximum Transmission Unit (MTU)	1500	1500	Indicates maximum Ethernet frame size.
Delay Measurement	• off • one_step • two_step	one_step	Indicates support and way of measurement for eCPRI message type 5 delay measurement. When set to **off**, the IP does not include the delay measurement logic.
RX external data path FIFO depth	• 64 • 128 • 256	64	Indicates the depth of the RX external data path FIFO. The actual depth is 2^(value of this parameter).
Queue Miscellaneous FIFO depth	• 32 • 64 • 128 • 256	64	Indicates the depth of the Queue miscellaneous FIFO. The actual depth is 2^(value of this parameter).
Queue PTP FIFO Depth	• 32 • 64 • 128 • 256	32	Indicates the depth of the Queue PTP FIFO. The actual depth is 2^(value of this parameter).
Advance Mapping Mode	• On • Off	On	When you turn on this parameter, it allows the mapping of the destination MAC address and VLAN tag CSE to eCPRI message PC_ID field.

For parameters in the **Example Design** tab, refer to the *eCPRI Intel Stratix 10 FPGA Design Example User Guide*.

Related Information

- eCPRI Intel Stratix 10 FPGA Design Example User Guide
- 25G Ethernet Intel Stratix 10 FPGA IP User Guide
- E-tile Hard IP User Guide

4. Functional Description

The eCPRI Intel FPGA IP core provides the functionality described in the *eCPRI specification version 1.2*.

4.1. Interfaces

The eCPRI Intel FPGA IP IP supports the following interfaces:

- **Clock and Reset Interface**

  The main interface for the clock and reset signals in the eCPRI IP.

- **Configuration Avalon-Memory Mapped Interface**

  This interface provides access to the internal control and status registers of the eCPRI IP. This interface complies with Avalon Memory-Mapped (Avalon-MM) specification as defined in the *Avalon Interface Specifications*.

- **External MAC Source Interface**

  This interface provides datapath from eCPRI IP to 25G Ethernet MAC IP. This interface complies with Avalon-Steaming (Avalon-ST) specification as defined in the *Avalon Interface Specifications*.

- **External MAC Sink Interface**

  This interface provides datapath from 25G Ethernet MAC IP to eCPRI IP. This interface complies with Avalon-Steaming specification as defined in the *Avalon Interface Specifications*.

- **eCPRI IP Source Interface**

  This interface provides datapath from eCPRI IP to client logic. This interface includes a number of sideband signals which align with the Avalon-ST clock. This interface complies with Avalon-Steaming specification as defined in the *Avalon Interface Specifications*.

- **eCPRI IP Sink Interface**

  This interface provides datapath from client logic to eCPRI IP . This interface includes a number of sideband signals which align with the Avalon-ST clock. This interface complies with Avalon-Steaming specification as defined in the *Avalon Interface Specifications*.

- **External ST Source Interface**

  This interface provides datapath from eCPRI IP to client logic. This interface is a primary output interface for PTP and C&M messages. This interface complies with Avalon-Steaming specification as defined in the *Avalon Interface Specifications*. This interface includes a number of sideband signals which align with the Avalon-ST clock.

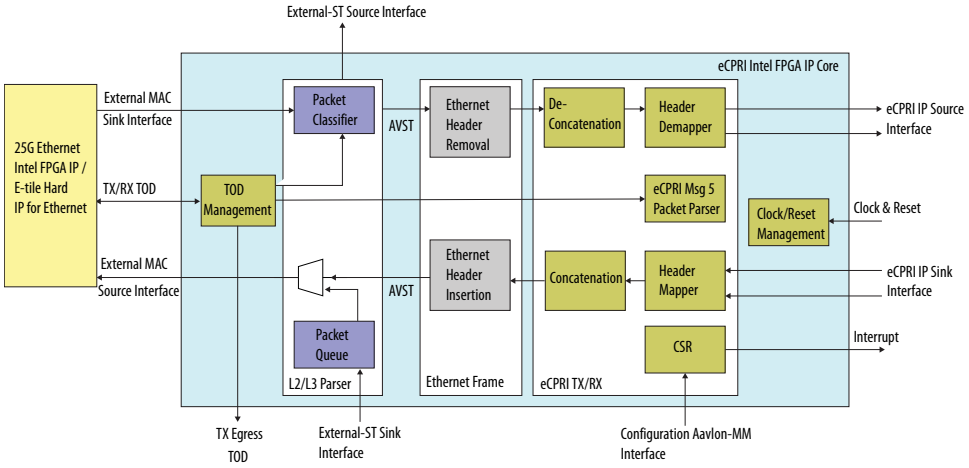
- **External ST Sink Interface**

  This interface provides datapath from client logic to eCPRI IP. This interface is a primary input for PTP and C&M messages. This interface complies with Avalon-Steaming specification as defined in the *Avalon Interface Specifications*.

- **TX and RX Time-of-Day (TOD) Interface**:

  This interface provides 96-bit timestamp from PTP module to eCPRI IP core and to client logic.

Figure 6. eCPRI Intel FPGA IP High-Level System Overview



Related Information

- eCPRI Specification V1.2
- Avalon Interface Specifications
- 1588 Precision Time Protocol Interfaces
  For 25G Ethernet Intel Stratix 10 Intel FPGA IP
- 1588 Precision Time Protocol Interfaces
  For E-tile Hard IP for Ethernet Intel FPGA IP

4.2. High Level Data Path Flow

The eCPRI IP core consists of two paths:

- Transmit TX path
- Receive RX Path

4.2.1. Transmit TX Path

There are two sets of Avalon-ST source and sink interface signals available to the incoming packets on the transmit TX path. Avalon-ST source/sink interface connects to the eCPRI IP and external source/sink interface connects to external user logic. The incoming eCPRI packets passes through Ethernet header insertion block to insert Ethernet header, optionally with different VLAN tags, IPv4, and UDP headers configured during configuration time.

You can send different types of packets through the external source/sink interface signal (For example, C&M and synchronization packets) which arbitrates with eCPRI packets and the IP sends winning packets to Ethernet MAC for transmission. The incoming external user packets are expected to arrive with Ethernet MAC header inserted on the packets.

The priority of the packets send to Ethernet MAC is listed as below, with highest priority order from top to bottom:

- PTP synchronization packet
- eCPRI packet
- C&M packets and remaining type of packets

The C&M and PTP synchronization packets are send/receive through external source/sink interface signal. The C&M and PTP synchronization packets are generally low bandwidth traffic. When there is collision between external PTP synchronization packets and eCPRI packets, backpressure to the eCPRI IP occurs to stop eCPRI packets from transmitting. The eCPRI IP implements a counter to track the number of eCPRI packets and PTP packets granted and raise the priority of the C&M packet when the counter reaches a programmable threshold to allow the C&M packet transmission to Ethernet MAC and avoid starvation.

You need to ensure that the bandwidth of external source/sink interface signal won't starve the overall bandwidth and cause interruption on eCPRI traffics.

4.2.2. Receive RX Path

The receiving Ethernet frames from the Ethernet MAC first enters packet classifier block. Packet classifier block classifies the packet into eCPRI packets and non-eCPRI packets. The packet classifier sends eCPRI packets with matching MAC address to the next component while sends all other non-eCPRI packets to external user logic for processing. For detailed information on conditions when the IP classifies packet as eCPRI packet, refer to section *Packet Classifier*.

Related Information

Packet Classifier on page 18

4.2.3. Supported Ethernet Variants

The eCPRI Intel FPGA IP pairs together with the 25G Ethernet MAC. The eCPRI IP is validated together with the 25G Ethernet MAC. The IP supports **25G Ethernet Intel FPGA IP** for H-tile and **E-tile Hard IP for Ethernet Intel FPGA IP** for E-tile variants. You must generate **25G Ethernet Intel FPGA IP** or **E-tile Hard IP for Ethernet Intel FPGA IP** with **Enable IEEE 1588** parameter enabled for the eCPRI IP to support client PTP message and eCPRI one-way delay measurement. The 25G

Ethernet MAC only supports 96-bit (V2) timestamp format. If you use **25G Ethernet Intel FPGA IP**, you must generate this IP with **Enable preamble passthrough** and **Enable TX CRC passthrough** parameters disabled.

Related Information

- 25G Ethernet Intel Stratix 10 FPGA IP User Guide
- E-tile Hard IP User Guide

4.3. Operation of the eCPRI IP Blocks

The following section explains the operation of the eCPRI IP blocks.

4.3.1. Packet Classifier

The packet classifier parses the incoming Ethernet frame to identify the types of incoming packets. The incoming packets could be eCPRI packet, PTP packet, or C&M packet with different types of frames (e.g., standard Ethernet frame, IPv4, and etc.)

Packet classifier redirects eCPRI packets to next component for further processing and classifies a packet as eCPRI packet if all the condition listed in the table below met. The packet classifier sends all non-eCPRI packets to external Avalon-ST interface.

Table 7. Ethernet Frame Format (User Data over Ethernet)

Number of Bits	RX Frame	Condition
48	MAC Destination address	Destination MAC address matches receiver source MAC address.
48	MAC Source address	Do not check.
32 (Optional)	VLAN tag	Packet parser checks for VLAN tag and adjust the offset accordingly.
32 (Optional)	Stack VLAN tag	Packet parser checks for SVLAN tag and adjusts the offset accordingly.
16	Ethertype (2 Bytes)= IP	Ethertype is equal to 0xAEFE

If eCPRI message transmitted over IP/UDP, the IP supports only IPv4 with UDP.

Table 8. Ethernet Frame Format with IPv4 (User Data over IP)

No. of Bits	IPv4 Header	Condition
48	MAC destination address	Destination MAC address matches receiver source MAC address.
48	MAC source address	Do not check.
32 (Optional)	VLAN Tag	Packet parser checks for VLAN tag and adjusts the offset accordingly.
32 (Optional)	Stack VLAN Tag	Packet parser checks for SVLAN tag and adjusts the offset accordingly.
16	Ethertype (2B) = IP	Ethertype must be 0x0800 for IPv4.
4	Version	Version must be 4'h4.
4	Internet Header Length	The value must be 4'h5. The IP does not support IPv4 "Options" field.

continued...

No. of Bits	IPv4 Header	Condition
6	Differentiated Services Code Point (DSCP)	Do not check.
2	Explicit Congestion Notification (ECN)	Do not check.
16	Total length	Do not check.
16	Identification	Do not check.
3	Flags	Do not check.
13	Fragment offset	Do not check.
8	Time To live	Do not check.
8	Protocol	The protocol value must be equal to 0x11 for IPv4 with UDP.
16	Header checksum	Do not check.
32	Source address	Do not check.
32	Destination address	Destination IP address matches receiver source IP address.
16	Source port	Do not check.
16	Destination port	Destination port number matches receiver UDP port number.
16	Length	Do not check.
16	Checksum	Do not check.

4.3.2. Ethernet Header Insertion/Removal

The Ethernet header insertion block inserts Ethernet header to incoming eCPRI packet on TX path. Optionally it can insert IPv4/UDP headers to the packet based on the configuration and the Ethernet header removal block removes Ethernet header to incoming eCPRI packet on RX path. Optionally it can remove IPv4/UDP headers to the packet based on the configuration selected. The Ethernet header encapsulated the incoming eCPRI packets as shown in the figure below. The table listed the source of each fields within the Ethernet header.

