CPRI v6.0 IP Core User Guide

Last updated for Quartus Prime Design Suite: 17.0 IR3, 17.0, 17.0 Update 1, 17.0 Update 2
UG-20008
2019.01.02

Subscribe
Send Feedback

101 Innovation Drive
San Jose, CA 95134
www.altera.com

ALTERA® now part of Intel

Contents

About the CPRI v6.0 IP Core

1

2019.01.02

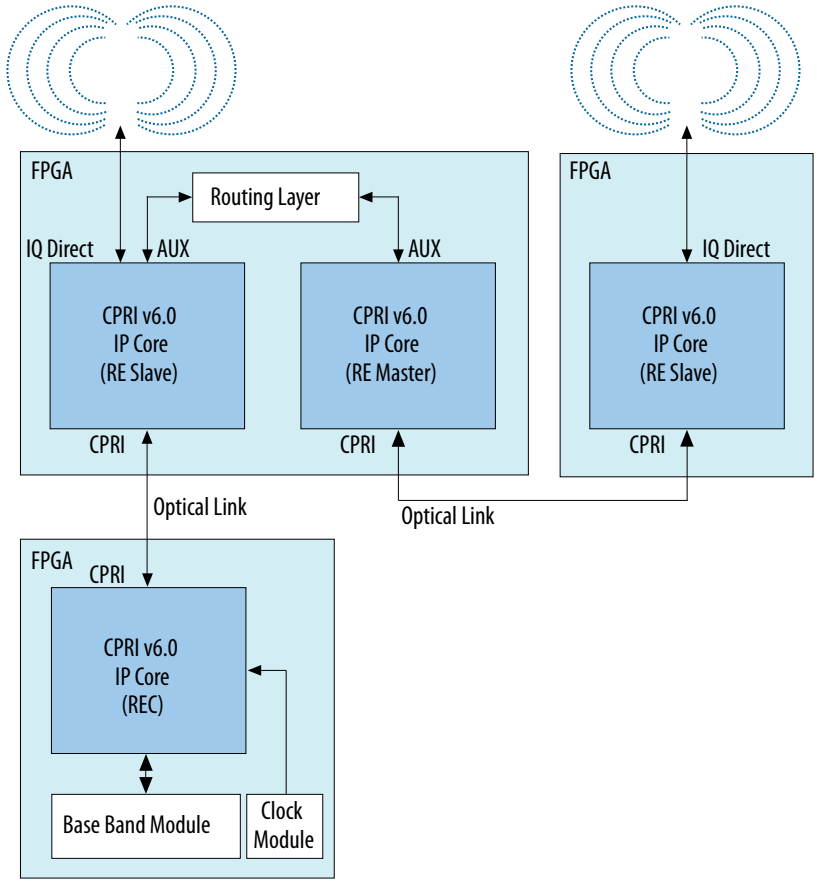
UG-20008 Subscribe Send Feedback

The Common Public Radio Interface (CPRI) v6.0 Intel® FPGA IP core implements the *CPRI Specification V6.0 (2013-08-30)*. CPRI is a high-speed serial interface for network radio equipment controllers (REC) to receive data from and provide data to remote radio equipment (RE).

The CPRI v6.0 IP core targets high-performance, remote, radio network applications. You can configure the CPRI v6.0 IP core as an RE or an REC.

Figure 1-1: Typical CPRI Application on Intel FPGA Devices

Example system implementation with a two-hop daisy chain. Optical links between devices support high performance.



CPRI v6.0 IP Core Supported Features

The CPRI v6.0 IP core offers the following features:

- Compliant with the *Common Public Radio Interface (CPRI) Specification V6.0 (2013-08-30) Interface Specification* available on the CPRI Industry Initiative website (www.cpri.info).
- Supports radio equipment controller (REC) and radio equipment (RE) module configurations.
- Supports the following CPRI link features:
  - Configurable CPRI communication line bit rate (to 0.6144, 1.2288, 2.4576, 3.0720, 4.9152, 6.144, 8.11008, 9.8304, or 10.1376 Gbps) using Intel FPGA on-chip high-speed transceivers.
  - CPRI line bit rate auto-rate negotiation support.
  - Configurable and run-time programmable synchronization mode: master port or slave port on a CPRI link.
  - Scrambling and descrambling at 8.11008 and 10.1376 Gbps.
  - Optional scrambling and descrambling at 4.9152, 6.1440, and 9.8304 Gbps.
  - Transmitter (Tx) and receiver (Rx) delay measurement and calibration.
    - Optional support for single-trip delay calibration.
    - Optional round-trip delay calibration.
  - L1 link status and alarm (Z.130.0) control and status monitoring.
  - Access to all Vendor Specific data.
  - Diagnostic parallel reverse loopback paths.
  - Diagnostic serial and parallel forward loopback paths.
  - Diagnostic stand-alone slave testing mode.
- Includes the following interfaces:
  - Register access interface to external or on-chip processor, using the Intel Avalon® Memory-Mapped (Avalon-MM) interconnect specification.
  - Optional auxiliary (AUX) interface for full access to raw CPRI frame. Provides direct access to full radioframe, synchronizes the frame position with timing references, and enables routing application support from slave to master ports to implement daisy-chain topologies.
  - Optional choice of IEEE 802.3 100BASE-X compliant 10/100 Mbps MII or 1000BASE-X compliant 1Gbps GMII for Ethernet frame access.
  - Optional direct I/Q access interface enables integration of all user-defined air standard I/Q mapping schemes.
    - Optional external I/Q mapper and demapper modules with reference design support.
    - Optional external I/Q compression and decompression modules with reference design support.
  - Optional vendor specific data access interfaces provide direct access to Vendor Specific (VS), Control AxC (Ctrl_AxC), and Real-time Vendor Specific (RTVS) subchannels.
  - Optional HDLC serial interface provides direct access to slow control and management subchannels.
  - Optional L1 inband interface provides direct access to Z.130.0 link status and alarm control word.

Related Information

- **CPRI Industry Initiative website**
  For a detailed specification of the CPRI protocol refer to the *CPRI Specification V6.0 (2013-08-30) Interface Specification* available on the CPRI Industry Initiative website.
- **Altera wiki CPRI v6.0 IP core information**
  Includes links to the I/Q mapper and other CPRI v6.0 IP core reference designs.
- **Intel FPGA Design Store**
  Includes CPRI v6.0 reference designs.

CPRI v6.0 IP Core Device Family and Speed Grade Support

The following sections list the device family and device speed grade support offered by the CPRI v6.0 IP core:

Device Family Support

Table 1-1: 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	Intel has verified the IP core 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	Intel has verified the IP core 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 1-2: CPRI v6.0 IP Core Device Family Support

Shows the level of support offered by the CPRI v6.0 IP core for each Intel FPGA device family.

Device Family	Support
Intel Stratix® 10	Advance Intel Stratix 10 device support is available only in the Intel Quartus® Prime Pro Edition software v17.0 IR3. It is not available in software releases 17.0, 17.0 Update 1, and 17.0 Update 2. The software release v17.0 IR3 supports only L-tile devices.
Intel Arria® 10	Default support level provided in the Intel Quartus Prime software. Refer to the *Quartus Prime Standard Edition Software and Device Support Release Notes* and the *Quartus Prime Pro Edition Software and Device Support Release Notes* for the relevant software release.
Arria V (GX and GT)	Default support level provided in Intel Quartus Prime Standard Edition software. Refer to the *Quartus Prime Standard Edition Software and Device Support Release Notes* for the relevant software release.
Arria V GZ	Default support level provided in Intel Quartus Prime Standard Edition software. Refer to the *Quartus Prime Standard Edition Software and Device Support Release Notes* for the relevant software release.
Cyclone V (GX and GT)	Default support level provided in Intel Quartus Prime Standard Edition software. Refer to the *Quartus Prime Standard Edition Software and Device Support Release Notes* for the relevant software release.
Stratix V (GX and GT)	Default support level provided in Intel Quartus Prime Standard Edition software. Refer to the *Quartus Prime Standard Edition Software and Device Support Release Notes* for the relevant software release.
Other device families	No support

Related Information

- **CPRI v6.0 IP Core Performance: Device and Transceiver Speed Grade Support** on page 1-5

- **Timing and Power Models**
  Reports the default device support levels in the current version of the Intel Quartus Prime Standard Edition software.
- **Timing and Power Models**
  Reports the default device support levels in the current version of the Intel Quartus Prime Pro Edition software.

CPRI v6.0 IP Core Performance: Device and Transceiver Speed Grade Support

Table 1-3: Slowest Supported Device Speed Grade and Supported Transceiver Speed Grade

Lower device speed grade numbers correspond to faster devices. The entry -x indicates that both the industrial speed grade Ix and the commercial speed grade Cx are supported for this device family and CPRI line bit rate. Table entries show slowest supported device speed grade / supported transceiver speed grade.

Device Family	CPRI Line Bit Rate (Gbps)								
	0.6144	1.2288	2.4576	3.072	4.9152	6.1440	8.11008	9.8304	10.1376
Intel Stratix 10	(1)	-2 / -3							
Intel Arria 10	(1)	-3 / -4							
Stratix V GT	-3 / H3						-2 / H2		
Stratix V GX	-4 / H3						-2 / H2		
Arria V GZ	-4 / H3						-3 / H2		(1)
Arria V GX	-6 / H6				-5 / H4	-5 / H4	(1)		
Arria V GT	-5/H3						(1)		
Cyclone V GT	-7 / H5					(1)			
Cyclone V GX	-8 / H7	-7 / H6			(1)				

(1) The CPRI v6.0 IP core does not support this CPRI line bit rate for this device family.

IP Core Verification

To ensure functional correctness of the CPRI v6.0 IP core, Intel performs extensive validation through both simulation and hardware testing. Before releasing a version of the CPRI v6.0 IP core, Intel runs comprehensive regression tests in the associated version of the Intel Quartus Prime software.

Related Information

- **Knowledge Base Errata for CPRI v6.0 IP core**
  Some exceptions to functional correctness are documented in the CPRI v6.0 IP core errata.
- **Altera wiki CPRI v6.0 Errata page**
  Other exceptions to functional correctness are documented on the Altera wiki CPRI v6.0 Errata page.

Resource Utilization for CPRI v6.0 IP Cores

Resource utilization changes depending on the parameter settings you specify in the CPRI v6.0 parameter editor. For example, with every additional interface you enable, the IP core requires additional resources to implement the module that supports that interface.

The resource utilization numbers are approximate as the Intel Quartus Prime Fitter assigns resources based on the entirety of your design. The numbers below result from a single run on a simple design. Your results may vary.

Table 1-4: Minimum and Maximum IP Core Variations for Resource Utilization Reporting

The IP core FPGA resource utilization table reports resource utilization for a minimum IP core variation and a maximum IP core variation. Parameters not specified remain at their default values, or their values do not affect resource utilization.

Parameter	Minimum Variation	Maximum Variation
Line bit rate	1.2288 Gbps for target device in the Intel Arria 10 and Intel Stratix 10 device families, 0.6144 Gbps for all other device families	Maximum bit rate (device family dependent)
Synchronization mode	Master	Master
Operation mode	TX/RX Duplex	TX/RX Duplex
Core clock source input	Internal	Internal
Receiver soft buffer depth	4	8
Auxiliary and direct interfaces write latency cycle(s)	—	9
Enable interface, for all optional direct interfaces in the L1 Features tab	Off	On

Parameter	Minimum Variation	Maximum Variation
Ethernet PCS interface	NONE	GMII
L2 Ethernet PCS Tx/Rx FIFO depth	—	11
Enable single-trip delay calibration	Off	Off
Enable round-trip delay calibration	Off	On
Round-trip delay calibration FIFO depth	—	4

Table 1-5: IP Core FPGA Resource Utilization

Lists the resources and expected performance for minimum and maximum variations of the CPRI v6.0 IP core in each supported device family.

These results were obtained using the Intel Quartus Prime v17.0 IR3 software on an Intel Stratix 10 device, and using the Intel Quartus Prime v17.0 software for all other target device families.

- The numbers of ALMs and logic registers are rounded up to the nearest 100.
- The numbers of ALMs, before rounding, are the **ALMs needed** numbers from the Intel Quartus Prime Fitter Report.

Intel Stratix 10 Device (with L-Tile Transceivers)	ALMs	Logic Registers	M20K Blocks
Minimum (1.2288 Gbps CPRI line bit rate)	1100	1600	2
Maximum (10.1376 Gbps CPRI line bit rate)	4000	5100	23
Intel Arria 10 Device	**ALMs**	**Logic Registers**	**M20K Blocks**
Minimum (1.2288 Gbps CPRI line bit rate)	700	1400	2
Maximum (10.1376 Gbps CPRI line bit rate)	4000	5000	24
Arria V GX or GT Device	**ALMs**	**Logic Registers**	**M10K Blocks**
Minimum (0.6144 Gbps CPRI line bit rate)	700	1300	3
Maximum (6.144 Gbps CPRI line bit rate)	3300	4800	39
Arria V GZ Device	**ALMs**	**Logic Registers**	**M20K Blocks**
Minimum (0.6144 Gbps CPRI line bit rate)	700	1400	2
Maximum (9.8304 Gbps CPRI line bit rate)	3700	5000	21

Cyclone V GX or GT Device	ALMs	Logic Registers	M10K Blocks
Minimum (0.6144 Gbps CPRI line bit rate)	700	1400	2
Maximum (4.9512 Gbps CPRI line bit rate)	4100	5800	27
Stratix V GX or GT Device	**ALMs**	**Logic Registers**	**M20K Blocks**
Minimum (0.6144 Gbps CPRI line bit rate)	700	1400	2
Maximum (10.1376 Gbps CPRI line bit rate)	3700	5500	20

Related Information

Fitter Resources Reports in the Intel Quartus Prime Help
Information about Intel Quartus Prime resource utilization reporting, including **ALMs needed**.

Release Information

Table 1-6: CPRI v6.0 IP Core Current Release Information

Item	Description
Compatible Intel Quartus Prime Software Version	17.0 IR3, 17.0, 17.0 Update 1, and 17.0 Update 2
Release Date	2017.08.07
Ordering Codes	IP-CPRI-V6

Getting Started with the CPRI v6.0 IP Core

2

2019.01.02

UG-20008

Subscribe Send Feedback

Explains how to install, parameterize, and simulate the CPRI v6.0 IP core.

Related Information

Introduction to Intel FPGA IP IP Cores
Additional information about generating an Intel FPGA IP core and integrating it in your Intel Quartus Prime project.

Installation and Licensing

The CPRI v6.0 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 Altera Self-Service Licensing Center (SSLC). Refer to Related Information below for the correct link for this IP core.

Figure 2-1: IP Core Installation Directory Structure

Directory structure after you install the CPRI v6.0 IP core.

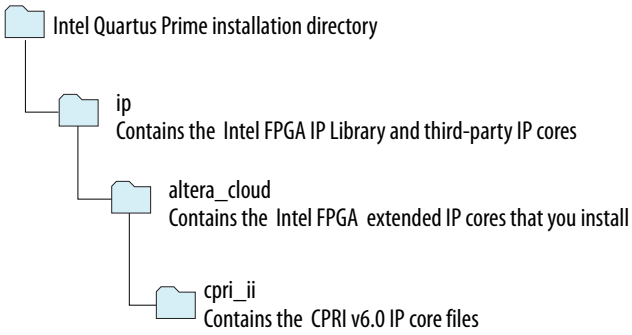


Table 2-1: IP Core Installation Locations

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

Related Information

- **Intel FPGA website**

- **Self-Service Licensing Center (SSLC)**
  After you purchase the CPRI v6.0 IP core, the IP core is available for download from the SSLC page in your myAltera account. You must create a myAltera 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.

Generating CPRI v6.0 IP Cores

You can quickly configure a custom IP variation in the parameter editor. Use the following steps to specify CPRI v6.0 IP core options and parameters in the parameter editor.