Figure 7. Ethernet Header Field

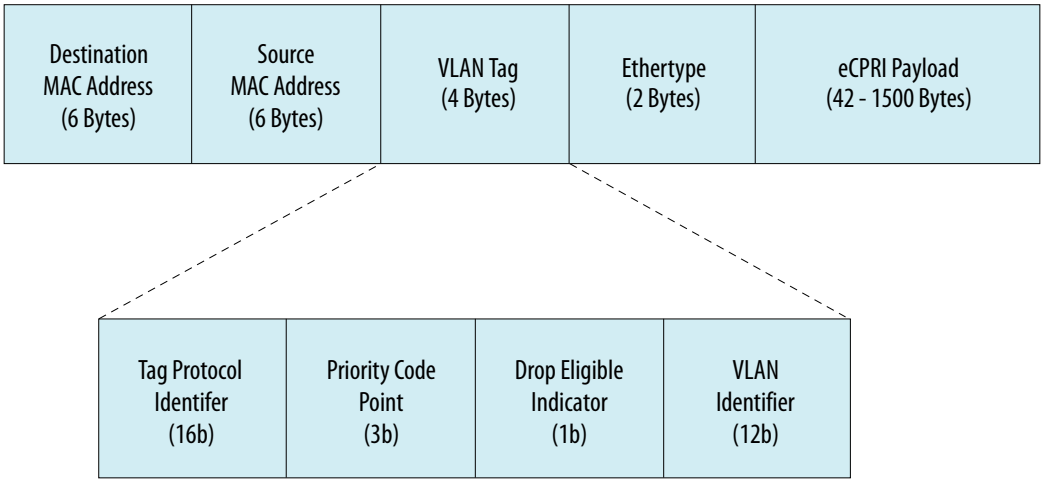


Table 9. Ethernet Header Field and CSR

Ethernet Header Field	CSR
Destination MAC address	Destination MAC address <N=0,1,2,3,4,5,6,7> Register 0,1 With enabled Advance mapping mode: N – eCPRI message PCID [2:0] Default mapping mode: N = 0 for all eCPRI message
Source MAC address	Source MAC address register 0, and 1
VLAN tag	VLAN Tag Register <N=0,1,2,3,4,5,6,7> With Enabled advance mapping mode: N – eCPRI message PCID [2:0] Default mapping mode: N = 0 for all eCPRI message
Ethertype	AEFE
eCPRI payload	Incoming eCPRI packet from the eCPRI IP

If you select IPv4 header as encapsulation to eCPRI payload, the following table lists the CSR to fill the IPv4 header fields:

Table 10. IPv4 Field and CSR

Number of Bits	IPv4 Header	CSR
48 (6 Bytes)	MAC destination Address	Refer to the *Table: Ethernet Header Field and CSR* above.
48 (6 Bytes)	MAC source address	Refer to the *Table: Ethernet Header Field* above.
16 (2 Bytes)	Ethertype (2 Bytes)= IP	0x0800
4	Version	`Ipv4_dw0_address`
4	Internet header length	`Ipv4_dw0_address`
6	Differentiated Services Code Point (DSCP)	`Ipv4_dw0_address`

continued...

Number of Bits	IPv4 Header	CSR
2	Explicit Congestion Notification (ECN)	`Ipv4_dw0_address`
16	Total length	eCPRI IP calculates incoming payload length
16	Identification	`Ipv4_dw1_address`
3	Flags	`Ipv4_dw1_address`
13	Fragment offset	`Ipv4_dw1_address`
8	Time to live	`Ipv4_dw2_address`
8	Protocol	`Ipv4_dw2_address`
16	Header checksum	eCPRI IP calculates incoming payload checksum
32	Source address	`Ipv4_src_address_0`
32	Destination address	`Ipv4_dst_address_0`

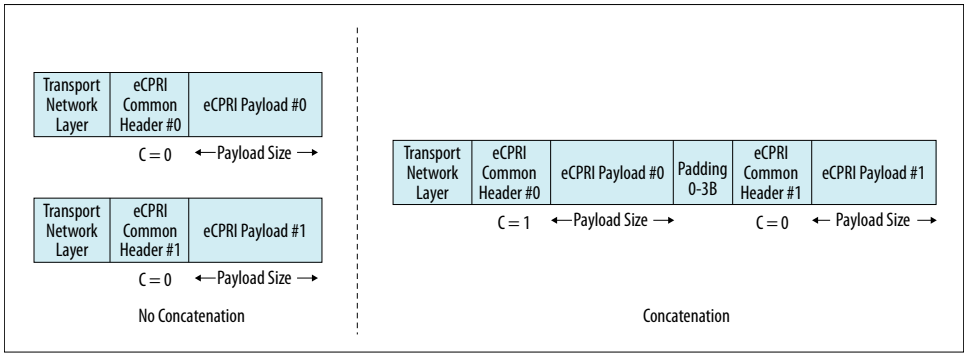
Table 11. UDP Field and CSR

Number of Bytes	IPv4 Header	CSR
14	Ethernet header	Refer to the *Table: Ethernet Header Field and CSR* above.
20	IPv4 header	Refer to the *Table: IPv4 Field and CSR Header* above.
2	Source port	`mudp_dw0_address`
2	Destination port	`mudp_dw0_address`
2	Length	eCPRI IP calculates incoming payload length
2	Checksum	eCPRI IP calculates incoming payload length

4.3.3. Concatenation/De-concatenation

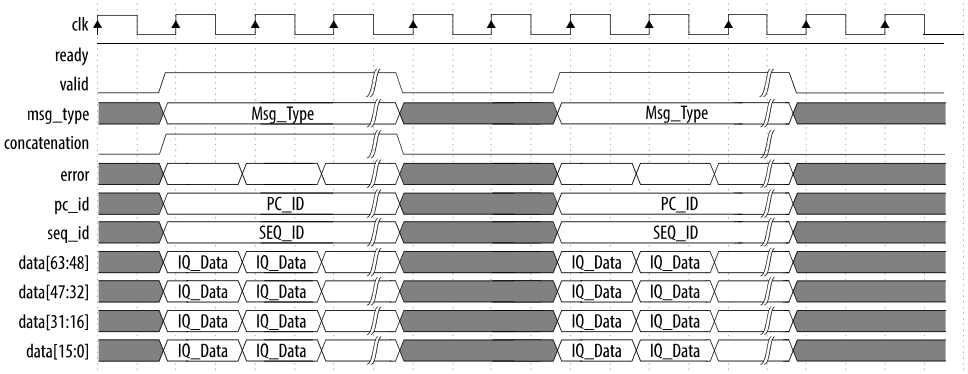
The Concatenation/De-concatenation blocks of the eCPRI IP implements concatenation logic of the eCPRI messages into single Ethernet frame or single IP/UDP packet. The `sink_concatenation` sideband signal identifies packets that required concatenation. The below diagrams illustrates the eCPRI messages with and without concatenation.

Figure 8. eCPRI Message Concatenation



When multiple eCPRI messages are concatenated together, 0 to 3 "zero" padding bytes are added if the following message does not start at a 4 byte boundary. The payload size specified in the eCPRI common header does not include this extra zero padding bytes.

Figure 9. Concatenation/De-concatenation Example Waveform



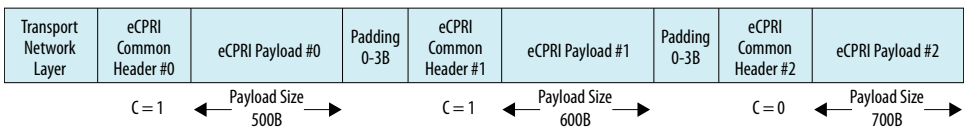
The above waveform shows two incoming eCPRI messages entering to eCPRI IP, first eCPRI message with concatenation sideband interface signal = 1 and the second eCPRI message with concatenation sideband interface signal = 0. These 2 eCPRI messages are combined and send through single transport network layer protocol. On the receiving end, the combined eCPRI message will then de-concatenate into 2 eCPRI messages and output to Avalon-ST interface.

There is a timeout counter used to detect the end of the concatenation message. If the counter overflows and no message with C=0 is detected, an error will be logged and the message with C=1 will be converted to message with C=0 and send to MAC.

The message type allowed for concatenation is restricted to message type 0,1,2,3 and 6. The de-concatenation is supported on all message type except message type 5.

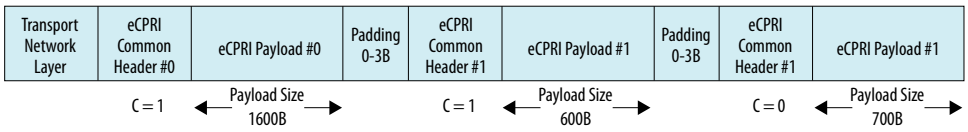
There are 2 different eCPRI packet concatenation scenarios which trigger error and it is shown in below diagram.

Figure 10. eCPRI Packet Concatenation Scenario 1



In the first scenario, there are three incoming Avalon-ST packets payload size of 500 bytes, 600 bytes and 700 bytes. The total payload size after concatenation is 1800 bytes which is bigger than maximum eCPRI IP supported MTU size of 1500 bytes. In this case, error will be logged in the eCPRI TX error message register and payload 0 and 1 will be sent as concatenated packets while payload 2 will be sent by itself.

Figure 11. eCPRI Packet Concatenation Scenario 2



In the second scenario, the first packet payload size is more than 1500 bytes. In this case, all the packets drop and error logged in eCPRI TX error message register.

4.3.4. Header Mapper/De-Mapper

The Header mapper/De-mapper block append or remove the eCPRI common header from the eCPRI message. The Mapper block calculates the payload size of the incoming Avalon-ST packet and append it into the packet as part of the eCPRI common header field. The table below shows the eCPRI common header format. The eCPRI protocol version is a read only field and the **Protocol Revision** parameter determines the value of this field. The concatenation and message type are determined from the Avalon-ST sink sideband interface signals which come along with eCPRI message. The payload size is calculated when the eCPRI message enter eCPRI IP at Avalon-ST interface.

Table 12. eCPRI Common Header Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
eCPRI Protocol Version= 0001b				Reserved			Concatenation	1
eCPRI Message Type								1
eCPRI Payload Type								2

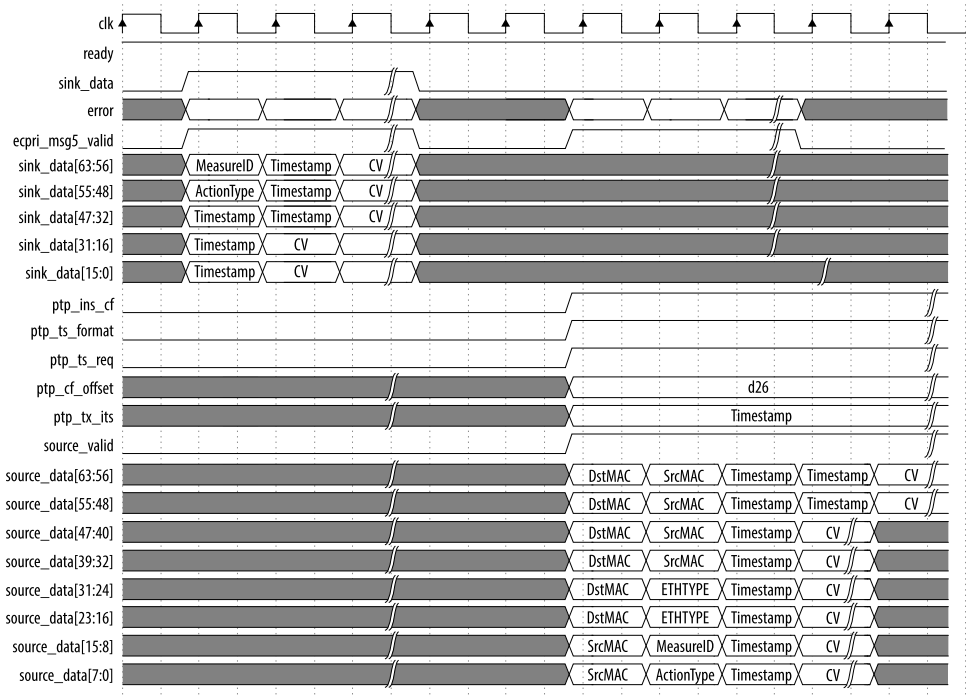
4.3.5. eCPRI Message 5 Packet Parser

This block is responsible to initiate and calculate the eCPRI one-way delay measurement on the transport link. The process is initiated when a CSR write to eCPRI Message 5 Control Register. The packet parser assembles an eCPRI message 5 with timestamp t1 taken from Time-of-Day (TOD) module. Then, this eCPRI message 5 is sent through the Ethernet MAC with compensation value cv1 filled using 1588 PTP hardware.

On the receiving end, the eCPRI IP responses the message 5 with t2 and cv2. Upon receiving the response packet, this calculates the transport delay using the formula:$t_{D12} = (t_2 - t_{CV2}) - (t_1 + t_{CV1})$

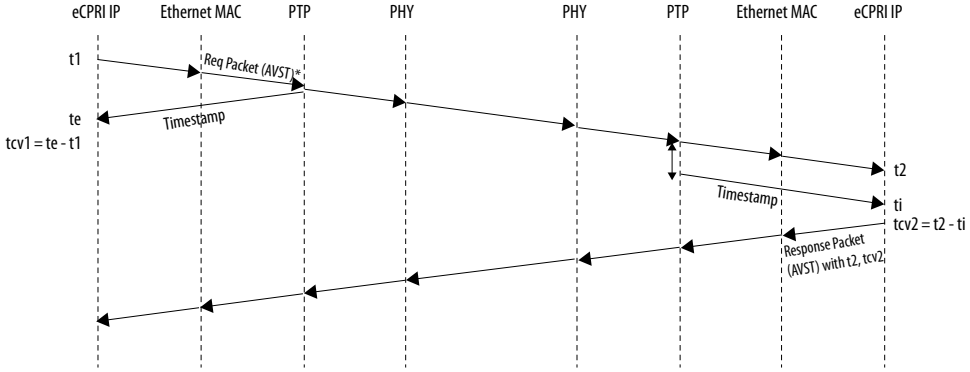
The waveform below shows an example of the Avalon-ST source and sink data through L2/L3 parser. The example in this section uses E-tile Ethernet Hard IP with 1588 PTP feature enabled.

Figure 12. Timing Diagram of One-Way Delay Measurement Example



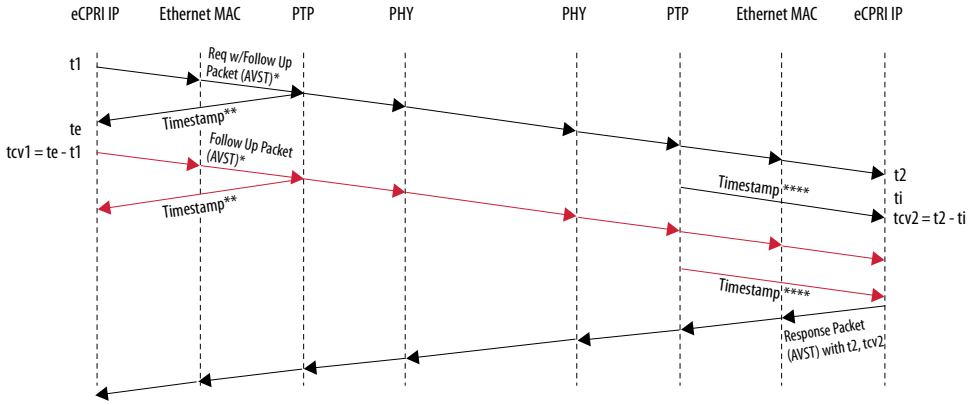
The timing diagram below illustrates the eCPRI message 5 in one-step one way delay measurement.

Figure 13. Timing Diagram of eCPRI Message Type 5 in one-step



The timing diagram below illustrates the eCPRI message 5 in two-step one way delay measurement.

Figure 14. Timing Diagram of eCPRI Message Type 5 in two-step



The one and two- steps one way delay measurement sequences uses same remote request type. The only difference is destination eCPRI IP measures t1 and tcv1 while the source eCPRI IP measures t2 and tcv2.

4.3.6. Packet Queue

This block is responsible to stage user incoming Ethernet frames (e.g., Control and Management packets, synchronization packets & etc) and arbitrate with eCPRI packets. These user Ethernet frames share the same Ethernet link with eCPRI packets. eCPRI IP does not encapsulate Ethernet header to these frames.

4.3.7. eCPRI Message Type

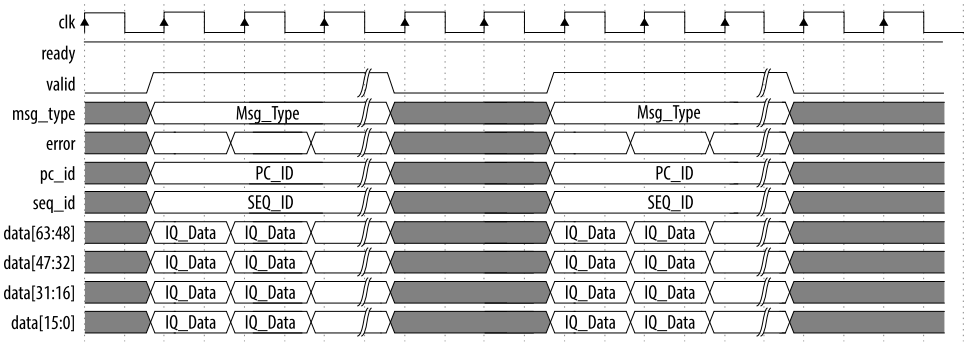
This section covers information about different types of eCPRI messages supported by eCPRI Intel FPGA IP.

4.3.7.1. eCPRI Message Type 0- IQ Data Transfer

Table 13. eCPRI Message Type 0- IQ Data Transfer Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
PC_ID								2
SEQ_ID								2
IQ_DATA								L

Figure 15. eCPRI Message Type 0- IQ Data Transfer Message Timing Diagram



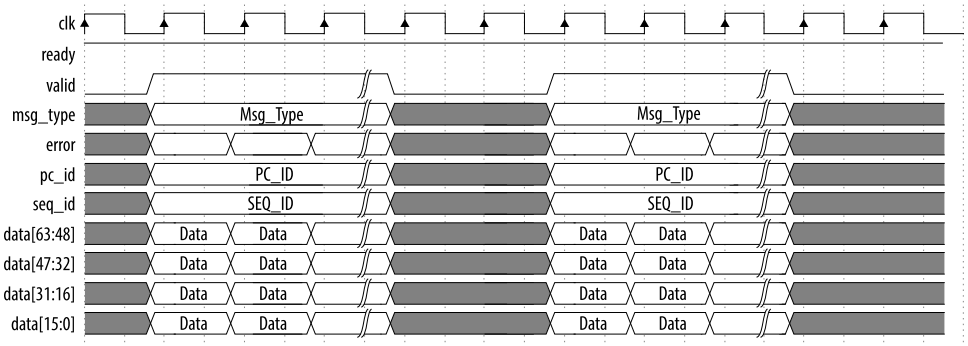
Note: The PC_ID, and SEQ_ID fields are 2 bytes wide. The PC_ID and PC_ID sideband interfaces are 4 bytes wide, so the MSB 2 bytes are set to zero.

4.3.7.2. eCPRI Message Type 1- Bit Sequence Transfer

Table 14. eCPRI Message Type 1- Bit Sequence Transfer Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
PC_ID								2
SEQ_ID								2
Bit Sequence of User Data								L

Figure 16. eCPRI Message Type 1 – Bit Sequence Transfer Message Timing Diagram

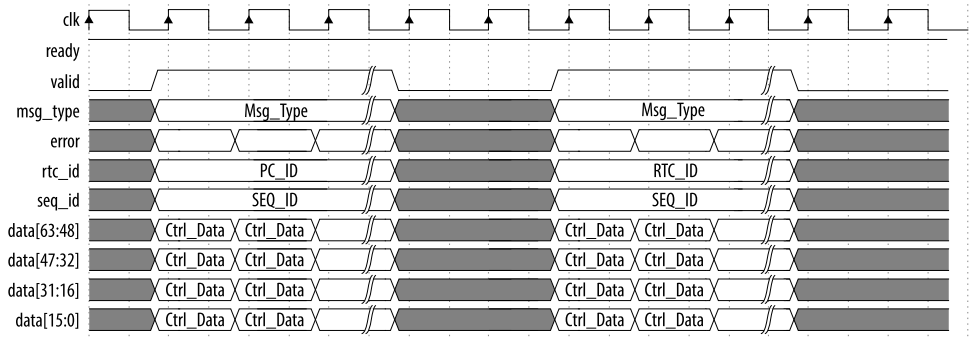


4.3.7.3. eCPRI Message Type 2- Real Time Control Data

Table 15. eCPRI Message Type 1- Real Time Control Data Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
PC_ID								2
SEQ_ID								2
Real time control data								L

Figure 17. eCPRI Message Type 2 – Real Time Control Data Message Timing Diagram

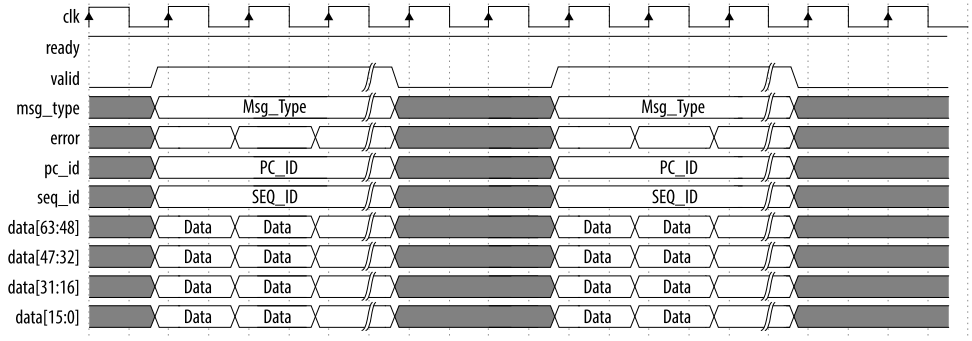


4.3.7.4. eCPRI Message Type 3- Generic Data Transfer

Table 16. eCPRI Message Type 3- Generic Data Transfer Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
PC_ID								2
SEQ_ID								2
Data transferred								L

Figure 18. eCPRI Message Type 3 – Generic Data Transfer Message Timing Diagram



4.3.7.5. eCPRI Message Type 4- Remote Memory Access

Table 17. eCPRI Message Type 4- Remote Memory Access Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
Remote Memory Access ID								1
Read/Write				Req/Resp				1
Element ID								2
Address								6
Length								2
Data								L

The eCPRI IP core supports two different modes for remote memory access message type: basic mode and buffer mode. The basic mode provides direct tunneling on the memory access, with all the necessary information for the memory access output to user logic. User logic is responsible to send the same ID/Element ID upon responding to the original request.

The buffer mode keeps the receiving request ID/Element ID/Address/read/write operations per table below. User logic carries out the operation and response to the original request with number of read/write bytes and read data. eCPRI IP appends necessary fields and then send the eCPRI message back to sender.