1. In the IP Catalog (**Tools** > **IP Catalog**), locate and double-click the name **CPRI v6.0**. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>`.qsys` (in Intel Quartus Prime Standard Edition) or <*your_ip*>`.ip` (in Intel Quartus Prime Pro Edition). Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor, including one or more of the following. Refer to "IP Core Parameters" for information about specific IP core parameters.
   - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
   - Specify options for processing the IP core files in other EDA tools.
4. Click **Generate HDL**. The **Generation** dialog box appears.
5. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
6. To generate a simulation testbench, click **Generate Example Design**. Please refer to the instructions in the *Running the Testbench* section.
7. To generate an HDL instantiation template that you can copy and paste into your text editor, click **Generate** > **Show Instantiation Template**.
8. Click **Finish**. The parameter editor adds the top-level `.qsys` or `.ip` file to the current project automatically. If you are prompted to manually add the `.qsys` file to the project, click **Project** > **Add/Remove Files in Project** to add the file.
9. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

CPRI v6.0 IP Core File Structure

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

Figure 2-2: IP Core Generated Files

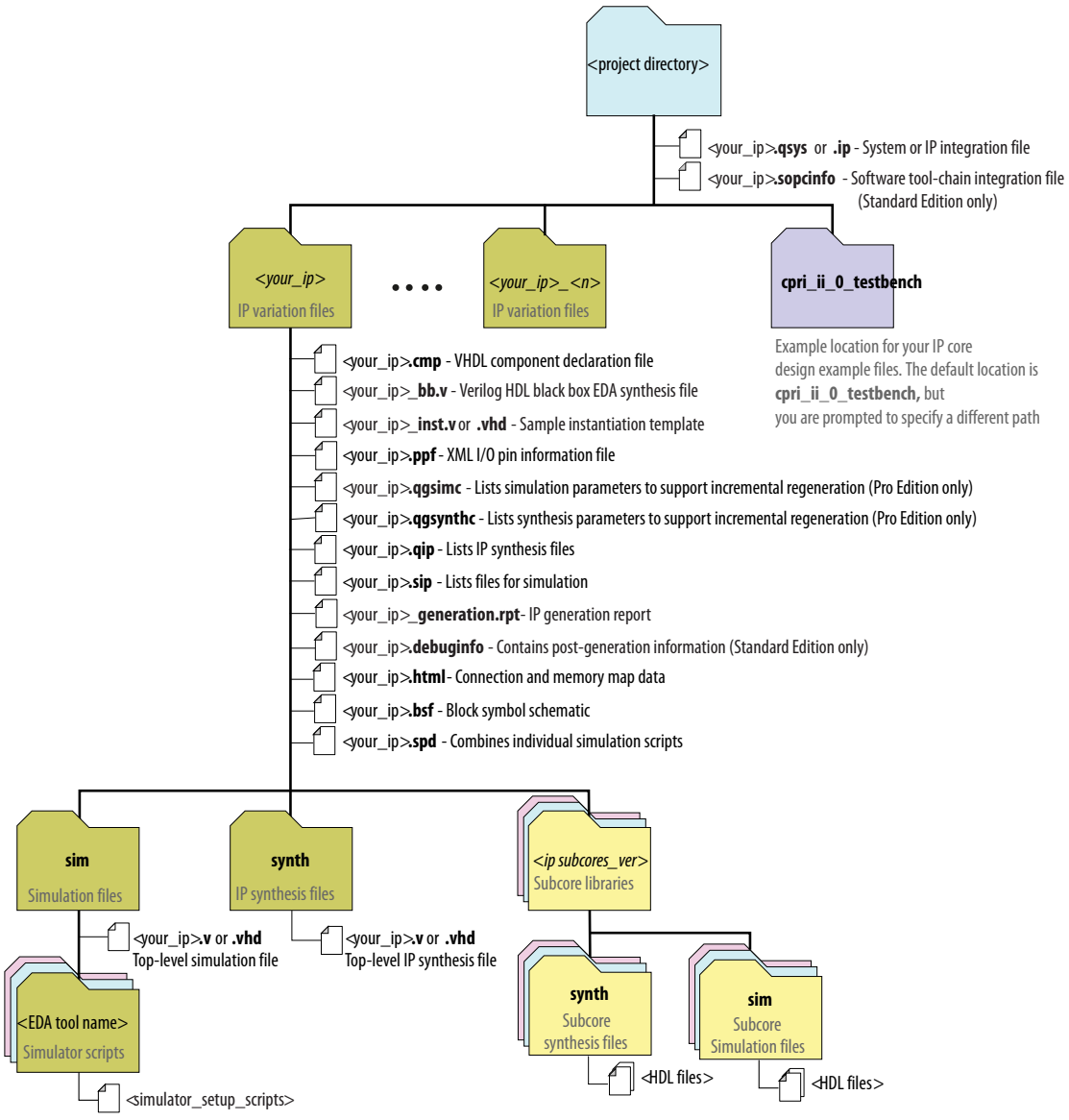


Table 2-2: IP Core Generated Files

File Name	Description
<*your_ip*>.qsys (Intel Quartus Prime Standard Edition only)	The Qsys system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.ip (Intel Quartus Prime Pro Edition only)	

File Name	Description
`<system>.sopcinfo`	Describes the connections and IP component parameterizations in your Qsys system. You can parse its contents to get requirements when you develop software drivers for IP components. (Intel Quartus Prime Standard Edition only) Downstream tools such as the Intel 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>.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 Qsys generation log file. A summary of the messages during IP generation.
`<your_ip>.debuginfo`	Contains post-generation information. Used to pass System Console and Bus Analyzer Toolkit information about the Qsys interconnect. The Bus Analysis Toolkit uses this file to identify debug components in the Qsys interconnect. (Intel Quartus Prime Standard Edition only)
`<your_ip>.qgsimc`	Lists simulation parameters to support incremental regeneration. (Intel Quartus Prime Pro Edition only)
`<your_ip>.qgsynthc`	Lists synthesis parameters to support incremental regeneration. (Intel Quartus Prime Pro Edition only)
`<your_ip>.qip`	Contains all the required information about the IP component to integrate and compile the IP component in the Quartus Prime software.
`<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 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.

File Name	Description
<*your_ip*>.sip	Contains information required for NativeLink simulation of IP components. You must add the **.sip** file to your Intel Quartus Prime project.
<*your_ip*>_inst.v and _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*>.regmap	If IP contains register information, **.regmap** file generates. The **.regmap** file describes the register map information of master and slave interfaces. This file complements the **.sopcinfo** file by providing more detailed register information about the system. This enables register display views and user customizable statistics in the System Console.
<*your_ip*>.svd	Allows hard processor system (HPS) System Debug tools to view the register maps of peripherals connected to HPS within a Qsys system. During synthesis, the **.svd** files for slave interfaces visible to System Console masters are stored in the **.sof** file in the debug section. System Console reads this section, which Qsys can query for register map information. For system slaves, Qsys can access the registers by name.
<*your_ip*>.v and <*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.
aldec/	Contains a Riviera-PRO script rivierapro_setup.tcl to setup 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.
submodules/	Contains HDL files for the IP core submodule.
<*child IP cores*>/	For each generated child IP core directory, Qsys generates synth/ andsim/ sub-directories.

CPRI v6.0 IP Core Parameters

The CPRI v6.0 parameter editor provides the parameters you can set to configure the CPRI v6.0 IP core and simulation testbench.

The CPRI v6.0 parameter editor has three tabs.

Table 2-3: General CPRI v6.0 IP Core Parameters

Describes the general parameters for customizing the CPRI v6.0 IP core. These parameters appear on the **General** tab in the CPRI v6.0 parameter editor.

Parameter	Options	Default Setting	Parameter Description
Line bit rate (Mbits/s)	• 614.4 • 1228.8 • 2457.6 • 3072.0 • 4915.2 • 6144.0 • 8110.08 • 9830.4 • 10137.6	Lowest bit rate supported for the device family	Selects the CPRI line bit rate. Refer to **CPRI v6.0 IP Core Performance: Device and Transceiver Speed Grade Support** on page 1-5 for supported CPRI line bit rates in the supported device families. The parameter editor does not allow you to specify a CPRI line bit rate that the target device does not support.
Synchronization mode	• Master • Slave	Master	Specifies whether the CPRI v6.0 IP core is configured as a CPRI link master or a CPRI link slave. The value of this parameter determines the initial and reset clocking mode of the CPRI v6.0 IP core. You can modify the IP core clocking mode dynamically by modifying the value of the `synchronization_mode` field of the `L1_CONFIG` register.
Operation mode	• TX/RX Duplex • TX Simplex • RX Simplex	TX/RX Duplex	Specifies whether the CPRI v6.0 IP core is configured with RX functionality only (RX Simplex), with TX functionality only (TX Simplex), or with both RX and TX functionality (TX/TX Duplex). If you specify a simplex mode, the Quartus Prime Fitter synthesizes logic for only one direction of traffic. If the CPRI v6.0 IP core is in TX simplex operation mode, it can transmit on the CPRI link but cannot receive. If the CPRI v6.0 IP core is in RX simplex operation mode, it can receive traffic on the CPRI link but cannot transmit. Currently, IP core variations that target an Intel Stratix 10 device support only the **TX/RX Duplex** mode.

Parameter	Options	Default Setting	Parameter Description
Core clock source input	• External • Internal	Internal	Specifies the clock source of the `cpri_coreclk`. In the internal clocking scheme, you should drive the `cpri_coreclk` with the cleaned-up `xcvr_recovered_clk` in CPRI slave IP cores, and with the external master clock in CPRI master IP cores. In this clocking scheme, the IP core uses the `cpri_coreclk` input clock only when the IP core is running at the CPRI line bit rate of 8.11008 or 10.1376 Gbps, The external clocking scheme supports the single-trip delay calibration feature. In this clocking scheme, the IP core uses this clock at all CPRI line bit rates. You can drive the `cpri_coreclk` input clock with the `tx_clkout` output clock from the TX PCS.
Transmitter local clock division factor	• 1 • 2 • 3 • 4	1	Specifies the division factor for the local clock divider. The IP core divides the high speed clock from the transceiver TX PLL (`xcvr_ext_pll_clk`) to generate the serial TX clock. This feature supports the configuration of multiple instances of the CPRI v6.0 IP core that run at different CPRI line bit rates but share use of the same TX PLL. This parameter is not available if you set the value of **Operation mode** to **RX Simplex**. IP core variations that target an Intel Arria 10 device or an Intel Stratix 10 device, with **Line bit rate** set to 4915.2 Mbps or slower, support only the value of 1.

Parameter	Options	Default Setting	Parameter Description
Number of receiver CDR reference clock(s)	• 1 • 2	1	Specifies the width of the receiver reference clock that controls the receiver. The CPRI v6.0 IP core supports the selection of one or two clocks. This option supports auto-negotiation to and from the CPRI line bit rate of 10.1376 Gbps in CPRI v6.0 IP core variations that target a Stratix V device. Refer to "IP Core Clocking Structure." If you set this parameter to the value of **1**, the `xcvr_cdr_refclk` is a single clock. If you set this parameter to the value of **2**, the `xcvr_cdr_refclk` input signal is two bits wide, to support two distinct reference clocks. Intel recommends that you specify a two-bit clock for Stratix V variations that are expected to implement auto-negotiation up to a 10.1376 Gbps CPRI line bit rate. In this case the typical design drives one bit of the `xcvr_cdr_refclk` clock with a common 307.2 MHz clock for the lower CPRI line bit rates and drives the other bit with a 253.44 MHz clock for the 10.1376 Gbps CPRI line bit rate. However, these specific clock frequencies are not required. If the value of this parameter is 2, the receiver clocks the CDR with the `xcvr_cdr_refclk[0]` input signal by default. You can switch the receiver to use `xcvr_cdr_refclk[1]`, or back to `xcvr_cdr_refclk[0]`, by dynamically reconfiguring the RX transceiver. IP core variations that target a device family other than the Stratix V device family, support only a single-bit receiver reference clock. This parameter is not available if you set the value of **Operation mode** to **TX Simplex**.
Receiver CDR reference clock frequency (MHz)	Per drop-down menu	307.2	Specifies the incoming reference clock frequency for the receiver CDR PLL, in MHz. You must drive the input clock `xcvr_cdr_refclk` or `xcvr_cdr_refclk[0]` at the frequency you specify for this parameter. This parameter is not available if you set the value of **Operation mode** to **TX Simplex**.

Parameter	Options	Default Setting	Parameter Description
VCCR_GXB and VCCT_GXB supply voltage for the transceiver	• 1_1V • 1_0V	1_0V	Specifies whether the transceiver supply voltage is 1.1 V or 1.0 V. The supply voltage must match the voltage you specify for this parameter, in IP core variations that target an Intel Stratix 10 device. This parameter affects only IP core variations that target an Intel Stratix 10 device. You can ignore it for other device families.
Recovered clock source	• PCS • PMA	PCS	Specifies the clock source of the `xcvr_recovered_clk`. Intel recommends that you set this parameter to the value of **PMA** in IP core variations that target a Stratix V device, if you expect your IP core to auto-negotiate to or from the CPRI line bit rate of 10.1376 Gbps. In this case, sourcing the recovered clock from the PMA improves jitter on that clock. If you specify the PCS source, the IP core switches between two PCS-internal clocks at auto-negotiation to or from the CPRI line bit rate of 10.1376 Gbps. This parameter is not available for • CPRI master IP cores • IP cores that target an Intel Stratix 10 device • IP cores for which you set the value of **Operation mode** to **TX Simplex**
Receiver soft buffer depth	4, 5, 6, 7, or 8	6	The value you specify for this parameter is $\log_2$ of the IP core Layer 1 Rx buffer depth.The IP core supports a maximum Layer 1 RX buffer depth of 256. The default depth of the buffer is 64, specified by the parameter default value of 6. For most systems, the default buffer depth is adequate to handle dispersion, jitter, and drift that can occur on the link while the system is running. However, the parameter is available for cases in which additional depth is required. Increasing the value of this parameter increases resource utilization. Increasing the value of this parameter affects latency only when the buffer fills beyond the default capacity. In that case, the larger buffer increases latency but prevents data loss. The user guide refers to this parameter value as `RX_BUF_DEPTH`. This parameter is not available if you set the value of **Operation mode** to **TX Simplex**.

Parameter	Options	Default Setting	Parameter Description
Enable line bit rate auto-negotiation	• On • Off	Off	Turn on the **Enable line bit rate auto-negotiation** parameter to specify that your CPRI v6.0 IP core supports auto-rate negotiation. If you turn on this parameter, your IP core does not implement auto-negotiation. You must dynamically reconfigure the transceiver to modify the CPRI line bit rate and implement auto-negotiation. However, if you turn off this parameter, the IP core does not support bit line rate auto-negotiation, and you cannot modify the CPRI line bit rate dynamically. If you turn off this parameter and also turn off **Enable start-up sequence state machine**, **Enable single-trip delay calibration**, and in Intel Arria 10 devices, the **Enable ADME, transceiver capability, control and status registers access**, the transceiver reconfiguration interface is not available. This parameter is available when you specify a CPRI line bit rate (value for the **Line bit rate** parameter) that is greater than 614.4 Mbps. This parameter is not yet available for Intel Stratix 10 devices.
Enable line bit rate auto-negotiation down to 614.4 Mbps	• On • Off	Off	Turn on this parameter to specify that your auto-rate negotiation enabled CPRI v6.0 IP core can support auto-rate negotiation all the way down to the CPRI line bit rate of 0.6144 Gbps. This parameter is available for devices that support a CPRI line bit rate of 614.4 Mbps, and when you turn on **Enable line bit rate auto-negotiation**.

Table 2-4: CPRI v6.0 IP Core Interface Feature Parameters

Describes the parameters for customizing the CPRI v6.0 IP core Layer 1 and Layer 2 interfaces and testing features. These parameters appear on the **Interfaces** tab in the CPRI v6.0 parameter editor.

Parameter	Options	Default Setting	Parameter Description
L1 Features			
Management (CSR) interface standard	Currently, only the Avalon-MM CPU interface is available in the CPRI v6.0 IP core.		Selects the interface specification that describes the behavior of the CPRI v6.0 IP core register access interface.

Parameter	Options	Default Setting	Parameter Description
Avalon-MM interface addressing type	• Word • Byte	Word	Specifies the addressing mode for the Avalon-MM CPU interface.If the addressing mode is **Word**, you must ensure you correctly align the connections between Avalon-MM components. This parameter specifies how other components must connect to the `cpu_address` bus on the CPU interface.
Auxiliary and direct interfaces write latency cycle(s)	0 to 9	0	Specifies the additional write latency on the AUX TX interface and other direct TX interfaces to the CPRI v6.0 IP core. The write latency is the number of `cpri_clkout` cycles from when the `aux_tx_seq` output signal has the value of 0 to when user logic writes data to the AUX TX interface. For other direct interfaces, the IP core notifies user logic when it is ready for input and the user does not need to monitor the `aux_tx_seq` signal. When **Auxiliary and direct interfaces write latency cycle(s)** has the value of zero, the write latency on the direct TX interfaces is one `cpri_clkout` cycle. When **Auxiliary and direct interfaces write latency cycle(s)** has the value of N, the write latency is (1+N) `cpri_clkout` cycles. Set this parameter to a value that provides user logic with sufficient advance notice of the position in the CPRI frame. The processing time that user logic requires after determining the current position in the CPRI frame is implementation specific. This parameter is available if you turn on at least one direct interface in your CPRI v6.0 IP core variation.
Enable auxiliary interface	• On • Off	Off	Turn on this parameter to include the AUX interface in your CPRI v6.0 IP core. The AUX interface provides full access to the raw CPRI frame.

Parameter	Options	Default Setting	Parameter Description
Enable all control word access via management interface	• On • Off	Off	Turn on this parameter to enable access to all control words in a hyperframe using the CPRI v6.0 `CTRL_INDEX`, `TX_CTRL`, and `RX_CTRL` registers. Use this option with caution. During transmission, this feature has higher priority than the MII, the GMII, the HDLC serial interface, the L1 control and status interface, and the generation of special symbols (K28.5, D16.2, /S/, /T/) , and can overwrite standard control words in the hyperframe.
Enable direct Z.130.0 alarm bits access interface	• On • Off	Off	Turn on this parameter to include a dedicated L1 control and status interface to communicate the contents of the CPRI frame Z.130.0 word, which includes alarms and reset signals.
Enable direct ctrl_axc access interface	• On • Off	Off	Turn on this parameter to include a dedicated interface to access the Ctrl_AxC subchannels in the CPRI frame.
Enable direct vendor specific access interface	• On • Off	Off	Turn on this parameter to include a dedicated interface to access the VS subchannels in the CPRI frame.
Enable direct real-time vendor specific interface	• On • Off	Off	Turn on this parameter to include a dedicated interface to access the RTVS subchannel in the CPRI frame. This parameter is available when you specify a CPRI line bit rate of 10137.6 Mbps.
Enable start-up sequence state machine	• On • Off	Off	Turn on this parameter to include a start-up sequence state machine in the CPRI v6.0 IP core. If you turn off this parameter and also turn off **Enable line bit rate auto-negotiation**, **Enable single-trip delay calibration**, and in Intel Arria 10 and Intel Stratix 10 devices, the **Enable ADME, transceiver capability, control and status registers access**, the transceiver reconfiguration interface is not available. This parameter is available if you set the value of **Operation mode** to **TX/RX Duplex**.

Parameter	Options	Default Setting	Parameter Description
Enable protocol version and C&M channel setting auto-negotiation	• On • Off	Off	Turn on this parameter to include a negotiator block that performs auto-negotiation of L1 inband protocol version (communicated in CPRI frame position Z.2.0) and L2 C&M rates (communicated in CPRI frame positions Z.66.0 and Z.194.0). This parameter is available when you turn on **Enable start-up sequence state machine**.
Enable direct I/Q mapping interface	• On • Off	Off	Turn on this parameter to include a dedicated interface to access the raw I/Q data bytes in the CPRI frame.
L2 Features			
Enable HDLC serial interface	• On • Off	Off	Turn on this parameter to include a dedicated interface to communicate the contents of the slow C&M subchannels. For full HDLC communication, you must connect a user-defined HDLC module to this interface.
Ethernet PCS interface	• NONE • MII • GMII	NONE	Specify whether to include an MII or GMII port to communicate with the fast C&M (Ethernet) CPRI subchannel. You can also specify that the IP core does not support either interface. • An MII port complies with the IEEE 802.3 100BASE-X 100Mbps MII specification, • A GMII port complies with the IEEE 802.3 1000BASE-X 1Gbps GMII specification. For full Ethernet communication, you must connect a user-defined Ethernet MAC to this interface.
L2 Ethernet PCS Tx/Rx buffer depth	7, 8, 9, 10, 11	7	The value you specify for this parameter is $\log_2$ of the IP core Layer 2 Ethernet PCS Rx buffer depth and Tx buffer depth. The IP core supports a maximum Layer 2 Ethernet PCS buffer depth of 1024 for MII and 2048 for GMII. This parameter is available when you include an MII or GMII port to communicate with the fast C&M (Ethernet) CPRI subchannel by selecting the value of **MII** or **GMII** for the **Ethernet PCS interface** parameter. The new value of 11 is supported only for GMII.
Debug Features			

Parameter	Options	Default Setting	Parameter Description
Enable L1 debug interfaces	• On • Off	Off	Turn on this parameter to include dedicated transceiver status and L1 Rx status interfaces to support debug. This parameter is not available if you set the value of **Operation mode** to **TX Simplex**.
Enable ADME, transceiver capability, control and status registers access	• On • Off	Off	Turn on this parameter to support debugging through the System Console and to expose transceiver registers. If you turn off this parameter and also turn off **Enable line bit rate auto-negotiation**, **Enable start-up sequence state machine**, and **Enable single-trip delay calibration**, the Intel Arria 10 or Intel Stratix 10 transceiver reconfiguration interface is not available. This parameter is available only for Intel Arria 10 and Intel Stratix 10 devices.
Enable transceiver PMA serial forward loopback path	• On • Off	Off	Turn on this parameter to enable transceiver PMA serial forward loopback. To turn on transceiver PMA serial forward loopback (Tx to Rx), you must also write the value of 2'b01 to the `loop_forward` field of the `LOOPBACK` register at offset 0x44. This parameter is not available if you set the value of **Operation mode** to **TX Simplex** or to **RX Simplex**.
Enable parallel forward loopback paths	• On • Off	Off	Turn on this parameter to enable other internal parallel forward loopback paths (Tx to Rx). To turn on internal parallel forward loopback, you must also write a non-zero value to the `loop_forward` field of the `LOOPBACK` register at offset 0x44. This parameter is not available if you set the value of **Operation mode** to **TX Simplex** or to **RX Simplex**.

Parameter	Options	Default Setting	Parameter Description
Enable parallel reversed loopback paths	• On • Off	Off	Turn on this parameter to enable internal parallel reverse loopback (Rx to Tx). To turn on reverse loopback, you must also write a non-zero value to the `loop_reversed` field of the `LOOPBACK` register at offset 0x44, to specify the parts of the CPRI frame that are sent on the loopback path. This parameter is not available if you set the value of **Operation mode** to **TX Simplex** or to **RX Simplex**.

Table 2-5: CPRI v6.0 IP Core Advanced Feature Parameters

Describes the parameters for customizing the CPRI v6.0 IP core delay calibration features. These parameters appear on the **Advanced** tab in the CPRI v6.0 parameter editor.

Parameter	Options	Default Setting	Parameter Description
Enable single-trip delay calibration	• On • Off	Off	Turn on this parameter to specify that your CPRI v6.0 IP core supports single-trip delay calibration. If you turn on this parameter, your IP core implements single-trip delay calibration only if you connect it according to **Adding and Connecting the Single-Trip Delay Calibration Blocks** on page 2-23. Intel provides the required external blocks but you must connect them to the IP core in your design. This parameter is only available in IP core variations that target an Intel Arria 10 device. If you turn off this parameter and also turn off **Enable line bit rate auto-negotiation**, **Enable start-up sequence state machine** the transceiver reconfiguration interface is not available. This parameter is available only if you set the value of the **Core clock source input** parameter to **External**.
Enable round-trip delay calibration	• On • Off	Off	Turn on this parameter to to specify that your CPRI v6.0 IP core supports round-trip delay calibration. This parameter is available only if you set the value of the **Synchronization mode** parameter to **Master**.

Parameter	Options	Default Setting	Parameter Description
Round-trip delay calibration FIFO depth	• 2 • 3 • 4	2	The value you specify for this parameter is $\log_2$ of the IP core RTD calibration buffer depth.The IP core supports a maximum RTD calibration buffer depth of 16. The default depth of the buffer is 4, specified by the parameter default value of 2. For buffer depth *N*, the Read pointer can move (*N*/2)-1 entries in either direction from its initial state.

Related Information

- **LOOPBACK Register** on page 5-15
- **CPRI v6.0 IP Core Clocking Structure** on page 3-3

Integrating Your IP Core in Your Design: Required External Blocks

You must connect your CPRI v6.0 IP core to some additional required design components. Your design can simulate and compile without some of these connections and logical blocks, but it will not function correctly in hardware unless all of them are present and connected in your design.

The CPRI v6.0 IP core requires that you define, instantiate, and connect the following additional software and hardware modules for all CPRI v6.0 IP core variations:

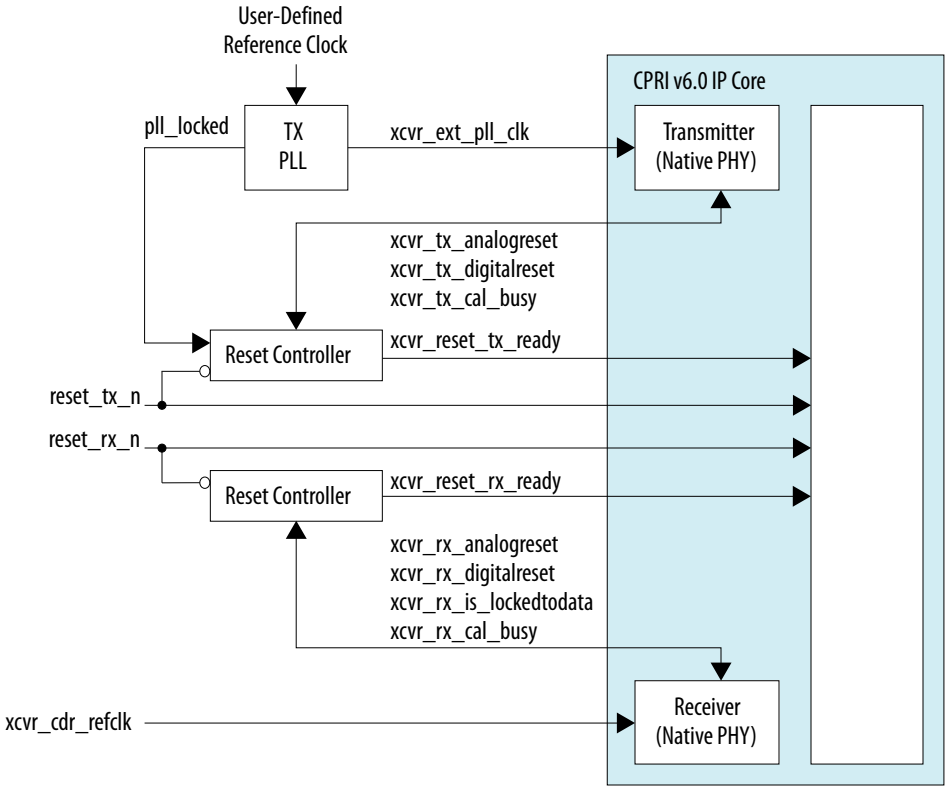
- An external PLL IP core to configure the transceiver TX PLL. Although the FPGA elements this IP core configures are physically part of the device transceiver, you must instantiate this IP core in software separately from the CPRI v6.0 IP core. In Intel Arria 10 and Intel Stratix 10 devices, this requirement supports the configuration of multiple Intel FPGA IP cores using the same transceiver block in the device.
- One or more external reset controllers to coordinate the reset sequence for the CPRI v6.0 IP core in your design.

In addition, some IP core variations require additional modules to function correctly in hardware.

- CPRI link slave modules require an off-chip clean-up PLL.
- Variations that target a 28-nm device (Arria V, Arria V GZ, Cyclone V, or Stratix V device family) require an external transceiver reconfiguration controller.
- Variations with the single-trip delay calibration feature require additional blocks that Intel provides but does not connect in your design.

Figure 2-3: Required External Blocks

An example showing how you could connect required components to a single CPRI v6.0 IP core that targets an Arria 10 device.



Related Information

Adding the Reset Controller on page 2-20

Adding the Transceiver TX PLL IP Core

The CPRI v6.0 IP core requires that you generate and connect a TX transceiver PLL IP core. The transceiver PLL IP core configures the TX PLL in the transceiver on the device, but you must generate the transceiver PLL IP core separately from the CPRI v6.0 IP core in software. If you do not generate and connect the transceiver PLL IP core, the CPRI v6.0 IP core does not compile.

You can use the IP Catalog to generate the external PLL IP core that configures a TX PLL on the device. In the IP Catalog, select an Intel FPGA IP core that configures an appropriate PLL on your target device.

For your 28-nm design, you can select **Altera PLL (FPLL)** or **Transceiver PLL v17.0** in the IP Catalog. In the parameter editor for the TX PLL IP core you select, you must set the PLL output frequency to the expected input frequency for the CPRI v6.0 IP core `xcvr_ext_pll_clk` input signal.

For your Intel Arria 10 design, you can select **Arria 10 Transceiver ATX PLL**, **Arria 10 Transceiver CMU PLL**, or **Arria 10 FPLL** in the IP Catalog. For your Intel Stratix 10 design, you can select one of the

Transceiver PLL options available for your device tile type in the IP Catalog. In the parameter editor for the TX PLL IP core you select, you must set the following parameter values:

- **PLL output frequency** to one half the per-lane data rate of the IP core variation, multiplied by the value of the **Transmitter local clock division factor** parameter of the CPRI v6.0 IP core. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting drives the transceiver with the correct clock for the CPRI v6.0 IP core line bit rate.

  For example, if your CPRI v6.0 IP core variation has a CPRI line bit rate of 10.1376 Gbps and you set the **Transmitter local clock division factor** parameter to the value of 1, you must set the TX PLL **PLL output frequency** parameter to the value of 5068.8 MHz.
- **PLL reference clock frequency** to a frequency at which you can drive the TX PLL input reference clock. You must drive the external PLL reference clock input signal at the frequency you specify for this parameter.

  For example, if your CPRI v6.0 IP core variation has a CPRI line bit rate of 10.1376 Gbps and you set the **Transmitter local clock division factor** parameter to the value of 1, you can set the **PLL reference clock frequency** to the value of 307.2 MHz.

Intel Arria 10 and Intel Stratix 10 devices and the Intel Quartus Prime software support multiple options for configuring an Intel Arria 10 TX PLL or Intel Stratix 10 TX PLL. Depending on the TX PLL IP core you select and the configuration options you prefer, you have a wide range of choices in parameterizing the external TX PLL for a design that targets one of these two device families.

You must connect the external TX PLL signals and the CPRI v6.0 IP core transceiver TX PLL interface signals according to the following rules.

Table 2-6: Required Connections Between Transceiver TX PLL and CPRI v6.0 IP Core

Connect the `xcvr_ext_pll_clk` input signal of the CPRI v6.0 IP core to the `pll_clkout`, `tx_serial_clk`, or `outclk0` output signal of the external PLL IP core. Information about connecting the transceiver TX PLL to the Reset Controller is available in **Adding the Reset Controller** on page 2-20.

CPRI v6.0 IP Core Signal	TX PLL Signal
`xcvr_ext_pll_clk` (input)	Intel Arria 10 or Intel Stratix 10 CMU or ATX PLL: `tx_serial_clk`
	Intel Arria 10 FPLL: `outclk0`
	28-nm device CMU or ATX PLL: `pll_clkout`
	28-nm device FPLL: `outclk0`

If your CPRI v6.0 IP core is an RE slave, drive the input signal of the external PLL IP core with the output of the off-chip cleanup PLL.

User logic must provide the connections. Refer to the demonstration testbench for example working user logic including one correct method to instantiate and connect the external PLL to a single CPRI v6.0 IP core.

Related Information

- **Interface to the External PLL** on page 3-74

- **Altera Transceiver PHY IP Core User Guide**
  Information about how to configure an external PLL for your Arria V, Arria V GZ, Cyclone V, or Stratix V design.
- **Intel Arria 10 Transceiver PHY User Guide**
  Information about how to configure an external PLL for your own Intel Arria 10 design.
- **Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide**
  Information about how to configure an external PLL for your Intel Stratix 10 design that targets an Intel Stratix 10 ES1 device.
- **Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide**
  Information about how to configure an external PLL for your Intel Stratix 10 design that targets a production device.