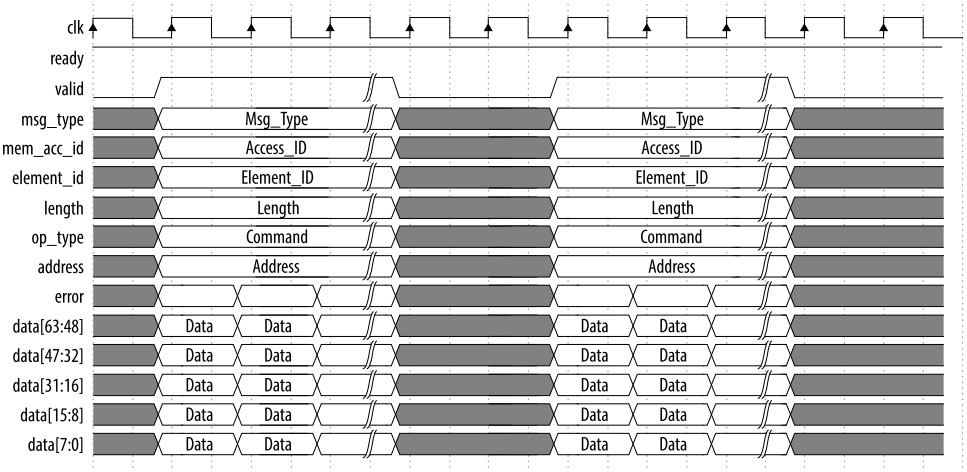
eCPRI IP expects the request and response to be in-order for this buffer mode. eCPRI IP can hold up to a maximum of eight pending requests. If there are eight pending requests in the queue and there is additional request received, the IP drops the additional request and logs error in RX error register.

eCPRI IP operates in basic mode by default.

Table 18. Parameter Handling

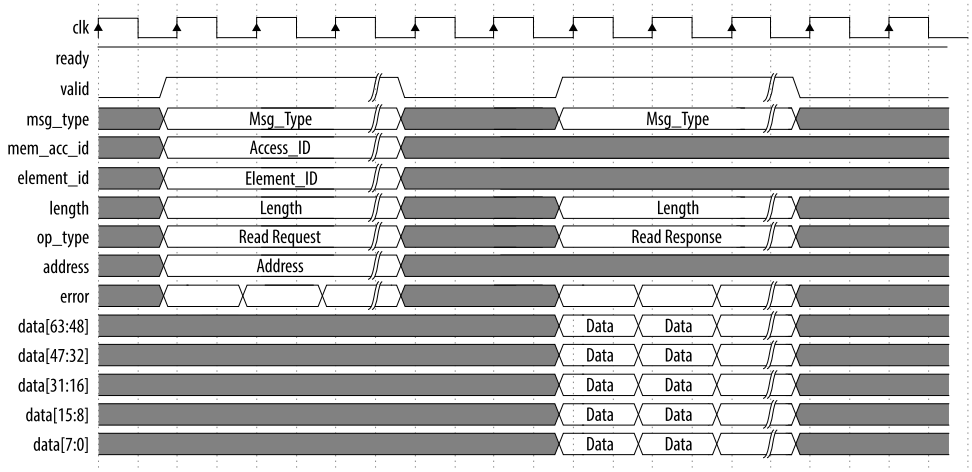
Action	ID	Read/Write	Request/ Response	Element ID	Address	Length	Data
Read request	Set	Set to read	Set to request	Set	Set	Set	No data
Read response	Copied	Copied	Set to response	Copied	Copied	No. of read bytes	Read data
Write request	Set	Set to write	Set to request	Set	Set	Set	The data to be written
Write response	Copied	Copied	Set to response	Copied	Copied	No. of written bytes	Vendor specific
Write no response	Set	Set to write no response	Set to request	Set	Set	Set	The data to be written
Failure response	Copied	Copied	Set to Failure	Copied	Copied	Vendor specific	Vendor specific

Figure 19. eCPRI Message Type 4 – Remote Memory Access in Basic Mode Message Timing Diagram



The waveform below illustrates the buffer mode where read request is sent to user logic. User logic response with read data and the actual length of the operation and message type 4. eCPRI IP extracts the memory access ID/element ID and address from internal buffer and combine with read data to send back to sender.

Figure 20. eCPRI Message Type 4 – Remote Memory Access Message in Buffer Mode Timing Diagram



4.3.7.6. eCPRI Message Type 5- One-Way Delay Measurement

Table 19. eCPRI Message Type 5- One Way Delay Measurement Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
Measurement ID								1
Action Type								1
Timestamp								10
Compensation Value								8
Dummy Bytes								L

Related Information

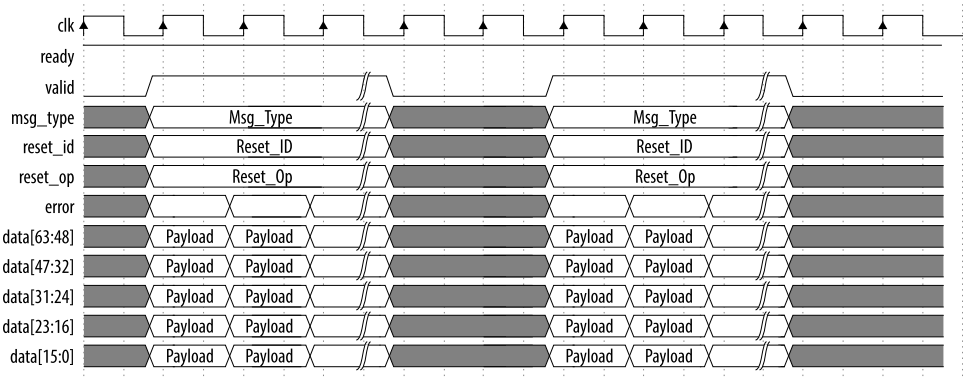
eCPRI Message 5 Packet Parser on page 23
Refer to this section for more information on one-way delay measurement.

4.3.7.7. eCPRI Message Type 6- Remote Reset

Table 20. eCPRI Message Type 6- Remote Reset Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
Reset ID								2
Reset Code Op								1
Payload								L

Figure 21. eCPRI Message Type 6 – Remote Reset Message Timing Diagram



Related Information

eCPRI Specification V1.2

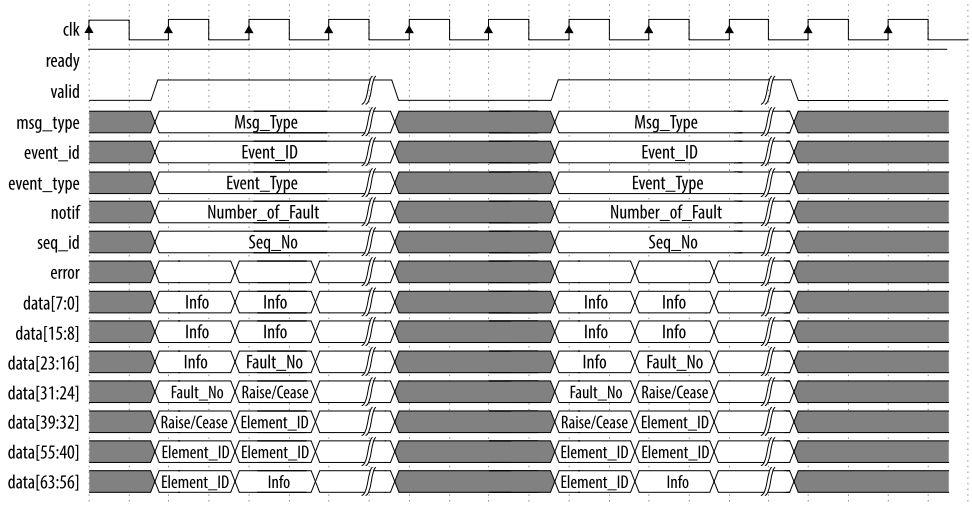
Refer to this specification for information on Op code.

4.3.7.8. eCPRI Message Type 7- Event Indication

Table 21. eCPRI Message Type 7- Event Indication Message Format

0 (MSB)	1	2	3	4	5	6	7 (LSB)	No. of Bytes
Event ID								1
Event Type								1
Sequence Number								1
Number Of Faults/Notif = N								1
Element ID # N								2
Raise/Cease #N				Fault/Notif #N MSB				1
Fault/Notif #N LSB								1
Additional Information								4

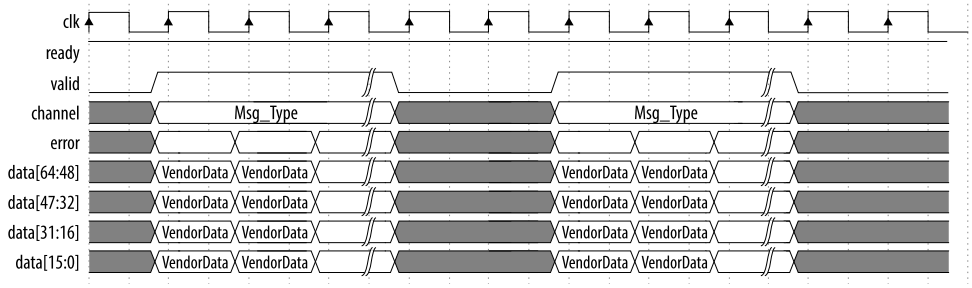
Figure 22. eCPRI Message Type 7 – Event Indication Message Timing Diagram



4.3.7.9. eCPRI Message Type 64- 255 Vendor Specific

Vendor specific eCPRI message types are not defined in the eCPRI specification. The eCPRI IP allows a direct pass through for vendor specific data. The output of the eCPRI IP for this message type is eCPRI common header and vendor specific data.

Figure 23. eCPRI Message Type 64 – 255 Vendor Specific Message Timing Diagram



4.3.8. Error Handling

Table 22. Error Condition Behavior

Events	Hardware Logging	Mitigations
Invalid measurement ID received on eCPRI message type 5	Log last error measurement ID and action type in eCPRI RX error message register.	None
Timeout no response for eCPRI message type 5	Log pending measurement ID and action type in eCPRI RX error message register.	None
Timeout no end of concatenation message received	eCPRI TX error message register.	Convert last message to C=0 and send out the messages.
Invalid eCPRI message types, 8 to 63	Log last error message type in eCPRI TX error message register	eCPRI message drop.

continued...

Events	Hardware Logging	Mitigations
Invalid message type 5 action types	eCPRI RX error message register	None
Multiple message concatenation size is greater than MTU	eCPRI TX error message register.	Split the messages into 2 or more PDU and send out the messages.
Single message concatenation size is greater than MTU	eCPRI RX error message register.	eCPRI message drop.
Timeout no reset access response	eCPRI RX error message register.	None
Timeout no memory access response	Log last memory access ID and op code in eCPRI RX error message register.	None
Missing start of packet (SOP)	eCPRI TX/RX error message register.	Incoming data drop.
Missing end of packet (EOP)	eCPRI TX/RX error message register.	Incoming data drop.
Buffer overflow	eCPRI TX/RX error message register.	None
M20K ECC	eCPRI TX/RX error message register.	None
Rx eCPRI payload length not match payload size	eCPRI RX error message register	eCPRI error asserted only on the last packet of the concatenated packet and Avalon-ST error interface asserted at EOP. That means earlier packet(s) have integrity issues.
RX eCPRI invalid concatenation bit	eCPRI Rx Error Message Register	eCPRI error asserted only on the last packet of the concatenated packet and Avalon-ST error interface asserted at EOP. That means earlier packet(s) have integrity issues.

The eCPRI IP behaves as follows upon observing timeout error due to multiple no memory access responses in receiver:

- When first request timeout due to no response, first timeout counter stops counting and error interrupt triggered.
- Second request timeout due to no response happen, second timeout counter stops counting as well. Now there are two errors pending in IP.
- Software service the interrupt routine and determine the error is due to timeout no response. The software clears the error.
- Interrupt is deasserted and then asserted again due to second error and software handling of error is repeated.

It is software responsibility to handle the timeout error to avoid software hang due to pending memory access response.

Related Information

eCPRI Specification V1.2
Refer to this specification for information on Op code.