Adding the Reset Controller

The CPRI v6.0 IP core requires that you provide reset control logic to handle the required reset sequence for the IP core transceiver on the device. For a duplex CPRI v6.0 IP core, you must generate and connect two Transceiver PHY Reset Controller IP cores to perform this function, one reset controller for the TX transceiver and one reset controller for the RX transceiver in the CPRI v6.0 IP core. If you do not implement the device-specific correct reset sequence, the IP core does not function correctly in hardware.

You can use the IP Catalog to generate Transceiver PHY Reset Controller IP cores for the device family that your CPRI v6.0 IP core targets.

Follow the instructions in the *Altera Transceiver PHY IP Core User Guide*, the *Intel Arria 10 Transceiver PHY User Guide*, or the appropriate Stratix 10 transceiver PHY user guide. The CPRI v6.0 IP core configures the Native PHY IP core for the target device family. You must configure the reset controllers to coordinate reset of the CPRI v6.0 IP core including the Native PHY IP core, and the transceiver PLL IP core. In the case of Arria V, Arria V GZ, Cyclone V, and Stratix V variations, the reset controllers must also coordinate with the transceiver reconfiguration controller.

To configure a TX reset controller, in the Transceiver PHY Reset Controller parameter editor, you must set the following parameter values:

- Set **Input clock frequency** to a value in the range of 100–150 MHz. You must drive the CPRI v6.0 IP core `reconfig_clk` at the same frequency you specify for this parameter.
- Turn on **Synchronize reset input**.
- Turn on **Use fast reset for simulation**.
- Turn on **Enable TX PLL reset control**.
- Set **pll_powerdown duration** to the value of 10.
- Turn on **Enable TX channel reset control**.
- Leave all other parameters turned off or for the parameters that do not turn on or off, at their default values.

To configure an RX reset controller, in the Transceiver PHY Reset Controller parameter editor, you must set the following parameter values:

- Set **Input clock frequency** to a value in the range of 100–150 MHz. You must drive the CPRI v6.0 IP core `reconfig_clk` at the same frequency you specify for this parameter.
- Turn on **Synchronize reset input**.
- Turn on **Use fast reset for simulation**.
- Turn on **Enable RX channel reset control**.
- Leave all other parameters turned off or for the parameters that do not turn on or off, at their default values.

You must connect the external reset controller signals to the CPRI v6.0 IP core reset controller interface signals and transceiver TX PLL signals according to the following rules. Refer to **Integrating Your IP Core in Your Design: Required External Blocks** on page 2-17 for an illustration of the connections.

Table 2-7: Required Connections to and From Reset Controllers in CPRI v6.0 Design

Lists the required connections between the reset controllers and the CPRI v6.0 IP core and the transceiver TX PLL IP core. For information about connecting the transceiver TX PLL to the CPRI v6.0 IP core, refer to **Adding the Transceiver TX PLL IP Core** on page 2-18.

Transmit-Side Reset Controller Signal	Connect to
`clock` (input)	Clock source for CPRI v6.0 IP core `reconfig_clk` input signal
`reset` (input)	Source of CPRI v6.0 IP core `reset_tx_n` input signal, inverted
`pll_powerdown` (output)	TX PLL `pll_powerdown`
`pll_locked` (input)	TX PLL `pll_locked`
`tx_analogreset` (output)	CPRI v6.0 IP core `xcvr_tx_analogreset`
`tx_digitalreset` (output)	CPRI v6.0 IP core `xcvr_tx_digitalreset`
`tx_cal_busy` (input)	CPRI v6.0 IP core `xcvr_tx_cal_busy`
`tx_ready` (output)	CPRI v6.0 IP core `xcvr_reset_tx_ready`
Receive-Side Reset Controller Signal	**Connect to**
`clock` (input)	Clock source for CPRI v6.0 IP core `reconfig_clk` input signal
`reset` (input)	Source of CPRI v6.0 IP core `reset_rx_n` input signal, inverted
`rx_is_lockedtodata` (input)	CPRI v6.0 IP core `xcvr_rx_is_lockedtodata`
`rx_analogreset` (output)	CPRI v6.0 IP core `xcvr_rx_analogreset`
`rx_digitalreset` (output)	CPRI v6.0 IP core `xcvr_rx_digitalreset`
`rx_cal_busy` (input)	CPRI v6.0 IP core `xcvr_rx_cal_busy`
`rx_ready` (output)	CPRI v6.0 IP core `xcvr_reset_rx_ready`

User logic must provide the connections. Refer to the demonstration testbench for example working user logic including one correct method to instantiate and connect the external reset controllers.

Related Information

- **Interface to the External Reset Controller** on page 3-73
- **Integrating Your IP Core in Your Design: Required External Blocks** on page 2-17
  Figure illustrates the required connections.
- **Altera Transceiver PHY IP Core User Guide**
  Information about how to configure and connect the Altera Transceiver PHY Reset Controller for your Arria V, Arria V GZ, Cyclone V, or Stratix V design.
- **Intel Arria 10 Transceiver PHY User Guide**
  Information about how to configure and connect the Transceiver PHY Reset Controller for your Intel Arria 10 design.
- **Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide**
  Information about how to configure and connect the Intel FPGA Transceiver PHY Reset Controller for your Stratix 10 design that targets a Intel Stratix 10 ES1 device.
- **Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide**
  Information about how to configure and connect the Intel FPGA Transceiver PHY Reset Controller for your Intel Stratix 10 design that targets a production device.

Adding the Transceiver Reconfiguration Controller

CPRI v6.0 IP cores that target Arria V, Arria V GZ, Cyclone V, and Stratix V devices require an external reconfiguration controller to compile and to function correctly in hardware. CPRI v6.0 IP cores that target Intel Arria 10 or Intel Stratix 10 devices include a transceiver reconfiguration controller block and do not require an external reconfiguration controller.

You can use the IP Catalog to generate the Altera Transceiver Reconfiguration Controller IP core.

When you configure the Altera Transceiver Reconfiguration Controller, you must specify the number of reconfiguration interfaces. The number of reconfiguration interfaces required for the CPRI v6.0 IP core depends on the CPRI v6.0 IP core configuration and your design.For example, you can configure your reconfiguration controller with additional interfaces if your design connects with multiple transceiver IP cores. You can leave other options at the default settings or modify them for your preference. Refer to the *Altera Transceiver PHY User Guide*.

You should connect the `reconfig_to_xcvr` and `reconfig_from_xcvr` ports of the CPRI v6.0 IP core to the corresponding ports of the reconfiguration controller.

You must drive the CPRI v6.0 IP core `reconfig_clk` input port and the Altera Transceiver Reconfiguration Controller `mgmt_clk_clk` input port from the same clock source. Drive both ports at a clock frequency in the range of 100–150MHz.

Related Information

- **Arria V, Arria V GZ, Cyclone V, and Stratix V Transceiver Reconfiguration Interface** on page 3-70
- **Altera Transceiver PHY IP Core User Guide**
  For more information about the Altera Transceiver Reconfiguration Controller.

Adding the Off-Chip Clean-Up PLL

If your CPRI v6.0 IP core is an RE slave, you must connect it to an off-chip clean-up PLL to clean up any jitter that occurs in the CDR output clock, before sending it to the reference clock input of the external TX PLL.

The clean-up PLL performs the clock synchronization necessary to address the CPRI v6.0 Specification requirements R-17, R-18, and R-18A, which address jitter and frequency accuracy in the RE core clock for radio transmisstion.

Drive the clean-up PLL with the CPRI v6.0 IP core `xcvr_recovered_clk` output clock, and connect the cleaned up output to the external TX PLL input reference clock port. In the internal clocking mode, in 8.11008 Gbps and 10.1376 Gbps IP core variations, you should connect the cleaned up output to the `cpri_coreclk` input clock port as well. In the external clocking mode, you can optionally connect the cleaned up output to the `cpri_coreclk` input clock port.

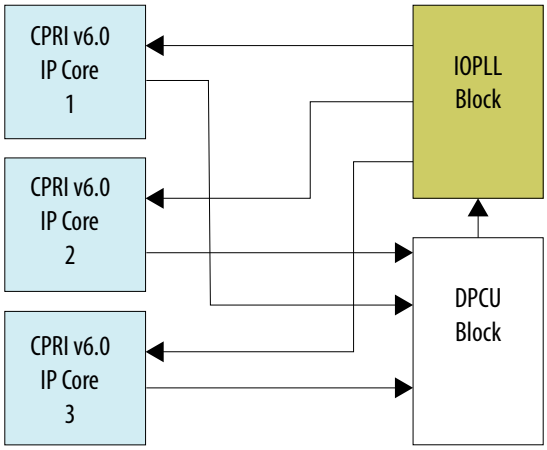
Related Information

CPRI v6.0 IP Core Clocking Structure on page 3-3

Adding and Connecting the Single-Trip Delay Calibration Blocks

If you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor, and **Synchronization mode** is set to **Slave**, the CPRI v6.0 IP core requires that you connect the IOPLL (`pll_core` block) and the Dynamic Phase Control Unit (DPCU) (`pll_dpcu` block). You must generate a device core PLL to create the IOPLL. Intel provides the DPCU block with the CPRI v6.0 IP core, but you must connect it, and the IOPLL, in your design. A single IOPLL block and a single DPCU block can connect to multiple CPRI v6.0 IP cores.

Figure 2-4: Connecting a Single DPCU Block and Single IOPLL Block to Multiple CPRI v6.0 IP Cores



You must generate a non-transceiver PLL IP core to create the IOPLL block. You can use the IP Catalog to generate the external PLL IP core that configures a core PLL on the device. In the IP Catalog, select an Intel FPGA IP core that configures an appropriate PLL on your target device.

For your Intel Arria 10 design, you must select **Arria 10 FPLL** in the IP Catalog. In the parameter editor for the fPLL IP core, you must set the following parameter values:

- Set **fPLL Mode** to **Core**.
- Set the PLL output frequency to the expected input frequency for the CPRI v6.0 IP core `cpri_coreclk` input signal.
- Turn on **Enable access to dynamic phase shift ports**.

You must connect the IOPLL and DPCU signals to the CPRI v6.0 IP core signals according to the following rules. Refer to **Example CPRI v6.0 Clock Connections in Different Clocking Modes** on page 3-8 for an illustration of the clock connections.

Table 2-8: Required Connections to and From IOPLL Block in CPRI v6.0 Design With Single-Trip Delay Calibration

IOPLL Block Signal	Connect to
`cntsel` (input)	DPCU `pll_cntsel` output signal
`num_phase_shifts` (input)	DPCU `pll_num_phase_shifts` output signal
`outclk_0` (output)	CPRI v6.0 IP core `cpri_coreclk` input clock signal
`phase_done` (output)	DPCU `pll_phase_done` input signal
`phase_en` (input)	DPCU `pll_phase_en` output signal
`refclk` (input)	CPRI v6.0 IP core `tx_clkout` output signal or output from the off-chip clean-up PLL
`rst` (input)	Drive with the inverse of the CPRI v6.0 IP core `reset_n` input signal
`scanclk` (input)	DPCU `pll_scanclk` output signal
`updn` (input)	DPCU `pll_updn` output signal

Table 2-9: Required Connections to and From DPCU Block in CPRI v6.0 Design With Single-Trip Delay Calibration

DPCU Block Signal	Connect to
`clk`	Drive in the frequency range of 100–150 MHz. **Note:** Intel recommends that you drive this DPCU input clock with the source for the `reconfig_clk`.
`reset_n` (input)	CPRI v6.0 IP core `reset_n` input signal

DPCU Block Signal	Connect to
csr_bit_rate (input)	If you turned on **Enable line bit rate auto-negotiation** in the CPRI v6.0 parameter editor, connect to the CPRI v6.0 IP core nego_bitrate_out[4:0] output signal. If autorate negotiation is not turned on, hardwire the DPCU csr_bit_rate signal to the encoded value for the CPRI line bit rate: • 5'b00001: 0.6144 Gbps • 5'b00010: 1.2288 Gbps • 5'b00100: 2.4576 Gbps • 5'b00101: 3.0720 Gbps • 5'b01000: 4.9150 Gbps • 5'b01010: 6.1440 Gbps • 5'b01100: 8.11008 Gbps • 5'b10000: 9.8304 Gbps • 5'b10100: 10.1376 Gbps
cal_status (output)	CPRI v6.0 IP core cal_status input signal
cal_ctrl (input)	CPRI v6.0 IP core cal_ctrl output signal
pll_cntsel (output)	IOPLL cntsel input signal
pll_num_phase_shifts (output)	IOPLL num_phase_shifts input signal
pll_phase_done (input)	IOPLL phase_done output signal
pll_phase_en (output)	IOPLL phase_en input signal
pll_scanclk (output)	IOPLL scanclk input signal
pll_updn (output)	IOPLL updn input signal

User logic must provide the connections.

Simulating Intel FPGA IP Cores

The Intel Quartus Prime software supports RTL- and gate-level design simulation of Intel FPGA IP cores in supported EDA simulators. Simulation involves setting up your simulator working environment, compiling simulation model libraries, and running your simulation.

You can use the functional simulation model and the testbench or example design generated with your IP core for simulation. The IP generation output also includes scripts to compile and run any testbench. For a complete list of models or libraries required to simulate your IP core, refer to the scripts generated with the testbench.

Related Information

- **Simulating Intel FPGA IP Cores**

- **Simulating Altera FPGA Designs chapter in Quartus Prime Standard Edition Handbook, Volume 3: Verification**
- **Simulating Intel FPGA Designs chapter in Intel Quartus Prime Pro Edition Handbook, Volume 3: Verification**

Understanding the Testbench

Intel provides a demonstration testbench with the CPRI v6.0 IP core.

If you click **Generate Example Design** in the CPRI v6.0 parameter editor, the Quartus Prime software generates the demonstration testbench. The parameter editor prompts you for the desired location of the testbench.

The testbench is static and does not necessarily match your IP core variation; you can generate it without generating an IP core. The testbench scripts generate a DUT that matches the testbench, but you must manually set the appropriate values for the DUT in the parameter editor before you create the demonstration testbench.

The testbench performs the following sequence of actions with the static DUT:

1. Enables transmission on the CPRI link by setting the `tx_enable` bit (bit [0]) of the CPRI v6.0 IP core `L1_CONFIG` register at offset 0x8 (and resetting all other fields of the register)>
2. Configures the DUT at the highest possible HDLC bit rate for the CPRI line bit rate, by setting the `tx_slow_cm_rate` field of the CPRI v6.0 `CM_CONFIG` register at offset 0x1C to the appropriate value.
3. Reads the `CM_CONFIG` register to confirm settings.
4. After the DUT and the testbench achieve frame synchronization, executes the following transactions:
   a. Performs several write transactions to the AUX Tx interface and confirms the testbench receives them on the CPRI link.
   b. Performs several write transactions to the VS interface and confirms the testbench receives them from the DUT on the CPRI link.
   c. Performs several write trasactions to the RTVS interface for the 10G variant, and confirms the testbench receives them form the DUT on the CPRI link.
   d. Performs several write transactions to the Ctrl_AxC interface and confirms the testbench receives them from the DUT on the CPRI link.
   e. Performs several HDLC transactions for 1G and 2G and confirms the testbench receives them from the DUT on the CPRI link.
   f. Performs several write transactions to the MI or GMI interface and confirms the testbench receives them from the DUT on the CPRI link.
   g. Calculates the round-trip delay through the IP core.

Running the Testbench

To run the CPRI v6.0 IP core demonstration testbench, follow these steps.

1. In the Quartus Prime software IP Catalog, select the CPRI v6.0 IP core and click **Add**.
2. When prompted, you can specify any output file type (HDL). This setting is relevant only for synthesis and does not impact simulation of the demonstration testbench.
3. In the CPRI v6.0 parameter editor, set the following parameter values:

Table 2-10: CPRI v6.0 IP Core Variation for Demonstration Testbench

The testbench scripts require that you set these values in the CPRI v6.0 parameter editor before you click **Generate Example Design**. The scripts generate the DUT but they require that you provide the parameter values.

Parameter	Value
Line bit rate (Mbit/s)	Any value the device family supports.
Synchronization mode	Master
Operation mode	TX/RX Duplex or RX Simplex
Transmitter local clock division factor	1
Number of receiver CDR reference clock(s)	1
Receiver CDR reference clock frequency (MHz)	253.44 if the Line bit rate is 8.11008 Gbps and the IP core targets the Intel Arria 10 or Intel Stratix 10 device family 253.44 if the Line bit rate is 8.11008 or 10.1376 Gbps and the IP core targets a 28-nm device family 307.2 for all other cases
Core clock source input	Internal
Recovered clock source	**PMA** if the Line bit rate is 10.1376 Gbps and IP core targets the Stratix V device family; **PCS** otherwise
Receiver soft buffer depth (value shown is $\log_2$ of actual depth)	6
Enable line bit rate auto-negotiation	Turn off
Enable line bit rate auto-negotiation down to 614.4 Mbps	Not available
Management (CSR) interface standard	AvalonMM
Avalon-MM interface addressing type	Word
Auxiliary and direct interfaces write latency cycle(s)	0
Enable auxiliary interface	**Turn on**
Enable all control word access via management interface	Turn off
Enable direct Z.130.0 alarm bits access interface	Turn off
Enable direct ctrl_axc access interface	**Turn on**
Enable direct vendor specific access interface	**Turn on**
Enable direct real-time vendor specific interface	Not available
Enable start-up sequence state machine	Turn off

Parameter	Value
Enable protocol version and C&M channel setting auto-negotiation	Not available
Enable direct IQ mapping interface	Turn off
Enable HDLC serial interface	**Turn on**
Ethernet PCS interface	**MII** or **GMII**
L2 Ethernet PCS Tx/Rx FIFO depth (value shown is log2 of actual depth)	8
Enable L1 debug interfaces	Turn off
Enable ADME, transceiver capability, control and status registers access	Turn off
Enable transceiver PMA serial forward loopback path	Turn off
Enable parallel forward loopback paths	Turn off
Enable parallel reversed loopback paths	Turn off
Enable single-trip delay calibration	Not available
Enable round-trip delay calibration	Turn off
Round-trip delay calibration FIFO depth	Not available

4. In the CPRI v6.0 parameter editor, click the **Generate Example Design** button and specify the desired location of the testbench.
5. After you generate the demonstration testbench, in the Quartus Prime software, click **View > Utility Windows > Tcl Console**.
6. In the Tcl Console, change directory to your specified testbench directory's `ip_sim` subdirectory.
7. Type `source gen_sim_verilog.tcl` or `source gen_sim_vhdl.tcl`, depending on the language of the model you wish to simulate. Running this script generates the DUT and testbench files.
8. If you are using a simulator that requires that you open a user interface, open your target simulator.

   Note: You must select a simulator that is supported by the Intel Quartus Prime software version you are using.
9. Change directory to your specified testbench directory's `testbench/`<*simulator vendor*> subdirectory.
10. Execute the simulation script in the directory.
    - In the Mentor Graphics ModelSim simulator, type `do run_altera_cpri_v6_tb.tcl`
    - In the Synopsys VCS-MX simulator, type `sh run_altera_cpri_v6_vcsmx_tb.sh`
    - In the Cadence NCSIM simulator, type `sh run_altera_cpri_v6_tb.sh`

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 FPGA with the Programmer and verify the design in hardware.

Note: The IP core directory includes a Synopsys Constraint (`.sdc`) file at *<IP core instance directory>*`/altera_cpri_ii_instance_`*<Intel Quartus Prime release>*`/synth/altera_cpri.sdc`.

Related Information

- **Incremental Compilation for Hierarchical and Team-Based Design**
- **Programming Intel Devices**

Functional Description 3

2019.01.02

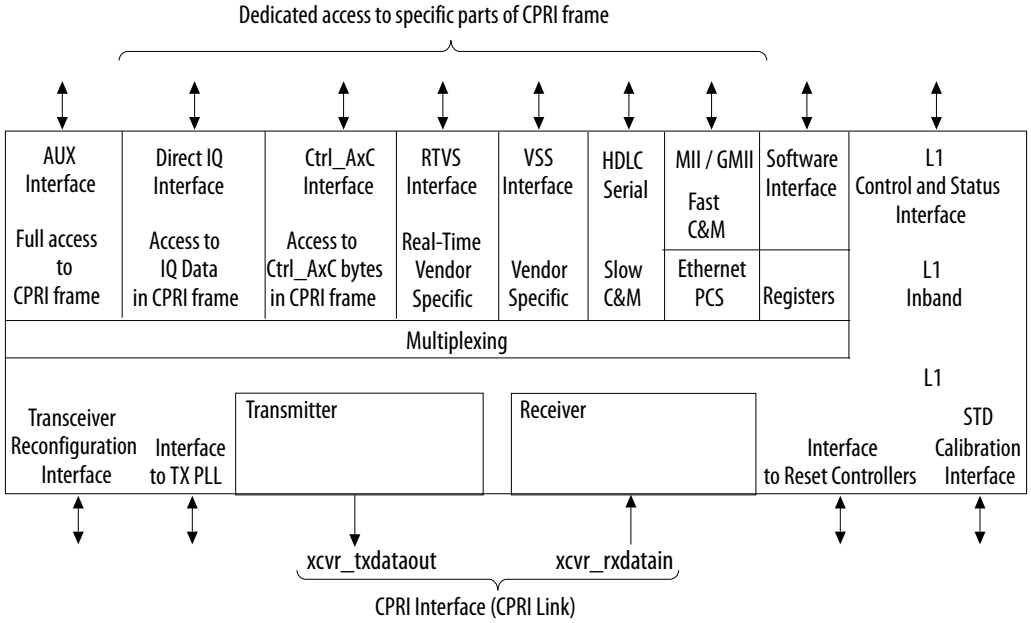
UG-20008 Subscribe Send Feedback

The CPRI v6.0 IP core implements Layer 1 of the CPRI V6.0 specification and provides optional CPRI V6.0 Layer 2 access points through various interfaces.

Interfaces Overview

Figure 3-1: CPRI v6.0 IP Core Interfaces

The IP core assembles the outbound CPRI frame control words and data from all of these interfaces, and unloads and routes control words and data from the inbound CPRI frame to the appropriate interfaces, based on configuration and register settings. With parameter settings, you control the presence or absence of the AUX interface, the L1 control and status interface, and each of the interfaces that provide dedicated access to specific parts of the CPRI frame. In contrast, the CPRI interface, the transceiver interfaces, and the software interface to the IP core registers are always implemented.



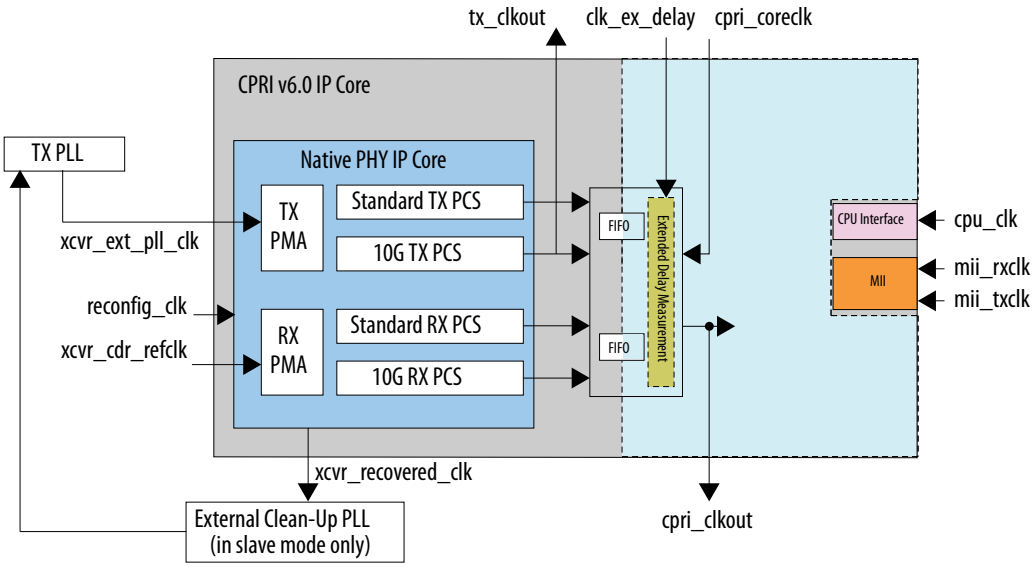
Multiple interfaces control the contents of the outbound CPRI frame control words and data. The CPRI v6.0 implements the following transmission priorities among these interfaces:

- CPRI frame control words:
  1. If the IP core implements the AUX interface, the AUX interface `aux_tx_data` bus, with appropriate delay, has first priority in filling in the outbound CPRI frame control words.
  2. If the IP core does not implement the AUX interface, or the `aux_tx_mask` value associated with the relevant incoming data blocks the relevant `aux_tx_data` bits, each of the following interfaces, if implemented, has secondary priority in filling the relevant part of the outbound CPRI frame control words:
     - Real-time vendor specific interface (RTVS)
     - Vendor specific interface (VS)
     - AxC control information interface (Ctrl_AxC)
  3. For any part of the CPRI frame control words not filled in by one of the previous methods, the transmission-enabled values most recently written to the control transmit table through the full control word access registers `CTRL_INDEX` and `TX_CTRL` determine the contents of the outbound CPRI frame control words. If the most recently written word for a CPRI frame position is not transmission-enabled, no transmission is authorized from the control transmit table to that CPRI frame position.
  4. If none of the previous methods provides the content for a position in the CPRI frame control word, the following interfaces, if implemented, have the lowest priority in filling the relevant part of the outbound CPRI frame control words:
     - Fast control and management (Ethernet) MII or GMII interface
     - Slow control and management (HDLC) serial interface
     - L1 control and status interface
     - Dedicated registers that contain or control content for control word positions in the CPRI frame. For example, the `rx_prot_ver_filter` field of the `PROT_VER` register
     - Transmission of special symbols according to the CPRI protocol. For example, K28.5, D16.2, /S/, or /T/
- CPRI frame I/Q data words:
  1. If the IP core implements the AUX interface, the AUX interface `aux_tx_data` bus, with appropriate delay, has first priority in filling in the outbound CPRI frame I/Q data words.
  2. If the IP core does not implement the AUX interface, or the `aux_tx_mask` value associated with the relevant incoming data blocks the relevant `aux_tx_data` bits, the Direct I/Q interface, if implemented, has secondary priority in filling the relevant part of the outbound CPRI frame I/Q data words.

CPRI v6.0 IP Core Clocking Structure

Figure 3-2: CPRI v6.0 IP Core Clocking Structure

Illustrates the clocks and clock domains in the CPRI v6.0 IP core. Clock domains shown are `cpri_clkout`, `clk_ex_delay`, `cpu_clk`, and `mii_{rx,tx}clk`. The external clean-up PLL is only required in slave clocking mode. The `tx_clkout` is only available in external clocking mode, and the `xcvr_recovered_clk` is only available in slave clocking mode.



The main CPRI v6.0 IP core clock is `cpri_clkout`.

Table 3-1: CPRI v6.0 IP Core Input Clocks

CPRI v6.0 Input Clock	Information
`xcvr_ext_pll_clk`	Clocks the transmitter PMA. You should drive this input clock with the output of the external TX PLL. The frequency of this clock must be one half of the CPRI line bit rate, multiplied by the local clock division factor. You must configure a PLL IP core that is capable of driving the required frequency.

CPRI v6.0 Input Clock	Information
xcvr_cdr_refclk or xcvr_cdr_refclk[1:0]	Receiver CDR reference clock. You must drive this clock at the frequency you specified for the **Receiver CDR reference clock frequency (MHz)** parameter in the CPRI v6.0 parameter editor. If you set the **Number of receiver CDR reference clock(s)** parameter in the CPRI v6.0 parameter editor to the value of **2**, this clock is two bits wide. In the case of a two-bit xcvr_cdr_refclk port, drive xcvr_cdr_refclk[0] with the reference clock for the initial CPRI line bit rate, because by default, this is the clock signal that drives the CDR. The IP core supports all CDR reference clock frequencies available in the drop-down menu for the **Receiver CDR reference clock frequency (MHz)** parameter.
reconfig_clk	In Arria V, Arria V GZ, Cyclone V, and Stratix V variations, clock for CPRI v6.0 IP core transceiver start-up and reconfiguration. In Intel Arria 10 and Intel Stratix 10 variations, clocks the signals on the CPRI v6.0 transceiver reconfiguration interface. In these variations, this clock is not present if you turn off all of **Enable start-up sequence state machine**, **Enable ADME, transceiver capability, control and status registers access**, and parameters only available in Intel Arria 10 variations, **Enable line bit rate auto-negotiation**, **Enable single-trip delay calibration**. In variations that target any other supported device family, this clock is not present if you turn off both **Enable line bit rate auto-negotiation**, **Enable start-up sequence state machine**. The supported frequency range of this clock is 100–150 MHz.
ex_delay_clk	Clock for extended delay measurement.
latency_sclk	Clock for delay measurement through the Intel Stratix 10 hard FIFO buffers in the PCS and the core. You can (but need not) drive this clock at the same frequency as ex_delay_clk. This clock is present only in IP cores that target an Intel Stratix 10 device.

<table>
<tr><th>CPRI v6.0 Input Clock</th><th>Information</th></tr>
<tr><td>cpri_coreclk</td><td>In internal clocking mode, drives the CPRI v6.0 IP core clock cpri_clkout when the IP core is running at the CPRI line bit rate of 8.11008 Gbps or 10.1376 Gbps. In external clocking mode, drives cpri_clkout at all CPRI line bit rates.

The frequency at which you must drive cpri_coreclk depends on the CPRI line bit rate:

<table>
<tr><th>CPRI Line Bit Rate</th><th>cpri_clkout Frequency</th></tr>
<tr><td>0.6144 Gbps</td><td>15.36 MHz</td></tr>
<tr><td>1.2288 Gbps</td><td>30.72 MHz</td></tr>
<tr><td>2.4576 Gbps</td><td>61.44 MHz</td></tr>
<tr><td>3.0720 Gbps</td><td>76.80 MHz</td></tr>
<tr><td>4.9152 Gbps</td><td>122.88 MHz</td></tr>
<tr><td>6.1440 Gbps</td><td>153.60 MHz</td></tr>
<tr><td>8.11008 Gbps</td><td>245.76 MHz</td></tr>
<tr><td>9.8304 Gbps</td><td>245.76 MHz</td></tr>
<tr><td>10.1376 Gbps</td><td>307.20 MHz</td></tr>
</table>

You must drive this clock from the same clock source as the xcvr_ext_pll_clk input signal to the IP core.</td></tr>
<tr><td>cpu_clk</td><td>Clocks the signals on the CPRI v6.0 CPU interface. Supports any frequency that the device fabric supports.</td></tr>
<tr><td>mii_txclk
mii_rxclk</td><td>mii_txclk clocks the MII transmitter interface and mii_rxclk clocks the MII receiver interface. You must drive these clocks at the frequency of 25 MHz/2.5 MHz to achieve the 100 Mbps/10 Mbps bandwidth required for this interface.

These clocks are present only if you set the value of Ethernet PCS interface to the value of MII in the CPRI v6.0 parameter editor.</td></tr>
<tr><td>gmii_txclk
gmii_rxclk</td><td>gmii_txclk clocks the GMII transmitter interface and gmii_rxclk clocks the GMII receiver interface. You must drive these clocks at the frequency of 125 MHz to achieve the 1000 Mbps bandwidth required for this interface.