4.3.9. RX Throttling

The eCPRI Intel FPGA IP doesn't support throttling on the RX side (from 25G Ethernet MAC to eCPRI). Packets from Ethernet MAC are continuously stream out either to external-ST source interface or eCPRI IP source interface and you can expect to allocate enough buffer to hold the packets.

5. Interface Overview

The eCPRI IP core communicates with the surrounding design though multiple external signals.

5.1. Clock Signals

Table 23. eCPRI IP Input Clocks

Signal Name	Width (Bits)	I/O Direction	Description
clk_tx	1	Input	Ethernet MAC TX clock. For the Intel Stratix 10 H-tile IP variations, the default frequency value is 390.625 MHz. For the Intel Stratix 10 E-tile IP variations, the default frequency value is 402.835 MHz.
clk_rx	1	Input	Ethernet MAC RX clock. For the Intel Stratix 10 H-tile IP variations, the default frequency value is 390.625 MHz. For the Intel Stratix 10 E-tile IP variations, the default frequency value is 402.835 MHz.
clk_csr	1	Input	CSR clock. The default frequency value is 100 MHz to 162 MHz.
ext_sink_clk	1	Input	External user interface clock. The frequency value is greater than or equal to 390.625 MHz.

5.2. Power, Reset, and Firewalls Signals

Table 24. eCPRI IP Reset, Power, and Firewalls Signals

These signals are asynchronous.

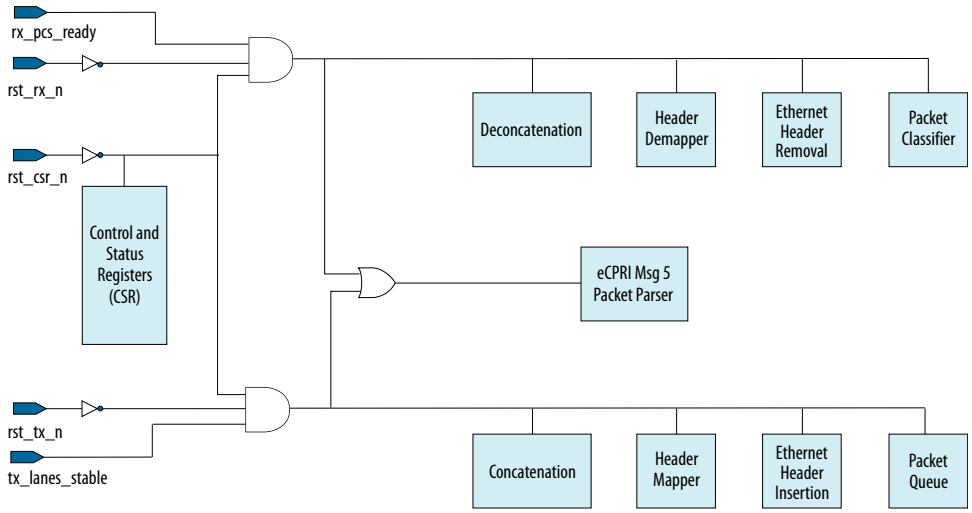
Signal Name	Width (Bits)	I/O Direction	Description
rst_tx_n	1	Input	Reset signal from Ethernet MAC TX. Resets the eCPRI IP in RX direction. Resets the De-concatenation, Header mapper/De-mapper,

continued...

Signal Name	Width (Bits)	I/O Direction	Description
			Ethernet header removal, eCPRI message 5 packet parser and, Packet classifier.
rst_rx_n	1	Input	Reset signal from Ethernet MAC RX. Resets the eCPRI IP in TX direction. Resets the Concatenation, Header mapper/De-mapper, Ethernet header insertion, eCPRI message 5 packet parser, and Packet queue.
rst_csr_n	1	Input	Reset signal for CSR logic. Resets the eCPRI IP control and status registers. When asserted, resets the eCPRI IP.
tx_lanes_stable	1	Input	Signal that indicates the clk_tx signal from MAC is stable and ready for operation.
rx_pcs_ready	1	Input	Signal that indicates the clk_rx signal from MAC is stable and ready for operation.

5.2.1. Reset Control and Initialization Flows

Figure 24. eCPRI IP Core Reset Logic



Three reset ports of the eCPRI IP assert together to fully reset the eCPRI IP. The deassertion of these three signals can happen together or the IP can just deassert rst_csr_n signal follows by rst_tx_n and rst_rx_n signals depending on use case.

You should perform reset before beginning IP core operation. Alternatively, you can trigger reset after you reconfigure the eCPRI IP during run time.

Reset Length Requirement

You need to assert reset signals for additional ten cycles after `tx_lanes_stable`, and `rx_pcs_ready` signals are asserted to ensure the Ethernet MAC clocks are stable and run at designated speed. The `avst_sink_ready`, and `mac_sink_ready` signals are asserted when the IP core exists from reset successfully and ready to accept client data.

5.3. TX Time of Day Interface

Table 25. Signals of the TX Time of Day Interface

All signals are synchronous to `clk_tx` clock.

Signal Name	Width (Bits)	I/O Direction	Description
`tx_tod_time_of_day_96b_data`	96	Input	Current V2-format (96-bit) TOD in `clk_txmac` clock domain.
`tx_egress_timestamp_96b_data`	96	Input	Provides the V2-format timestamp when a 1588 PTP frame begins transmission on the Ethernet link. Value is valid when the `tx_egress_timestamp_96b_valid` signal is asserted. This signal is present only in two-step clock mode.
`tx_egress_timestamp_96b_valid`	1	Input	Indicates that the `tx_egress_timestamp_96b_data` and `tx_egress_timestamp_96b_fingerprint` signals are valid in the current `clk_txmac` clock cycle. This signal is present only in two-step clock mode.
`tx_egress_timestamp_96b_fingerprint`	4	Input	Provides the fingerprint of the V2-format 1588 PTP frame currently beginning transmission on the Ethernet link. Value is valid when the `tx_egress_timestamp_96b_valid` signal is asserted. Valid values are: • 0: Unused • 1: eCPRI one way delay measurement packet • 2: 1588 PTP packet

Related Information

- 1588 PTP Interface Signals
  For more information on 1588 PTP signals for the 25G Ethernet Intel Stratix 10 IP.
- 1588 PTP Interface
  For more information on 1588 PTP signals for the E-tile Hard IP for Ethernet.

5.4. RX Time of Day Interface

Table 26. Signals of the RX Time of Day Interface

All signals are synchronous to `clk_rx` clock.

Signal Name	Width (Bits)	I/O Direction	Description
`rx_tod_time_of_day_96b_data`	96	Input	Current V2-format (96-bit) TOD in `clk_rxmac` clock domain.
`rx_ingress_timestamp_96b_data`	96	Input	Whether or not the current packet on the RX client interface is a 1588 PTP packet, indicates the V2-format timestamp when the IP core received the packet on the Ethernet link. The IP core provides a valid value on this signal in the same cycle it asserts the RX SOP signal for 1588 PTP packets.
`rx_ingress_timestamp_96b_valid`	1	Input	Indicates that the `rx_ingress_timestamp_96b_data` signal is valid in the current cycle. This signal is redundant with the RX SOP signal for 1588 PTP packets.
`ext_rx_ingress_timestamp_96b_data`	96	Output	Indicates V2-format timestamp when the IP core receives the RX packet on the Ethernet link. The IP core provides a valid value on this signal in the same cycle it asserts the RX SOP signal for 1588 PTP packets.

Related Information

- 1588 PTP Interface Signals
  For more information on 1588 PTP signals for the 25G Ethernet Intel Stratix 10 IP.
- 1588 PTP Interface
  For more information on 1588 PTP signals for the E-tile Hard IP for Ethernet.

5.5. Interrupt

Table 27. Interrupt Signals

This signal is synchronous to `clk_csr` signal.

Signal Name	Width (Bits)	I/O Direction	Description
`err_interrupt`	1	Output	Error interrupt signal. Indicates errors occur in the eCPRI IP. Software can poll eCPRI error message register to determine the error info.

5.6. Configuration Avalon-MM Interface

Table 28. Signals of the Configuration Avalon-MM Interface

This section lists ports that provides access to internal control and status registers of the eCPRI IP. All signals are synchronous to `clk_csr`.

Signal Name	Width (Bits)	I/O Direction	Description
`csr_address`	16	Input	Configuration register address.
`csr_write`	1	Input	Configuration register write enable.
`csr_writedata`	32	Input	Configuration register write data.
`csr_read`	1	Input	Configuration register read enable.
`csr_readdata`	32	Output	Configuration register wait request.
`csr_writerequest`	1	Output	Configuration register read data valid.
`csr_readdatavalid`	1	Output	Configuration register read data valid.

5.7. Ethernet MAC Source Interface

Table 29. Signals of the 25G Ethernet MAC Avalon-ST Source Interface

This section lists port from eCPRI IP to 25G Ethernet MAC. All signals are synchronous to `mac_clk_tx`.

Signal Name	Width (Bits)	I/O Direction	Description
`mac_source_valid`	1	Output	Indicates Avalon source valid from eCPRI to Ethernet MAC.
`mac_source_data`	DATA_WIDTH	Output	Indicates Avalon source write data from eCPRI to Ethernet MAC.
`mac_source_sop`	1	Output	Indicates Avalon source start of packet (SOP) from eCPRI to Ethernet MAC. Indicate the beginning of packet.
`mac_source_eop`	1	Output	Avalon source end of packet (EOP) from eCPRI to Ethernet MAC. Indicate the end of packet.
`mac_source_empty`	LOG2(DATA_WIDTH/8)	Output	Avalon source empty from eCPRI to Ethernet MAC. Indicates the number of

continued...

Signal Name	Width (Bits)	I/O Direction	Description
			symbols that are empty, that is, do not represent valid data.
mac_source_ready	1	Input	Avalon source ready driven from Ethernet MAC. Indicate Ethernet MAC can accept data.
mac_source_error	1	Output	Avalon source error from eCPRI to Ethernet MAC. A bit mask to mark errors affecting the data being transferred in the current cycle.

5.7.1. E-tile Hard IP for Ethernet 1588 PTP Signals

Table 30. Signals of the E-tile Hard IP for Ethernet 1588 PTP Interface

All signals are synchronous to clk_tx clock.

Signal Name	Width	Direction	Description
ptp_timestamp_insert	1	Output	Inserts an egress timestamp into the current TX Packet on the respective channel. Valid only when the TX valid and TX SOP signals are asserted.
ptp_tx_etstamp_ins_ctrl_residence_time_update	1	Output	When asserted, inserts a residence time timestamp into the correction field in the current TX packet on the respective channel. Valid only when the TX valid and TX SOP signals are asserted.
i_ptp_zero_csum	1	Output	When asserted, overwrites the checksum in a UDP packet carried inside the current TX packet with zeros during IPv4. Valid only when the TX valid and TX SOP signals are asserted.
i_ptp_update_eb	1	Output	When asserted, overwrites the extended bytes field in an IPv6 packet carried inside the current TX packet with a value that cancels out changes to the checksum due to changes to the UDP packet. Valid only when the TX valid and TX SOP signals are asserted.
i_ptp_ts_format	1	Output	When asserted, selects the format of the PTP 1-step operation on the respective channel. Tie to 1 to indicate the use of IEEE 1588v2 timestamp and correction field formats (96 bits) Valid only when either the egress time timestamp signal (i_ptp_ins_ets) or the residence time timestamp signal (i_ptp_ins_cf), and the TX valid signal, and SOP signal are asserted.

continued...