These clocks are present only if you set the value of Ethernet PCS interface to the value of GMII in the CPRI v6.0 parameter editor.</td></tr>
</table>

Table 3-2: CPRI v6.0 IP Core Output Clocks

CPRI v6.0 Output Clock	Information
`cpri_clkout`	Master clock for the CPRI v6.0 IP core. In internal clocking mode, when the IP core is running at the CPRI line bit rate of 8.11008 Gbps or 10.1376 Gbps, the `cpri_coreclk` input clock drives `cpri_clkout`. At all other CPRI line bit rates, the Tx PCS drives `cpri_clkout`. In external clocking mode, the `cpri_coreclk` input clock drives `cpri_clkout` at all CPRI line bit rates. The frequency of `cpri_clkout` depends on the CPRI line bit rate:

CPRI Line Bit Rate	`cpri_clkout` Frequency
0.6144 Gbps	15.36 MHz
1.2288 Gbps	30.72 MHz
2.4576 Gbps	61.44 MHz
3.0720 Gbps	76.80 MHz
4.9152 Gbps	122.88 MHz
6.1440 Gbps	153.60 MHz
8.11008 Gbps	245.76 MHz
9.8304 Gbps	245.76 MHz
10.1376 Gbps	307.20 MHz

<table>
<tr><th>CPRI v6.0 Output Clock</th><th>Information</th></tr>
<tr><td>xcvr_recovered_clk</td><td>Direct recovered clock from the receiver CDR. Use this output clock to drive the external clean-up PLL when your IP core is in slave mode.

The IP core drives this clock from the PCS or the PMA block of the transceiver, depending on the value you set for the Recovered clock source parameter in the CPRI v6.0 parameter editor.

This clock is present only in CPRI v6.0 IP cores in slave clocking mode that support RX traffic. (This clock is not present in CPRI v6.0 IP cores with Operation mode set to the value of TX simplex).

The frequency of xcvr_recovered_clk depends on the CPRI line bit rate:

<table>
<tr><th>CPRI Line Bit Rate</th><th>xcvr_recovered_clk Frequency</th></tr>
<tr><td>0.6144 Gbps</td><td>15.36 MHz</td></tr>
<tr><td>1.2288 Gbps</td><td>30.72 MHz</td></tr>
<tr><td>2.4576 Gbps</td><td>61.44 MHz</td></tr>
<tr><td>3.0720 Gbps</td><td>76.80 MHz</td></tr>
<tr><td>4.9152 Gbps</td><td>122.88 MHz</td></tr>
<tr><td>6.1440 Gbps</td><td>153.60 MHz</td></tr>
<tr><td>8.11008 Gbps</td><td>220.75 MHz</td></tr>
<tr><td>9.8304 Gbps</td><td>245.76 MHz</td></tr>
<tr><td>10.1376 Gbps</td><td>253.44 MHz</td></tr>
</table>
</td></tr>
<tr><td>tx_clkout</td><td>TX PCS clock. In external clocking mode, you can use this clock to drive the cpri_coreclk input clock. If your IP core is configured with the single-trip delay calibration feature, you can use this clock to drive the IOPLL block.</td></tr>
</table>

Related Information

Adding the Off-Chip Clean-Up PLL on page 2-22

Example CPRI v6.0 Clock Connections in Different Clocking Modes

Figure 3-3: CPRI v6.0 Slave IP Core in Internal Clocking Mode

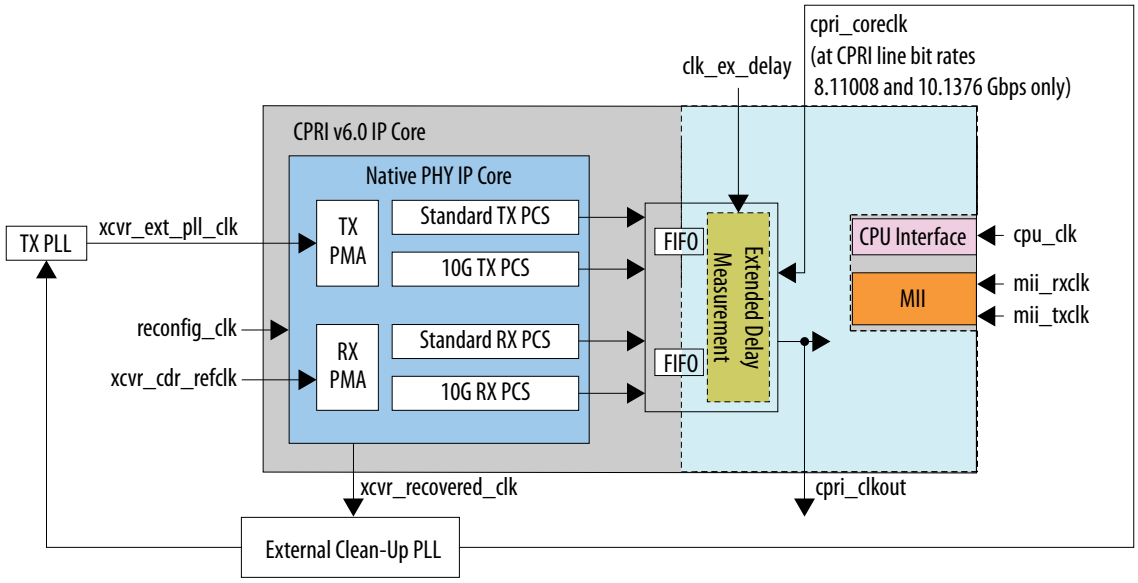


Figure 3-4: CPRI v6.0 Master IP Core in Internal Clocking Mode

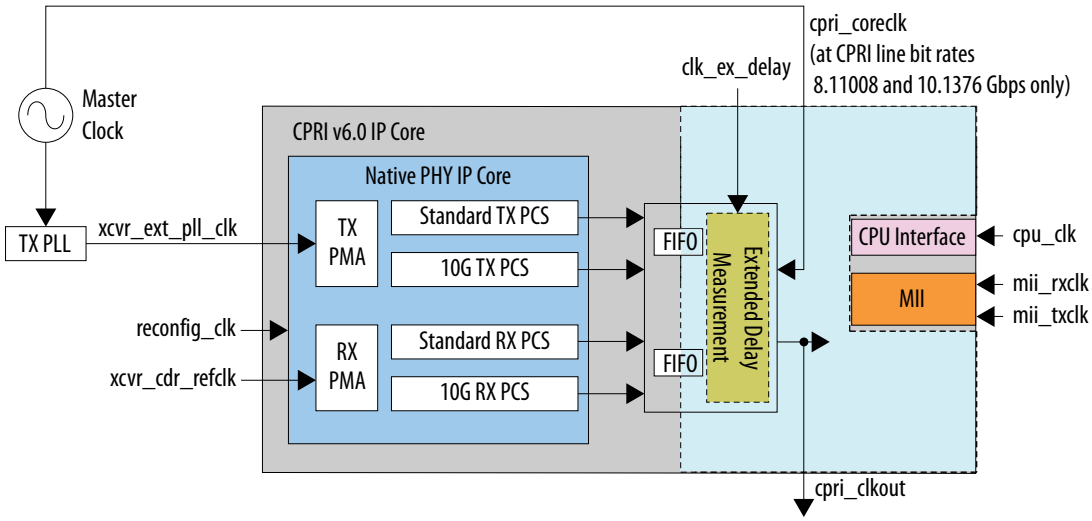


Figure 3-5: CPRI v6.0 Slave IP Core in External Clocking Mode

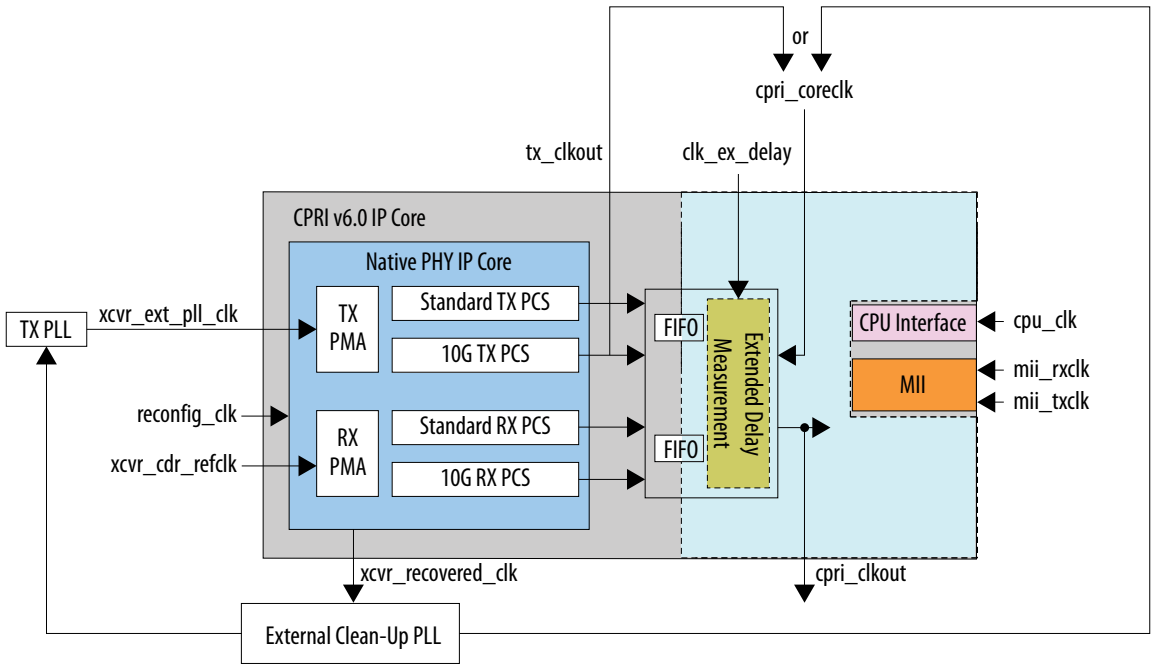


Figure 3-6: CPRI v6.0 Master IP Core in External Clocking Mode

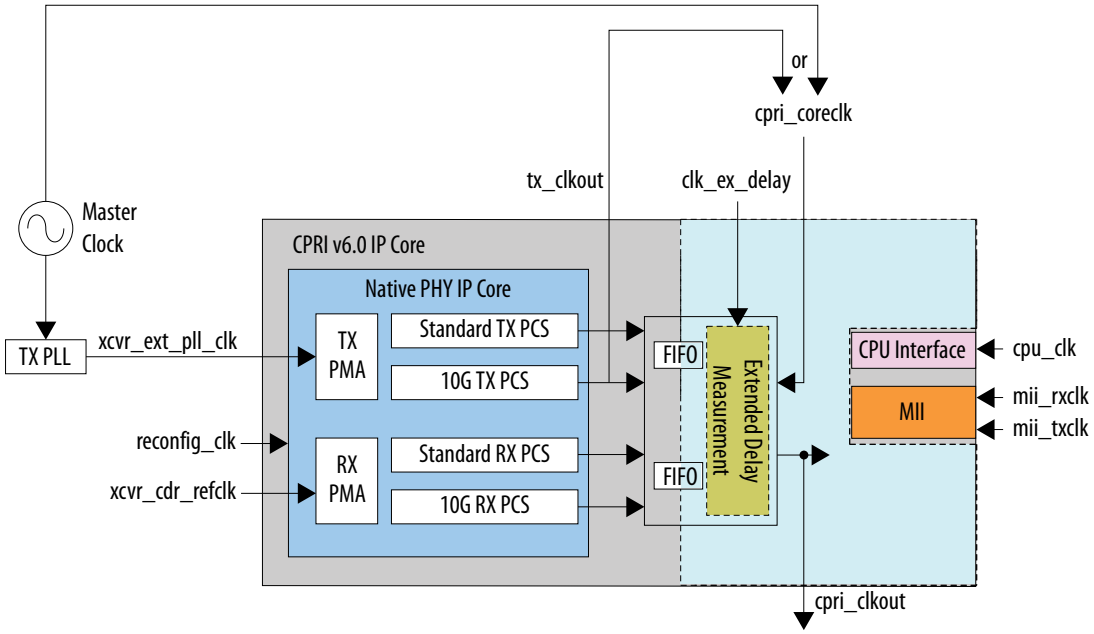
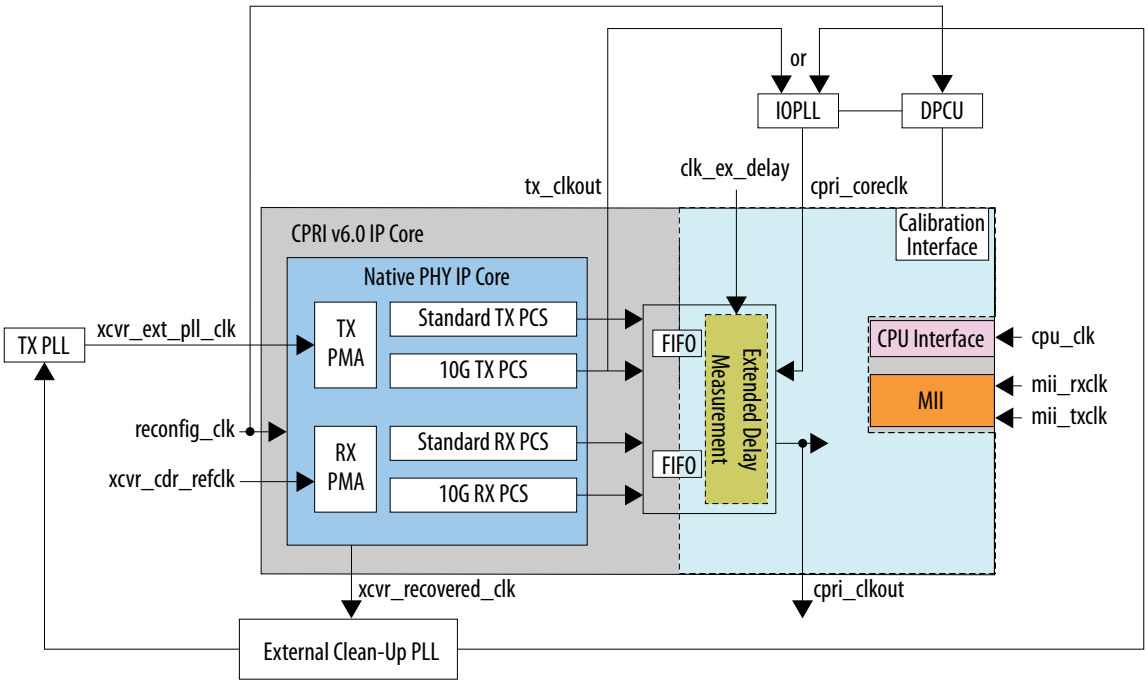


Figure 3-7: CPRI v6.0 Slave IP Core in External Clocking Mode with Single-Trip Delay Calibration Feature

Intel provides the IOPLL and DPCU blocks with the CPRI v6.0 IP core. For correct single-trip delay calibration functionality, you must connect these blocks as shown.



CPRI v6.0 IP Core Reset Requirements

To reset the entire CPRI v6.0 IP core, you must assert the reset signals to the IP core and to the required external reset controller logic. The external reset controller logic resets only the transceiver. If you instantiate a duplex CPRI v6.0 IP core, you must instantiate two PHY Reset Controllers to implement this logic, one for the TX data path and one for the RX data path. The two reset signals `reset_tx_n` and `reset_rx_n` each cause the reset logic to reset the relevant data path of the IP core. If you connect these two reset signals to the corresponding PHY Reset Controllers, each one also causes the transceiver in that data path to reset.

In the case of a duplex CPRI v6.0 IP core, you can assert the `reset_n` signal instead of asserting the two reset signals `reset_tx_n` and `reset_rx_n`. However, unless you also connect the `reset_n` signal to the external reset controllers, the transceivers do not reset in this case.

In addition, some individual interfaces to the IP core have their own reset signals to reset only the associated interface and logic.