Signal Name	Width	Direction	Description
ptp_offset_timestamp	16	Output	When asserted, indicates the position of the PTP timestamp field in the current TX packet. Valid only when the TX valid and TX SOP signals are asserted.
ptp_offset_correction_field	16	Output	When asserted, indicates the position of the PTP correction field in the current TX packet. Valid only when the TX valid and TX SOP signals are asserted.
i_ptp_csum_offset	16	Output	When asserted, indicates the position of the first byte of a UDP checksum field in the current TX packet. Valid only when the checksum overwrite in a UDP packet (e.g. i_ptp_zero_csum), TX valid, TX SOP signals are asserted.
i_ptp_eb_offset	16	Output	When asserted, indicates the position of the first byte of extended bytes field in the current TX packet. Valid only when the extended bytes overwrite in an IPv6 packet (e.g. i_ptp_update_eb), TX valid, TX SOP signals are asserted.
ptp_timestamp_request_valid	1	Output	Request a 2-step timestamp signal for the current TX packet. When asserted, generates a TX timestamp for the current packet. Valid only when the TX valid and TX SOP signals are asserted.
ptp_timestamp_request_fingerprint	4	Output	Fingerprint signal for current TX packet. Assigns an 8-bit fingerprint to a TX packet that is being transmitted, so that the 2-step or 1-step PTP/eCPRI one way delay measurement timestamp associated with the TX packet can be identified. The timestamp returns with the same fingerprint. Valid values: • 0 - Unused • 1 – eCPRI One-way delay measurement packet • 2 – 1588 PTP packet Valid only when the TX valid and TX SOP signals are asserted.
o_tx_ptp_ready	1	Input	TX PTP ready signal. When asserted, the core to ready to request for TX PTP functions on the respective channel.
o_rx_ptp_ready	1	Input	RX PTP ready signal. When asserted, indicates the RX PTP logic ready for use on the respective channel.

Related Information

1588 PTP Interface

For more information on 1588 PTP signals for the E-tile Hard IP for Ethernet.

5.7.2. 25G Ethernet MAC 1588 PTP Signals

Table 31. Signals of the 25G Ethernet MAC 1588 PTP

All signals are synchronous to `clk_tx` clock.

Signal Name	Width	Direction	Description
`ptp_timestamp_insert`	1	Output	Indicates the current packet on the TX client interface is a 1588 PTP packet and directs the IP core to process the packet in one-step processing insertion mode. In this mode, the IP core overwrites the timestamp of the packet with the timestamp when the packet appears on the TX Ethernet link. The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.
`ptp_tx_etstamp_ins_ctrl_residence_time_update`	1	Output	Indicates the current packet on the TX client interface is a 1588 PTP packet and directs the IP core to process the packet in one-step processing correction mode. In this mode, the IP core adds the latency through the IP core (residence time) to the current contents of the timestamp field. The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.
`tx_etstamp_ins_ctrl_times tamp_format`	1	Output	Specifies the timestamp format (V1 or V2 format) for the current packet if the TX client simultaneously asserts `tx_etstamp_ins_ctrl_timestamp_insert`. Valid value is: • Tie to 0 to indicate 96-bit timestamp format (V2). The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.
`tx_etstamp_ins_ctrl_residence_time_calc_format`	1	Output	Specifies the TOD format (Intel 64-bit TOD format or the V2 96-bit format) for the current packet if the TX client simultaneously asserts `tx_etstamp_ins_ctrl_residence_time_update`. Value is: • Tie to 0 to indicate 96-bit TOD format (V2) The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.
`ptp_offset_timestamp`	16	Output	Specifies the byte offset of the timestamp information in the current packet if the TX client simultaneously asserts `tx_etstamp_ins_ctrl_timestamp_insert`. The IP core overwrites the value at this offset.

continued...

Signal Name	Width	Direction	Description
			The TX client must maintain the desired value on this signal while the TX SOP signal is asserted. The timestamp has 96 bits. In this case, the IP core inserts ten bytes (bits [95:16]) of the timestamp at this offset and the remaining two bytes (bits [15:0]) of the timestamp at the offset specified in `tx_etstamp_ins_ctrl_offset_correction_field`.
`ptp_offset_correction_field`	16	Output	If the TX client simultaneously asserts `tx_etstamp_ins_ctrl_residence_time_update`, this signal specifies the byte offset of the correction field in the current packet. If the TX client simultaneously asserts `tx_etstamp_ins_ctrl_timestamp_insert` and deasserts (sets to the value of 0) the `tx_etstamp_ins_ctrl_timestamp_format` signal, this signal specifies the byte offset of bits [15:0]] of the timestamp. The TX client must maintain the desired value on this signal while the TX SOP signal is asserted.
`tx_etstamp_ins_ctrl_checksum_zero`	1	Output	The TX client asserts this signal during a TX SOP cycle to tell the IP core to zero the UDP checksum in the current packet. A zeroed UDP checksum indicates the checksum value is not necessarily correct. This information is useful to tell the application to skip checksum checking of UDP IPv4 packets. This function is illegal for UDP IPv6 packets.
`tx_etstamp_ins_ctrl_offset_checksum_field`	16	Output	Indicates the byte offset of the UDP checksum in the current packet. The TX client must ensure this signal has a valid value during each TX SOP cycle when it also asserts the `tx_etstamp_ins_ctrl_checksum_zero` signal. Holds the byte offset of the two bytes in the packet that the IP core should reset.
`tx_etstamp_ins_ctrl_checksum_correct`	1	Output	The TX client asserts this signal during a TX SOP cycle to tell the IP core to update (correct) the UDP checksum in the current packet. This signal is asserted for correct processing of UDP IPv6 packets.
`tx_etstamp_ins_ctrl_offset_checksum_correction`	16	Output	Indicates the byte offset of the UDP checksum in the current packet.

continued...

Signal Name	Width	Direction	Description
			The TX client must ensure this signal has a valid value during each TX SOP cycle when it also asserts the `tx_etstamp_ins_ctrl_checksum_correct` signal. Holds the byte offset of the two bytes in the packet that the IP core should correct. This signal is meaningful only in one-step clock mode.
`ptp_timestamp_request_valid`	1	Output	Indicates the current packet on the TX client interface is a 1588 PTP packet and directs the IP core to process the packet in two-step processing mode. In this mode, the IP core outputs the timestamp of the packet when it exits the IP core, and does not modify the packet timestamp information. The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet.
`ptp_timestamp_request_fingerprint`	4	Output	Fingerprint of the current packet. The TX client must assert and deassert this signal synchronously with the TX SOP signal for the 1588 PTP packet and eCPRI one way delay measurement packet.

Related Information

1588 PTP Interface Signals
For more information on 1588 PTP signals for the 25G Ethernet Intel Stratix 10 IP.

5.8. Ethernet MAC Sink Interface

Table 32. Signals of the 25G Ethernet MAC Sink Interface

This section lists port from 25G Ethernet MAC to eCPRI IP . All signals are synchronous to `mac_clk_rx`.

Signal Name	Width (Bits)	I/O Direction	Description
`mac_sink_valid`	1	Input	Indicates Avalon source valid from Ethernet to MAC eCPRI.
`mac_sink_data`	DATA_WIDTH	Input	Indicates Avalon source write data from Ethernet MAC to eCPRI.
`mac_sink_sop`	1	Input	Indicates Avalon sink start of packet (SOP) from Ethernet MAC to eCPRI. Indicate the beginning of packet.
`mac_sink_eop`	1	Input	Avalon source end of packet (EOP) from Ethernet MAC to eCPRI. Indicate the end of packet.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
mac_sink_empty	LOG2(DATA_WIDTH/8)	Input	Avalon source empty from Ethernet MAC to eCPRI. Indicates the number of symbols that are empty, that is, do not represent valid data.
mac_sink_ready	1	Output	Avalon sink ready driven from Ethernet MAC. Indicate eCPRI can accept data.
mac_sink_error	6	Input	Avalon sink error from Ethernet MAC to eCPRI. A bit mask to mark errors affecting the data being transferred in the current cycle.

5.9. External ST Source Interface

Table 33. Signals of the External ST Source Interface

All signals are synchronous to ext_sink_clk.

Signal Name	Width (Bits)	I/O Direction	Description
ext_source_valid	1	Output	Avalon source valid from L2/L3 parser to external user logic.
ext_source_data	DATA_WIDTH	Output	Avalon source write data from L2/L3 parser to external user logic.
ext_source_sop	1	Output	Avalon source start of packet from L2/L3 parser to external user logic. Indicate the beginning of packet.
ext_sink_eop	1	Output	Avalon source end of packet from L2/L3 parser to external user logic. Indicates the end of packet.
ext_source_empty	LOG2(DATA_WIDTH/8)	Output	Avalon source empty from L2/L3 parser to external user logic. Indicates the number of symbols that are empty, that is, do not represent valid data.
ext_source_error	1	Output	Avalon source error from L2/L3 parser to external user logic. A bit mask to mark errors affecting the data being transferred in the current cycle.

5.10. External ST Sink Interface

Table 34. Signals of the External ST Sink Interface

This table lists the ports from eCPRI IP to client logic. All signals are synchronous to ext_sink_clk.

Signal Name	Width (Bits)	I/O Direction	Description
ext_sink_valid	1	Input	Avalon sink valid from external user logic to L2/L3 parser.
ext_sink_data	DATA_WIDTH	Input	Avalon sink write data from external user logic to L2/L3 parser.
ext_sink_sop	1	Input	Avalon sink start of packet from external user logic to L2/L3 parser. Indicates the beginning of packet.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
ext_sink_eop	1	Input	Avalon sink end of packet from external user logic to L2/L3 parser. Indicates the end of packet.
ext_sink_empty	LOG2(DATA_WIDTH/8)	Input	Avalon sink empty from external user logic to L2/L3 parser. Indicates the number of symbols that are empty, that is, do not represent valid data.
ext_sink_ready	1	Output	Avalon sink ready driven from L2/L3 parser. Indicate L2/L3 parser can accept data.
ext_sink_error	1	Input	Avalon sink error from external user logic to L2/L3 parser. A bit mask to mark errors affecting the data being transferred in the current cycle.

5.11. eCPRI IP Source Interface

Table 35. Signals of the eCPRI IP Source Interface

This table lists the ports from eCPRI IP to client logic. All signals are synchronous to mac_clk_rx.

Signal Name	Width (Bits)	I/O Direction	Description
avst_source_valid	1	Output	Avalon source valid from eCPRI to client logic.
avst_source_data	DATA_WIDTH	Output	Avalon source write data from eCPRI to RRH PHY.
avst_source_sop	1	Input	Avalon source start of packet (SOP) from eCPRI to RRH PHY. Indicate the beginning of packet.
avst_source_eop	1	Input	Avalon source end of packet (EOP) from eCPRI to RRH PHY. Indicates the end of packet.
avst_source_empty	LOG2(DATA_WIDTH/8)	Input	Avalon source empty from eCPRI to RRH PHY. Indicates the number of symbols that are empty, that is, do not represent valid data.
avst_source_error	1	Output	Avalon source error from eCPRI to RRH PHY. A bit mask to mark errors affecting the data being transferred in the current cycle.
Sideband			
source_pc_id	32	Output	Indicates physical channel ID of eCPRI message. For message type 3, the physical channel id is 32-bit wide. For other message types, it is 16-bit wide and the 16 bit MSB is ignored. Valid on SOP assertion and stable until EOP assertion.
source_seq_id	32	Output	Indicates sequence ID of eCPRI message. For message type 3, the sequence id is 32- bit wide. For other message types, it is 16-bit wide and the 16 bit MSB is ignored. Valid on SOP assertion and stable until EOP assertion.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
source_rtc_id	16	Output	Real time control ID of eCPRI message. Valid on SOP assertion and stable until EOP assertion.
source_msg_type	8	Output	Indicates message type of the eCPRI message. Valid range is 0-7 and 64-255 for eCPRI v1.2 specification. Valid on SOP assertion and stable until EOP assertion.
source_mem_acc_id	8	Output	Indicates remote memory access id of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
source_op_type	8	Output	Indicates the operation of the eCPRI message type 4: • 8'b0000 0000 – Read Request • 8'b0000 0001 – Read Response • 8'b0001 0000 – Write Request • 8'b0001 0001 – Write Response • 8'b0010 xxxx – Write No Response • 8'bxxxx 0010 – Failure No Response Valid on SOP assertion and stable until EOP assertion.
source_element_id	16	Output	Indicates element ID of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
source_address	48	Output	Indicates memory address of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
source_length	16	Output	Indicates memory access length of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
source_reset_id	16	Output	Indicates reset ID of the eCPRI message type 6. Valid on SOP assertion and stable until EOP assertion.
source_reset_op	8	Output	Indicate reset operation type of the eCPRI message type 6: • 8'b0000 0000 – Remote Reset Request • 8'b0000 0001 – Remote Reset Response Others – Reserved Valid on SOP assertion and stable until EOP assertion.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
source_event_id	8	Output	Indicates event ID of the eCPRI message type 7. Valid on SOP assertion and stable until EOP assertion.
source_event_type	8	Output	Indicates event type of the eCPRI message type 7. Valid on SOP assertion and stable until EOP assertion.
source_notif	8	Output	Indicates number of faults for within eCPRI message type 7. Valid on SOP assertion and stable until EOP assertion.

5.12. eCPRI IP Sink Interface

Table 36. Signals of the eCPRI IP Sink Interface

This table lists the ports from client logic to eCPRI IP. All signals are synchronous to mac_clk_tx.

Signal Name	Width	Direction	Description
avst_sink_valid	1	Input	Avalon sink valid from RRH PHY to eCPRI
avst_sink_data	DATA_WIDTH	Input	Avalon sink write data from RRH PHY to eCPRI
avst_sink_sop	1	Input	Avalon sink start of packet (SOP) from RRH PHY to eCPRI. Indicate the beginning of packet.
avst_sink_eop	1	Input	Avalon sink end of packet (EOP) from RRH PHY to eCPRI. Indicate the end of packet.
avst_sink_empty	LOG2(DATA_WIDTH/8)	Input	Avalon sink empty from RRH PHY to eCPRI. Indicates the number of symbols that are empty, that is, do not represent valid data.
avst_sink_ready	1	Output	Avalon sink ready driven from eCPRI. Indicates eCPRI can accept data.
avst_sink_error	1	Input	Avalon sink error from RRH PHY to eCPRI. A bit mask to mark errors affecting the data being transferred in the current cycle.
Sideband			
sink_pc_id	32	Input	Physical channel ID of eCPRI message. For message type 3, the physical channel is 32-bit wide. For other message types, it is 16-bit wide and the 16 bit MSB is ignored. Valid on SOP assertion and stable until EOP assertion.
sink_seq_id	32	Input	Sequence ID of eCPRI message. For message type 3, the physical channel is 32-bit wide. For other message types, it is 16-bit wide and the 16 bit MSB is ignored. Valid on SOP assertion and stable until EOP assertion.
sink_rtc_id	16	Input	Real time control ID of eCPRI message. Valid on SOP assertion and stable until EOP assertion.
sink_concatenation	1	Input	Concatenation indication on the eCPRI message:

continued...

Signal Name	Width	Direction	Description
			• 0 - Indicates that the eCPRI message is the last one inside the eCPRI PDU. • 1 - Indicates that another eCPRI message follows this one within the eCPRI PDU. Valid on SOP assertion and stable until EOP assertion.
sink_msg_type	8	Input	Indicate message type of the eCPRI message. Valid range is 0-7 and 64-255 for eCPRI v1.2 specification. Valid on SOP assertion and stable until EOP assertion.
sink_mem_acc_id	8	Input	Indicate remote memory access id of the eCPRI message type 4: Valid on SOP assertion and stable until EOP assertion.
sink_op_type	8	Input	Indicate operation of the eCPRI message type 4: • 8'b0000 0000 – Read Request • 8'b0000 0001 – Read Response • 8'b0001 0000 – Write Request • 8'b0001 0001 – Write Response • 8'b0010 xxxx – Write No Response • 8'bxxxx 0010 – Failure No Response Valid on SOP assertion and stable until EOP assertion.
sink_element_id	16	Input	Indicates element id of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
sink_address	48	Input	Indicates memory address of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
sink_length	16	Input	Indicates memory access length of the eCPRI message type 4. Valid on SOP assertion and stable until EOP assertion.
sink_reset_id	16	Input	Indicates reset ID of the eCPRI message type 6. Valid on SOP assertion and stable until EOP assertion.
sink_reset_op	8	Input	Indicates reset operation type of the eCPRI message type 6: • 8'b0000 0000 – Remote Reset Request • 8'b0000 0001 – Remote Reset Response • Others – Reserved Valid on SOP assertion and stable until EOP assertion.
sink_event_id	8	Input	Indicate event ID of the eCPRI message type 7. Valid on SOP assertion and stable until EOP assertion.
sink_event_type	8	Input	Indicates event type of the eCPRI message type 7. Valid on SOP assertion and stable until EOP assertion.

6. Registers

The eCPRI IP core registers are 32-bits wide and are accessible to you using the Avalon-MM interface. This table lists the registers available in the IP core. All unlisted locations are reserved.

Table 37. Register Access Codes

Code	Description
RW	Read and write
RO	Read only
RW1C	Read, write, and clear. The user application writes 1 to the register bit(s) to invoke a defined instruction. The IP core clears the bit(s) upon executing instructions.

Table 38. eCPRI IP Core Register Map

Offset	Name
0x0000	eCPRI Version
0x0001	eCPRI Scratch
0x0002	eCPRI Common Control
0x0003	eCPRI Message 5 Control
0x0004	eCPRI TX Error Message
0x0005	eCPRI RX Error Message
0x0006	eCPRI Error Mask Message
0x0007	eCPRI Error Log Message
0x0008	eCPRI Error Message 5 Compensation Value 0
0x0009	eCPRI Error Message 5 Compensation Value 1
0x000A	eCPRI Transport Delay 0
0x000B	eCPRI Transport Delay 1
0x000C	eCPRI Transport Delay 2
0x0010	Ethernet Frame Scratch
0x0011	Source MAC Address 0
0x0012	Source MAC Address 1
0x0013	Destination MAC 0 Address 0
0x0014	Destination MAC 0 Address 1
0x0015	Destination MAC 1 Address 0

continued...

Offset	Name
0x0016	Destination MAC 1 Address 1
0x0017	Destination MAC 2 Address 0
0x0018	Destination MAC 2 Address 1
0x0019	Destination MAC 3 Address 0
0x001A	Destination MAC 3 Address 1
0x001B	Destination MAC 4 Address 0
0x001C	Destination MAC 4 Address 1
0x001D	Destination MAC 5 Address 0
0x001E	Destination MAC 5 Address 1
0x001F	Destination MAC 6 Address 0
0x0020	Destination MAC 6 Address 1
0x0021	Destination MAC 7 Address 0
0x0022	Destination MAC 7 Address 1
0x0023	VLAN Tag 0
0x0024	VLAN Tag 1
0x0025	VLAN Tag 2
0x0026	VLAN Tag 3
0x0027	VLAN Tag 4
0x0028	VLAN Tag 5
0x0029	VLAN Tag 6
0x002A	VLAN Tag 7
0x002B	Ethertype
0x002C	IPv4 Dw0
0x002D	IPv4 Dw1
0x002E	IPv4 Dw2
0x002F	IPv4 Source Address
0x0030	IPv4 Destination Address
0x0031	UDP Dw0
0x0032	UDP Port
0x0033-0x003C	Reserved
0x003D	MAC Packet Type Enable
0x003E	RX Error

6.1. eCPRI Version Register

Table 39. eCPRI Version Register at Offset 0x000

Register	Bit Width	Description	Access	Reset
ecpri_version	[31:0]	eCPRI IP Version: • 4'0001b- Current IP version • Reserved for other values.	RO	4'b0001

6.2. eCPRI Scratch Register

Table 40. eCPRI Scratch Register at Offset 0x0001

Register	Bit Width	Description	Access	Reset
ecpri_scratch	[31:0]	Test data	RW	0x0

6.3. eCPRI Common Control Register

Table 41. eCPRI Common Control Register at Offset 0x0002

Register	Bit Width	Description	Access	Reset
ecpri_common_ctrl	[31:11]	Reserved	RO	0x0
	[10]	Mapping feature enable	RW	0x0
	[9]	Interrupt enable	RW	0x0
	[8]	Error mask enable	RW	0x0
	[7:6]	Indicates fragmentation size. Valid values are: • 2'b01- MTU 1500 Bytes • 2'b10- MTU 9000 Bytes	RW	0x1
	[5]	Fragmentation enable	RO	0x0
	[4]	Message Type 4 buffer mode Enable. Valid values are: • 1'b0- Basic mode • 1'b1- Buffer mode	RW	0x0
	[3:0]	eCPRI protocol version. Valid values are: • 4'0001b- The interpretation of the eCPRI message shall follow eCPRI specification version 1.0 • Other- Reserved for future eCPRI protocol.	RO	4'b0001

6.4. eCPRI Message 5 Control Register

Table 42. eCPRI Message 5 Control Register at Offset 0x0003

Register	Bit Width	Description	Access	Reset
ecpri_msg5_ctrl	[31:20]	Reserved	RO	0x0
	[19:12]	Dummy bytes. Specifies the dummy bytes inserted into message 5. The byte count is in multiple of 8, with maximum value of 184.	RW	0x0
	[11]	One-way measurement completion	RW1C	0x0
	[10]	One-way measurement start	RW	0x0
	[9]	Remote request. Valid values are: • 0: Request action type • 1: Remote request action type	RW	0x1
	[8:1]	Measurement ID	RW	0x0
	[0]	Specifies the operation mode for the one-way measurement process. Ignore if **Delay measurement** parameter is off. Valid values are: • 0: 1-step operation mode • 1: 2-step operation mode	RW	0x0

6.5. eCPRI TX Error Message Register

Table 43. eCPRI TX Error Message Register at Offset 0x0004

Register	Bit Width	Description	Access	Reset
ecpri_tx_err_msg	[31:8]	Reserved	RO	0x0
	[7]	Single message concatenation size	RW1C	0x0
	[6]	Multiple messages concatenation size	RW1C	0x0
	[5]	Timeout no end of concatenation message	RW1C	0x0
	[4]	Invalid eCPRI message type	RW1C	0x0
	[3]	M20K ECC error	RW1C	0x0

continued...

Register	Bit Width	Description	Access	Reset
	[2]	Buffer overflow error	RW1C	0x0
	[1]	Missing EOP	RW1C	0x0
	[0]	Missing SOP	RW1C	0x0

6.6. eCPRI RX Error Message Register

Table 44. eCPRI RX Error Message Register at Offset 0x0005

Register	Bit Width	Description	Access	Reset
ecpri_rx_err_msg	[31:16]	Reserved	RO	0x0
	[15]	Invalid concatenation bit received	RW1C	0x0
	[14]	Payload length/Size mismatch	RW1C	0x0
	[13]	Message 4 buffer mode error. Valid values are: • 1'b0- Now error • 1'b1- Pending request queue overflow	RW1C	0x0
	[12]	Timeout no memory access response	RW1C	0x0
	[11]	Timeout no reset response	RW1C	0x0
	[10]	Timeout no response (message type 5)	RW1C	0x0
	[9]	Reserved	RW1C	0x0
	[8]	Invalid response message received. This bit indicates that the eCPRI message 5 packet parser receives invalid eCPRI action type.	RW1C	0x0
	[7]	Invalid follow up message received. Example, parser expects eCPRI Follow up messages but receives other messages.	RW1C	0x0
	[6]	Invalid request messages received. This bit indicates that the eCPRI message 5 packet parser receives invalid eCPRI action type. Example, parser receives messages other than expects to	RW1C	0x0

continued...