Table 3-3: CPRI v6.0 IP Core Reset Signals

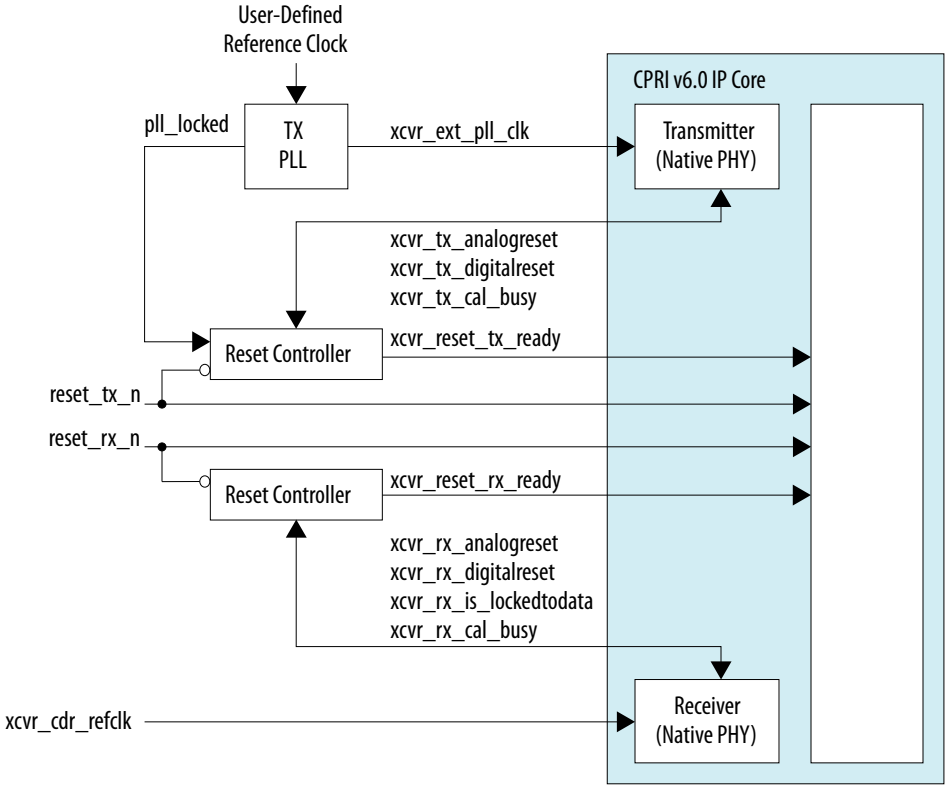
You can assert all reset signals asynchronously to any clock. However, you must hold each reset signal asserted for one full clock period of its associated clock, to ensure it is captured by the IP core.

CPRI v6.0 IP Core Reset Signal	Polarity	Associated Clock	Information
xcvr_tx_analogreset	Active high	—	Analog reset to transmitter from external reset controller.
xcvr_tx_digitalreset	Active high	—	Digital reset to transmitter from external reset controller.
xcvr_rx_analogreset	Active high	—	Analog reset to receiver from external reset controller.
xcvr_rx_digitalreset	Active high	—	Digital reset to receiver from external reset controller.
reset_n	Active low	reconfig_clk	Asynchronous global reset signal. Resets the IP core soft logic. This signal does not reset the CSR registers, the extended delay measurement settings, or the transceivers. To reset the CSR registers, you must assert the cpu_reset_n signal. To reset the extended delay measurement settings, you must assert the ex_delay_reset_n signal. To reset the transceiver you must drive the reset input port of the required PHY Reset Controllers.
reset_tx_n	Active low	reconfig_clk	Asynchronous global reset signal that resets the TX path of the CPRI v6.0 IP core. Resets the IP core soft logic. To reset the transceiver you must drive the reset input port of the required PHY Reset Controller.
reset_rx_n	Active low	reconfig_clk	Asynchronous global reset signal that resets the RX path of the CPRI v6.0 IP core. Resets the IP core soft logic. To reset the transceiver you must drive the reset input port of the required PHY Reset Controller.
ex_delay_reset_n	Active low	ex_delay_clk	Resets the extended delay measurement block.
latency_sreset_n	Active low	latency_sclk	Resets the Intel Stratix 10 hard FIFO delay measurement soft logic.

CPRI v6.0 IP Core Reset Signal	Polarity	Associated Clock	Information
reconfig_reset	Active high	reconfig_clk	Asynchronous reset signal. Resets the CPRI v6.0 Intel Arria 10 or Intel Stratix 10 transceiver reconfiguration interface and all of the registers to which it provides access. In IP cores that target a 28-nm device, this signal is involved in rate switching and auto-rate negotiation. In Intel Arria 10 and Intel Stratix 10 variations, this signal is not present if you turn off all of **Enable start-up sequence state machine**, **Enable ADME, transceiver capability, control and status registers access**, and parameters only available in Intel Arria 10 variations, **Enable line bit rate auto-negotiation**, **Enable single-trip delay calibration**. In variations that target other devices, this signal is not present if you turn off all of **Enable line bit rate auto-negotiation**, **Enable start-up sequence state machine**.
cpu_reset_n	Active low	cpu_clk	Resets the CPRI v6.0 CPU interface and all of the registers to which it provides access.
mii_txreset_n	Active low	mii_txclk	Resets the MII transmitter interface and FIFO write logic.
mii_rxreset_n	Active low	mii_rxclk	Resets the MII receiver interface and FIFO read logic.
gmii_txreset_n	Active low	gmii_txclk	Resets the GMII transmitter interface and FIFO write logic.
gmii_rxreset_n	Active low	gmii_rxclk	Resets the GMII receiver interface and FIFO read logic.

Figure 3-8: Required External Blocks

An example showing how you could connect required components to a single CPRI v6.0 IP core that targets an Arria 10 device.



To reset the CPRI v6.0 IP core, you must assert the active low `reset_tx_n`, `reset_rx_n`, `reset_tx_n` and `reset_rx_n`, or `reset_n` signals, as appropriate.

To reset the transceiver, you must trigger the reset controller logic. If you make the optional connection to drive the `reset_rx_n` or `reset_n` port from the same source as the reset signal for the RX side Reset Controller, asserting the active low `reset_rx_n` or `reset_n` signal also triggers the reset controller logic. If you make the optional connection to drive the `reset_tx_n` or `reset_n` port from the same source as the reset signal for the TX side Reset Controller, asserting the active low `reset_tx_n` or `reset_n` signal also triggers the TX reset controller logic.

When you trigger the reset controllers, they should deassert the `xcvr_reset_tx_ready` and `xcvr_reset_rx_ready` input ready signals to the IP core. After each reset controller completes resetting the transceiver and IP core data path, it should assert the relevant ready signal.

Related Information

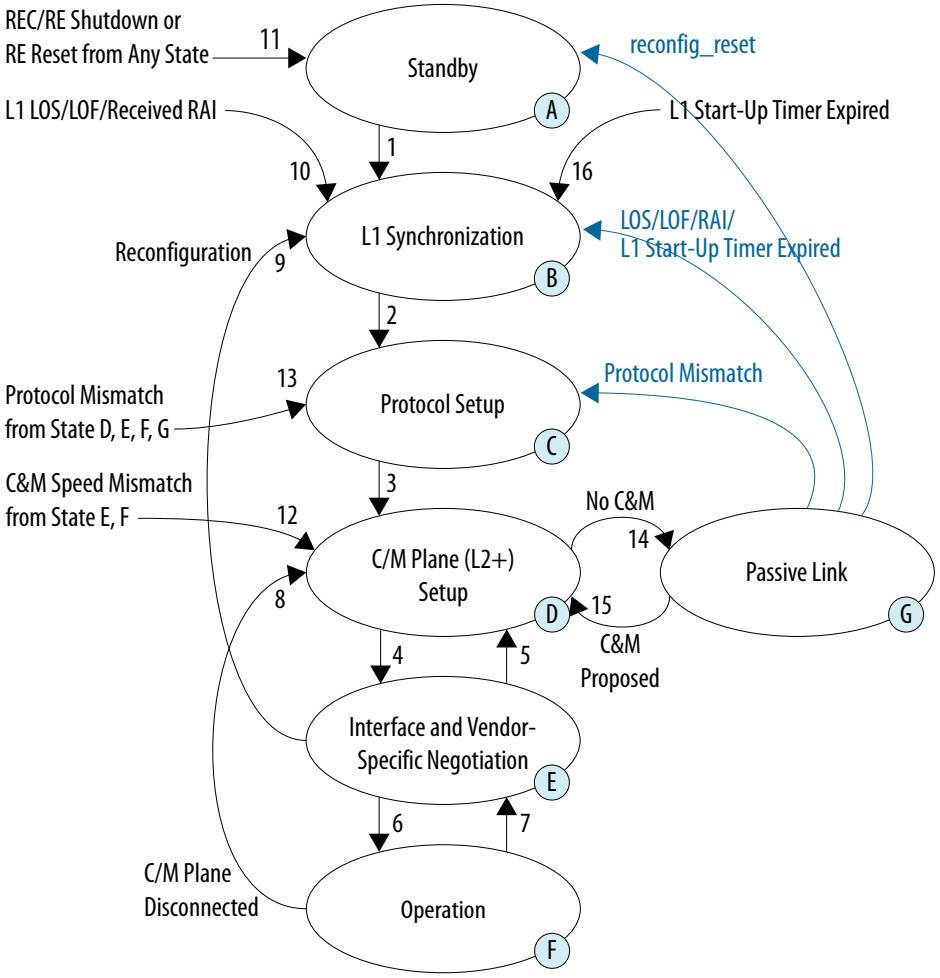
Integrating Your IP Core in Your Design: Required External Blocks on page 2-17
Shows main data path reset signals and how the reset controller connects to the IP core.

Start-Up Sequence Following Reset

After reset, if you turned on **Enable start-up sequence state machine** in the CPRI v6.0 IP core, the internal state machine performs link synchronization and other initialization tasks. If you did not turn on **Enable start-up sequence state machine**, user logic should perform these functions.

Figure 3-9: Start-up states and transitions Figure From CPRI v6.1 Specification With IP Core Additions

States and transitions marked in black are from the CPRI v6.1 specification Figure 30: Start-up states and transitions. Additional transitions from State G that the IP core implements are marked in blue.



The internal state machine implements the start-up state machine transitions shown in section 4.5.2, Figure 30: Start-up states and transitions, and described in section 4.5.3, in the CPRI specification. In addition, the internal state machine implements the following transitions from State G: Passive Link.

- If you assert `reconfig_reset`, the start-up state machine transitions to State A.
- If the IP core detects any of the following situations, the start-up state machine transitions to State B:
  - **Loss of signal (LOS).** Refer to *Direct L1 Control and Status Interface* and *FLSAR Register*.
  - **Loss of frame (LOF).** Refer to *Direct L1 Control and Status Interface* and *FLSAR Register*.
  - **Remote alarm indication (RAI).** Refer to *Direct L1 Control and Status Interface* and *FLSAR Register*.
  - **L1 start-up timer expiration.** If you turn on **Enable start-up sequence state machine** option, the IP core responds to both the `nego_l1_timer_expired` port and `startup_timer_expired` field of the `START_UP_SEQ` register at offset 0x24.
- If the IP core detects that protocol negotiation is not complete, the start-up state machine transitions to State C. The IP core detects that protocol negotiation is not complete if all of the following conditions hold:
  - `nego_protocol_complete` signal has the value of 0, and
  - `nego_protocol_complete` field of the `START_UP_SEQ` register at offset 0x24 has the value of 0, and
  - Slave protocol version does not match the protocol version of the CPRI master. Specifically, the `rx_prot_ver` and `tx_prot_ver` fields of the `PROT_VER` register at offset 0x10 have different values. When you turn on **Enable protocol version and C&M channel setting auto-negotiation** option, the IP core allows deframer to detect the incoming protocol version to the `csr_rx_prot_ver`, and compare it to the proposed `tx_prot_ver`. Then IP core detects that the protocol negotiation is complete.
- If the IP core detects that control and management negotiation is complete, the start-up state machine transitions to State D. The IP core detects that control and management negotiation is not yet complete if all of the following conditions hold:
  - `nego_cm_complete` signal has the value of 0, and
  - `nego_cm_complete` field of the `START_UP_SEQ` register at offset 0x24 has the value of 0, and
  - `rx_slow_cm_rate_valid` field of the `CM_STATUS` register has the value of 0, and
  - `rx_fast_cm_ptr_valid` field of the `CM_STATUS` register has the value of 0.

Related Information

- **Direct L1 Control and Status Interface** on page 3-40
- **FLSAR Register** on page 5-11

Start-Up Sequence Interface Signals

After reset, if you turned on **Enable start-up sequence state machine** in the CPRI v6.0 IP core, the internal state machine performs link synchronization and other initialization tasks. If you did not turn on **Enable start-up sequence state machine**, user logic must perform these functions.

The signals visible in the interface depend on whether or not you turned on **Enable start-up sequence state machine**.

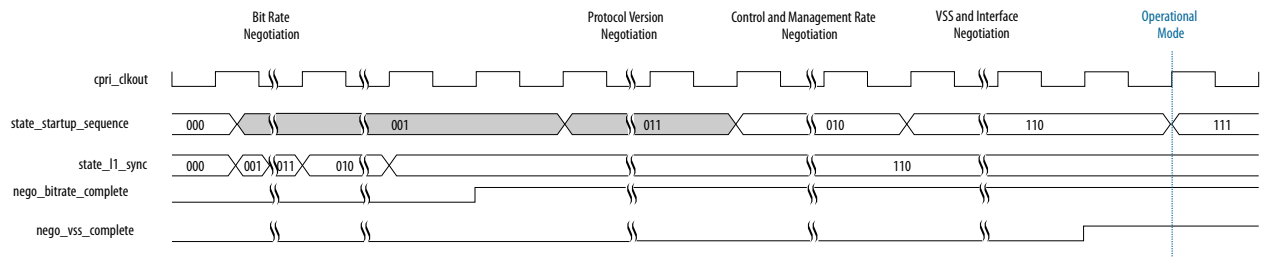
Table 3-4: Start-Up Sequence Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Signal Name	Direction	Description
state_startup_seq[2:0]	Output	Indicates the state of the CPRI start-up sequence state machine. This signal has the following valid values: • 3'b000: State A: Standby • 3'b001: State B: L1 Synchronization • 3'b011: State C: Protocol Setup • 3'b010: State D: Control and Management Setup • 3'b110: State E: Interface and VSS Negotiation • 3'b111: State F: Operation • 3'b101: State G: Passive Link This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor.
state_l1_synch[2:0]	Output	State B condition indicator. Indicates the state of the CPRI receiver L1 synchronization state machine. This signal has the following valid values: • 3'b000: XACQ1 • 3'b001: XACQ2 • 3'b011: XSYNC1 • 3'b010: XSYNC2 • 3'b110: HFNSYNC
nego_bitrate_complete	Input	Indicates the CPRI line bit rate negotiation is complete. Input from external CPRI line bit rate negotiation block. If you do not turn on **Enable line bit rate auto-negotiation** in the CPRI v6.0 parameter editor, you should tie this signal high. This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor. Asserting this signal advances the start-up sequence state machine from state B to state C. The IP core writes the value of this signal to the nego_bitrate_complete field of the START_UP_SEQ register at offset 0x24.
nego_protocol_complete	Input	Indicates the CPRI protocol version negotiation is complete. This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor. Asserting this signal advances the start-up sequence state machine from state C to state D. The IP core writes the value of this signal to the nego_protocol_complete field of the START_UP_SEQ register at offset 0x24.

Signal Name	Direction	Description
nego_cm_complete	Input	Indicates the Control and Management negotiation is complete. This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor. Asserting this signal advances the start-up sequence state machine from state D to state E. The IP core writes the value of this signal to the nego_cm_complete field of the START_UP_SEQ register at offset 0x24.
nego_vss_complete	Input	Indicates the Vendor Specific negotiation is complete. This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor. Asserting this signal advances the start-up sequence state machine from state E to state F. The IP core writes the value of this signal to the nego_vss_complete field of the START_UP_SEQ register at offset 0x24.
nego_l1_timer_expired	Input	If you do not turn on **Enable protocol version and C&M channel setting auto-negotiation** in the CPRI v6.0 parameter editor, drive this signal from your user-defined L1 timer to indicate that the L1 timer has expired. Note that if you do not turn on **Enable protocol version and C&M channel setting auto-negotiation**, user logic is expected to maintain an L1 timer outside the IP core. This signal is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor. If you also turn on **Enable protocol version and C&M channel setting auto-negotiation** in the CPRI v6.0 parameter editor, you should tie this signal low so it does not interfere with the internal L1 timer.

Figure 3-10: Start-Up Sequence State Machine Timing Diagram



AUX Interface

The CPRI v6.0 IP core auxiliary (AUX) interface provides direct access to the CPRI 10 ms radio frame, including I/Q data and control words. You can use this interface to support your specific application. For example, the AUX interface allows you to implement custom I/Q sample widths and custom mapping schemes.

The AUX interface also enables multi-hop routing applications and provides timing reference information for transmitted and received frames. Using this interface, you can load I/Q data in a precise location in the precise CPRI basic frame you target.