Register	Bit Width	Description	Access	Reset
		receive eCPRI request, remote request ore request with follow up message		
	[5]	Invalid measurement ID received	RW1C	0x0
	[4]	Invalid eCPRI message type When received eCPRI message type 8 to 63.	RW1C	0x0
	[3]	M20K ECC error	RW1C	0x0
	[2]	Buffer overflow error	RW1C	0x0
	[1]	Missing EOP	RW1C	0x0
	[0]	Missing SOP	RW1C	0x0

6.7. eCPRI Error Mask Message Register

Table 45. eCPRI Error Mask Message Register at Offset 0x0006

Register	Bit Width	Description	Access	Reset
ecpri_err_mask_msg	[31:19]	Reserved	RO	0x0
	[18]	Invalid concatenation bit error mask	RW	0x0
	[17]	Payload Length/Size error mask	RW	0x0
	[16]	Message 4 buffer mode error mask	RW	0x0
	[15]	M20K ECC error mask	RW	0x0
	[14]	Buffer overflow error mask	RW	0x0
	[13]	Missing EOP error mask	RW	0x0
	[12]	Missing SOP error mask	RW	0x0
	[11]	Timeout no memory access response error mask	RW	0x0
	[10]	Timeout no reset response error mask	RW	0x0
	[9]	Single message concatenation size error mask	RW	0x0
	[8]	Multiple messages concatenation size error mask	RW	0x0

continued...

Register	Bit Width	Description	Access	Reset
	[7]	Timeout no end of concatenation message error mask	RW	0x0
	[6]	Timeout no response (message type 5) error mask	RW	0x0
	[5]	Out of order eCPRI packets received error mask	RW	0x0
	[4]	Invalid eCPRI message type error mask	RW	0x0
	[3]	Invalid Measurement ID received error mask	RW	0x0
	[2]	Invalid response message received error mask	RW	0x0
	[1]	Invalid follow up message received error mask	RW	0x0
	[0]	Invalid request messages received error mask	RW	0x0

6.8. eCPRI Error Log Message Register

Table 46. eCPRI Error Log Message Register at Offset 0x0007

Register	Bit Width	Description	Access	Reset
ecpri_err_log_msg	[31:30]	Reserved	RO	0x0
	[29:22]	Error Measurement ID	RO	0x0
	[21:14]	Error message 5 action type	RO	0x0
	[13:8]	Error eCPRI message type	RO	0x0
	[7:0]	Error memory access ID	RO	0x0

6.9. eCPRI Error Message 5 Compensation Value 0 Register

Table 47. eCPRI Error Message 5 Compensation Value 0 Register at Offset 0x0008

Register	Bit Width	Description	Access	Reset
ecpri_msg5_cv0	[31:0]	eCPRI message 5 compensation value (LSB)	RW	0x0

6.10. eCPRI Error Message 5 Compensation Value 1 Register

Table 48. eCPRI Error Message 5 Compensation Value 1 Register at Offset 0x0009

Register	Bit Width	Description	Access	Reset
ecpri_msg5_cv1	[31:0]	eCPRI message 5 compensation value (MSB)	RW	0x0

6.11. eCPRI Transport Delay 0 Register

Table 49. eCPRI Transport Delay 0 Register at Offset 0x000A

Register	Bit Width	Description	Access	Reset
ecpri_msg5_td0	[31:0]	eCPRI message 5 transport delay, bytes [3:0]	RW	0x0

6.12. eCPRI Transport Delay 1 Register

Table 50. eCPRI Transport Delay 1 Register at Offset 0x000B

Register	Bit Width	Description	Access	Reset
ecpri_msg5_td1	[31:0]	eCPRI message 5 transport delay, bytes [7:4]	RW	0x0

6.13. eCPRI Transport Delay 2 Register

Table 51. eCPRI Transport Delay 2 Register at Offset 0x000C

Register	Bit Width	Description	Access	Reset
ecpri_msg5_td2	[31:16]	Reserved	RO	0x0
	[15:0]	eCPRI message 5 transport delay, bytes [9:8]	RW	0x0

6.14. Ethernet Frame Scratch Register

Table 52. Ethernet Frame Scratch Register at Offset 0x0010

Register	Bit Width	Description	Access	Reset
eth_frame_scratch	[31:0]	Test data.	RW	0x0

6.15. Source MAC Address <*i*> Register, where i= 0, 1

Table 53. Source MAC Address Register at Offset 0x0011 and 0x0012

Register	Bit Width	Description	Access	Reset
source_mac_addr_<i>	[31:0]	When i=0, it indicates lower four bytes of the source MAC address. Configure this register with a non-zero value before you enable the eCPRI IP for operations. Map the source MAC address as follows: • source_mac_addr_0: Lower four bytes of the source MAC address. Example, If the source MAC address is 00-1C-23-17-4A-CB, set value of this field to 0x23174ACB • source_mac_addr_1, it indicates upper two bytes of the source MAC address. Set the value of this field to 0x0000001C When i=1 , bits [31:16] are reserved.	RW	0x0

6.16. Destination MAC *n* Address <*i*> Register, where n= 0, 1, 2, 3, 4, 5, 6, 7 and i= 0, 1

Table 54. Destination MAC Address Register at Offset 0x0013, 0x0014, 0x0015, 0x0016, 0x0017, 0x0018, 0x0019, 0x001A, 0x001B, 0x001C, 0x001D, 0x001E, 0x001F, 0x0020, 0x0021, 0x0022

Register	Bit Width	Description	Access	Reset
`dest_mac_addr_<i>`	[31:0]	When i=0, it indicates lower four bytes of the destination MAC address. Configure this register with a non-zero value before you enable the eCPRI IP for operations. Map the source MAC address as follows: • `dest_mac_addr_0`: Lower four bytes of the destination MAC address. Example, If the source MAC address is 00-1C-23-17-4A-CB, set value of this field to 0x23174ACB • `dest_mac_addr_1`, it indicates upper two bytes of the destination MAC address. Set the value of this field to 0x0000001C When i=1 , bits [31:16] are reserved.	RW	0x0

6.17. VLAN Tag Register <*i*>, where i= 0, 1, 2, 3, 4, 5, 6, 7

Table 55. VLAN Tag Register at Offset 0x0023, 0x0024, 0x0025, 0x0026, 0x0027, 0x0028, 0x0029, 0x002A

Register	Bit Width	Description	Access	Reset
`mac_vlan_tag_<i>`	[31:16]	VLAN tag for eCPRI message type 0 TPID (Tag Protocol Identifier)	RO	0x81
	[15:13]	Priority Code Point (PCP)	RW	0x0
	[12]	Drop Eligible Indicator (DEI)	RO	0x0
	[11:0]	VLAN Identifier (VID)	RW	0x0

6.18. Ethertype Register

Table 56. Ethertype Register at Offset 0x002B

Register	Bit Width	Description	Access	Reset
`mac_ethertype`	[31:0]	eCPRI Ethertype	RO	0xAEFE

6.19. IPv4 Dw0 Register

Table 57. IPv4 Dw0 Register at Offset 0x002C

Register	Bit Width	Description	Access	Reset
Ipv4_dw0_address	[31:28]	Version	RO	0x4
	[27:24]	Header length	RO	0x5
	[23:16]	Type of service/ DiffServ	RW	0x0
	[15:0]	Reserved	RO	0x0

6.20. IPv4 Dw1 Register

Table 58. IPv4 Dw1 Register at Offset 0x002D

Register	Bit Width	Description	Access	Reset
Ipv4_dw1_address	[31:16]	Identifier	RW	0x4
	[15:12]	Flag	RW	0x5
	[11:0]	Fragment offset	RW	0x0

6.21. IPv4 Dw2 Register

Table 59. IPv4 Dw2 Register at Offset 0x002E

Register	Bit Width	Description	Access	Reset
mudp_dw0_address	[31:0]	Reserved	RO	0x0

6.22. IPv4 Source Address Register

Table 60. IPv4 Source Address Register at Offset 0x002F

Register	Bit Width	Description	Access	Reset
ipv4_src_address_0	[31:0]	IPv4 source address	RW	0x0

6.23. IPv4 Destination Address Register

Table 61. IPv4 Destination Address Register at Offset 0x0030

Register	Bit Width	Description	Access	Reset
ipv4_dst_address_0	[31:0]	IPv4 destination address	RW	0x0

6.24. UDP Dw0 Register

Table 62. UDP Dw0 Register at Offset 0x0031

Register	Bit Width	Description	Access	Reset
mudp_dw0_address	[31:0]	Reserved	RO	0x0

6.25. UDP Port Register

Table 63. UDP Port Register at Offset 0x0033

Register	Bit Width	Description	Access	Reset
mudp_address_0	[31:16]	UDP source port	RW	0x0
	[15:0]	UDP destination port	RW	0x0

6.26. MAC Packet Type Enable Register

Table 64. MAC Packet Type Enable Register at Offset 0x003D

Register	Bit Width	Description	Access	Reset
packet_type_enable	[31:5]	Reserved	RO	0x0
	[4]	UDP enable	RW	0x0
	[3]	Stack VLAN enable	RW	0x0
	[2]	VLAN enable	RW	0x0
	[1]	IPv4 enable	RW	0x0
	[0]	Reserved	RW	0x0

6.27. RX Error Register

Table 65. RX Error Register at Offset 0x003E

Register	Bit Width	Description	Access	Reset
rx_error_address	[31:2]	Reserved	RO	0x0
	[1]	UDP checksum error	RO	0x0
	[0]	IPv4 checksum error	RO	0x0

7. Document Revision History for the eCPRI Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.04.15	19.4	1.0.0	Corrected information in *Table: eCPRI Version Register at Offset 0x000*.
2020.04.13	19.4	1.0.0	Initial release.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for FPGA - Field Programmable Gate Array ***category:***

Click to view products by Intel ***manufacturer:***

Other Similar products are found below :

5AGXMA3D4F27I3N EP1S10F672C7 EP1S20F672I7N EP1SGX25DF672I6N EP2S30F672C5 EP2S60F672C5N PLUS16L87N PLUS16R67N PLUS20L87N PLUS20R87N 5AGXBA5D6F27C6N 5AGXMA5D6F27C6N EP1S20F672C7N EP1SGX25DF672I6 EP20K300EFI672-2X EP2S15F672C3N QP82S100/BXA LCMXO640C-3FT256CAHW 5962-9154507MXA EPF10K200SFI672-2B 5962-9154503MQA EPF10K50EQC240-1N EP1S10F672C7N ICE40UL1K-SWG16ITR1K ICE5LP4K-SG48ITR ICE5LP4K-SWG36ITR1K ICE40UL640-SWG16ITR1K ICE40LP1K-SWG16TR1K ICE40LP640-SWG16TR1K GW1NR-LV9MG100PC6/I5 GW1N-LV4CS72C6/I5 GW1NR-UV4MG81PC6/I5 GW1N-LV9UG332C6/I5 GW1NR-LV4MG81PC6/I5 GW1NZ-LV1CS16C6/I5 GW1NR-LV9QN88PC6/I5 GW1N-LV4MG160C6/I5 GW1NR-UV4QN88C6/I5 GW1N-UV4MG160C6/I5 GW1NR-UV9QN88PC6/I5 GW1NR-LV9LQ144PC6/I5 GW1N-UV9MG160C6/I5 EP2AGX45CU17I5G M2GL050-FGG896I M2GL050-FGG484 ICE40UL1K-CM36AITR1K LFE2M-20E-5F256C LFE2M20SE-5F256C LFE2M50E-5F484C LFXP15C-3FN388C