The AUX interface allows you to connect CPRI v6.0 IP core instances and other system components together by supporting a direct connection to a user-defined routing layer or custom mapping block. You implement this routing layer, which is not defined in the CPRI V6.0 Specification, outside the CPRI v6.0 IP core. The AUX interface supports the transmission and reception of I/Q data and timing information between an RE slave and an RE master, allowing you to define a custom routing layer that enables daisy-chain configurations of RE master and slave ports. Your custom routing layer determines the I/Q sample data to pass on to other REs to support multi-hop network configurations and custom mapping algorithms.

If you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor, your IP core includes this interface.

AUX Interface Signals

Table 3-5: AUX Interface Signals

If you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor, the AUX interface is available. This interface allows access to the entire CPRI frame and has the highest priority among the L1 interfaces.

You can alter the transmit write latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter. The default transmit latency, when **Auxiliary and direct interfaces write latency cycle(s)** has the value of zero, is one `cpri_clkout` cycle. You can specify additional latency cycles.

All interface signals are clocked by the `cpri_clkout` clock.

AUX RX Interface Status Signals		
Signal Name	Direction	Description
`aux_rx_rfp`	Output	Synchronization pulse for start of 10 ms radio frame. The pulse occurs at the start of the radio frame on the AUX RX interface.
`aux_rx_hfp`	Output	Synchronization pulse for start of hyperframe. The pulse occurs at the start of the hyperframe on the AUX RX interface.
`aux_rx_bfn[11:0]`	Output	Current radio frame number on the AUX RX interface.
`aux_rx_z[7:0]`	Output	Current hyperframe number on the AUX RX interface. Value is in the range 0–149.

AUX RX Interface Status Signals		
Signal Name	**Direction**	**Description**
aux_rx_x[7:0]	Output	Index number of the current basic frame in the current hyperframe on the AUX RX interface. Value is in the range 0–255.
aux_rx_seq[6:0]	Output	Index number of the current 32-bit word in the current basic frame on the AUX RX interface. The value range depends on the current CPRI line bit rate: • 0.6144 Gbps: range is 0–3 • 1.2288 Gbps: range is 0–7 • 2.4576 Gbps: range is 0–15 • 3.0720 Gbps: range is 0–19 • 4.9152 Gbps: range is 0–31 • 6.1440 Gbps: range is 0–39 • 8.11008 Gbps: range is 0–63 • 9.8304 Gbps: range is 0–63 • 10.1376 Gbps: range is 0–79
AUX RX Interface Data Signals		
Signal Name	**Direction**	**Description**
aux_rx_data[31:0]	Output	Data the IP core presents on the AUX link. Data is transmitted in 32-bit words. Byte [31:24] is transmitted first and byte [7:0] is transmitted last.
aux_rx_ctrl[3:0]	Output	Control slots indicator. Each asserted bit indicates that the corresponding byte position in aux_rx_data holds a byte from a CPRI control word.
AUX TX Interface Control and Status Signals		
Signal Name	**Direction**	**Description**
aux_tx_sync_rfp	Input	Synchronization input used in REC master to control the start of a new 10 ms radio frame. Asserting this signal resets the frame synchronization machine. The CPRI v6.0 IP core uses the rising edge of the pulse for synchronization.
aux_tx_err[3:0]	Output	Indicates that in the previous cpri_clkout cycle, aux_tx_mask bits masked one or more control words in the target CPRI frame. Each bit in aux_tx_err indicates whether the corresponding byte in the 32-bit value on aux_tx_data overwrites a control word in the target CPRI frame.
aux_tx_rfp	Output	Synchronization pulse for start of 10 ms radio frame. The pulse occurs at the start of the radio frame on the AUX TX interface.

AUX TX Interface Control and Status Signals		
Signal Name	**Direction**	**Description**
aux_tx_hfp	Output	Synchronization pulse for start of hyperframe. The pulse occurs at the start of the hyperframe on the AUX TX interface.
aux_tx_bfn[11:0]	Output	Current radio frame number on the AUX TX interface.
aux_tx_z[7:0]	Output	Current hyperframe number on the AUX TX interface. Value is in the range 0–149.
aux_tx_x[7:0]	Output	Index number of the current basic frame in the current hyperframe on the AUX TX interface. Value is in the range 0–255.
aux_tx_seq[6:0]	Output	Index number of the current 32-bit word in the current basic frame on the AUX TX interface. The value range depends on the current CPRI line bit rate: • 0.6144 Gbps: range is 0–3 • 1.2288 Gbps: range is 0–7 • 2.4576 Gbps: range is 0–15 • 3.0720 Gbps: range is 0–19 • 4.9152 Gbps: range is 0–31 • 6.1440 Gbps: range is 0–39 • 8.11008 Gbps: range is 0–63 • 9.8304 Gbps: range is 0–63 • 10.1376 Gbps: range is 0–79
AUX TX Interface Data Signals		
Signal Name	**Direction**	**Description**
aux_tx_data[31:0]	Input	Data the IP core receives on the AUX TX interface. The data is aligned with aux_tx_seq with a write delay of one cpri_clkout cycle plus the number of additional cpri_clkout cycles you specify as the value of the **Auxiliary and direct interfaces write latency cycle(s)** parameter. User logic is responsible to ensure that the write data in aux_tx_data is aligned with the write latency value of the **Auxiliary and direct interfaces write latency cycle(s)** parameter. Data is received in 32-bit words. For correct transmission in the CPRI frame, you must send byte [31:24] first and byte [7:0] last.

AUX TX Interface Data Signals		
Signal Name	**Direction**	**Description**
`aux_tx_mask[31:0]`	Input	Bit mask for insertion of data from `aux_tx_data` in the target CPRI frame. This signal aligns with `aux_tx_data` and therefore, aligns with `aux_tx_seq` with a delay of one `cpri_clkout` cycle plus the number of additional `cpri_clkout` cycles you specify as the value of the **Auxiliary and direct interfaces write latency cycle(s)** parameter. Assertion of a bit in this mask overrides insertion of data to the corresponding bit in the target CPRI frame from any other source. Therefore, you must deassert the mask bits during K28.5 character or /S/ /T/ insertion in the outgoing CPRI frame, which occurs when Z=X=0. If you do not deassert the mask bits during K28.5 or /S/ /T/ character insertion in the outgoing CPRI frame, the `aux_tx_err` output signal is asserted in the following `cpri_clkout` cycle.
`aux_tx_ctrl[3:0]`	Output	Control slots indicator. Each asserted bit indicates that the corresponding byte position, as indicated by `aux_tx_seq`, should hold a CPRI control word in the target CPRI frame.

Figure 3-11: AUX RX Interface Timing Diagram

AUX RX interface behavior in a CPRI v6.0 IP core running at 0.6144 Gbps.

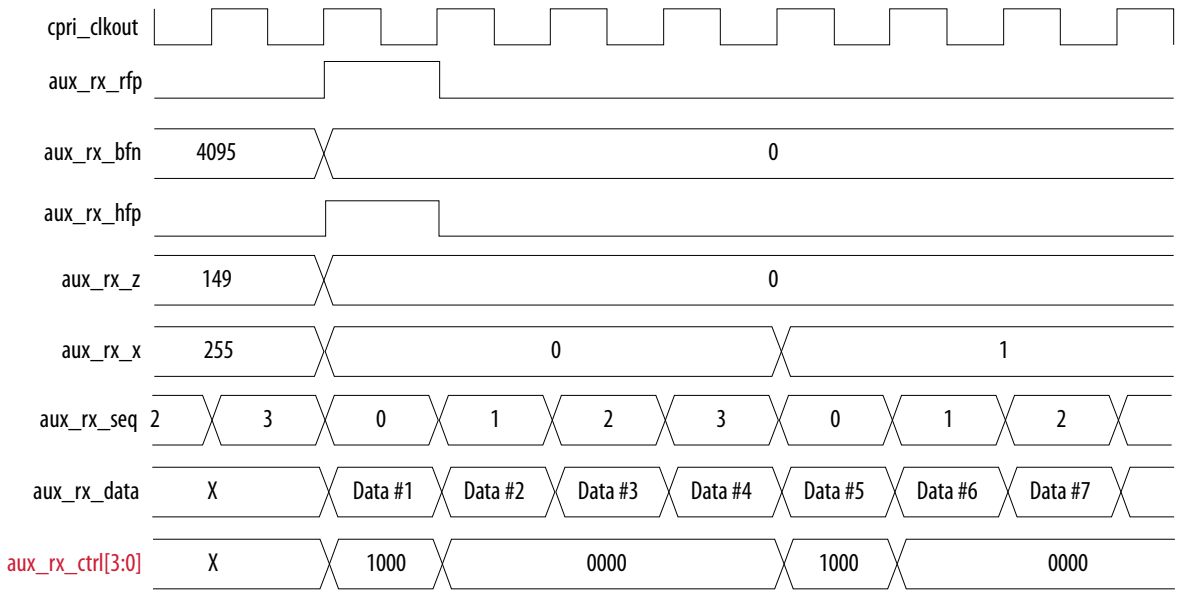


Figure 3-12: CPRI REC Master Response to aux_tx_sync_rfp Resynchronization Pulse

Asserting `aux_tx_sync_rfp` resets the hyperframe and basic frame numbers in an REC master CPRI v6.0 IP core. Shown for a CPRI v6.0 IP core running at 0.6144 Gbps.

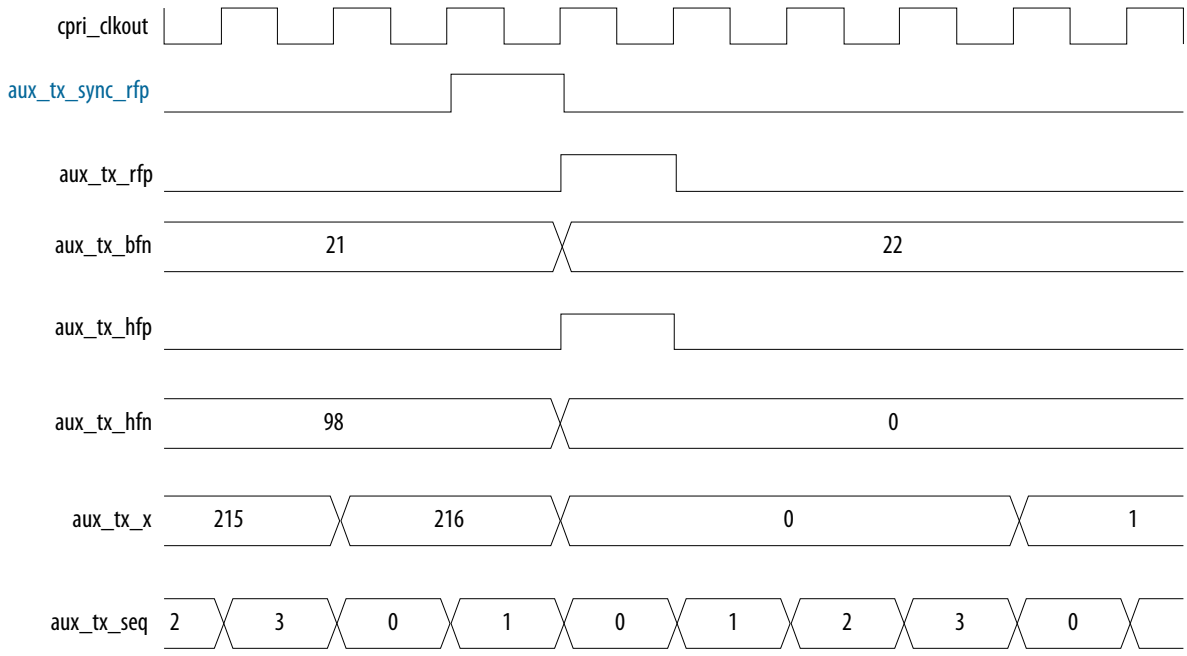


Figure 3-13: AUX TX Interface Timing Diagram with One Auxiliary Latency Cycle

Expected behavior on the AUX TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps. Illustrates the effect of setting the **Auxiliary and direct interfaces write latency cycle(s)** parameter to a a non-zero value. Shown for a CPRI v6.0 IP core with **Auxiliary and direct interfaces write latency cycle(s)** set to the value of 1.

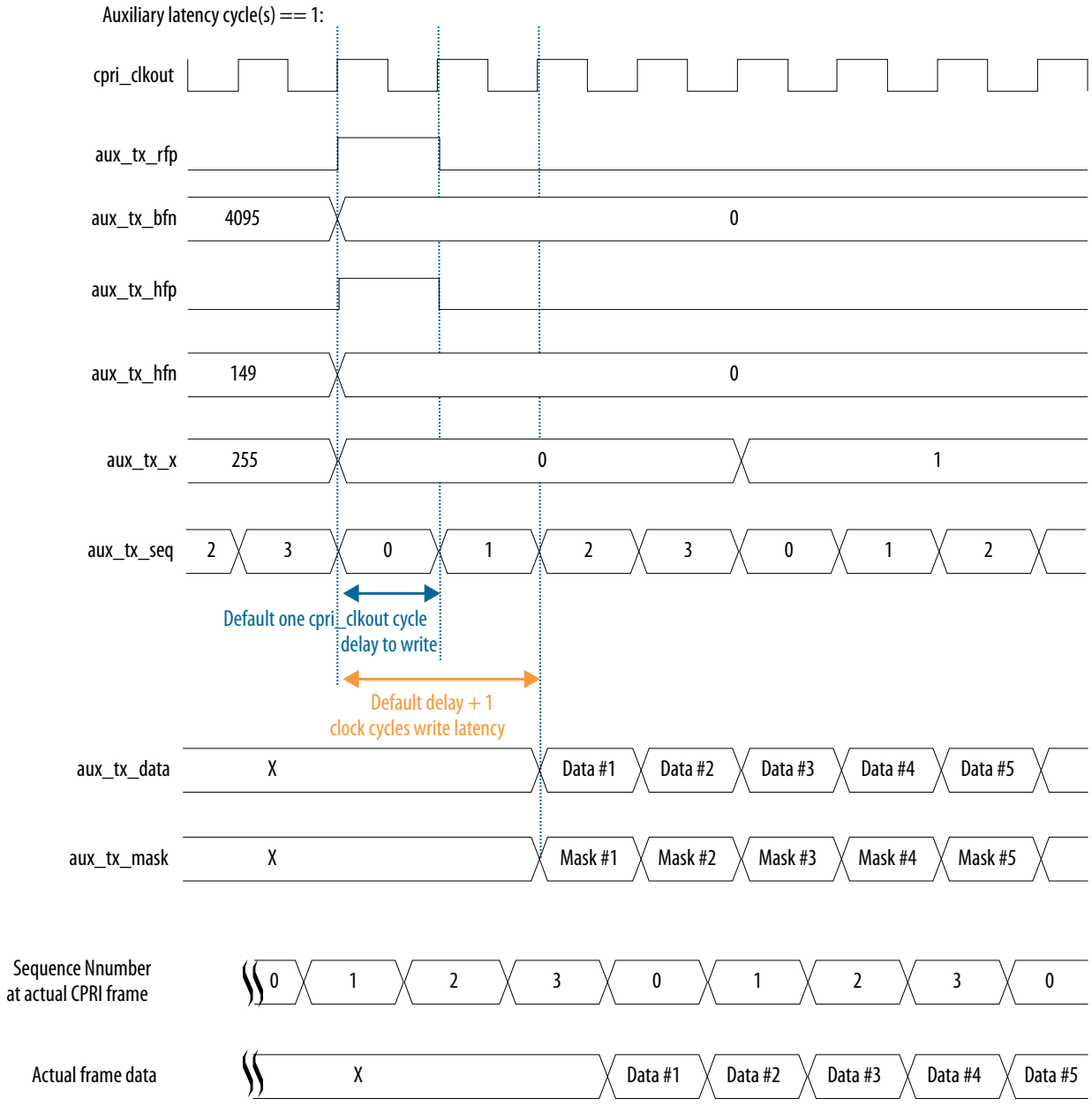


Figure 3-14: AUX TX Interface Timing Diagram with Four Auxiliary Latency Cycles

Expected behavior on the AUX TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps. Illustrates the effect of setting the **Auxiliary and direct interfaces write latency cycle(s)** parameter to the value of four.

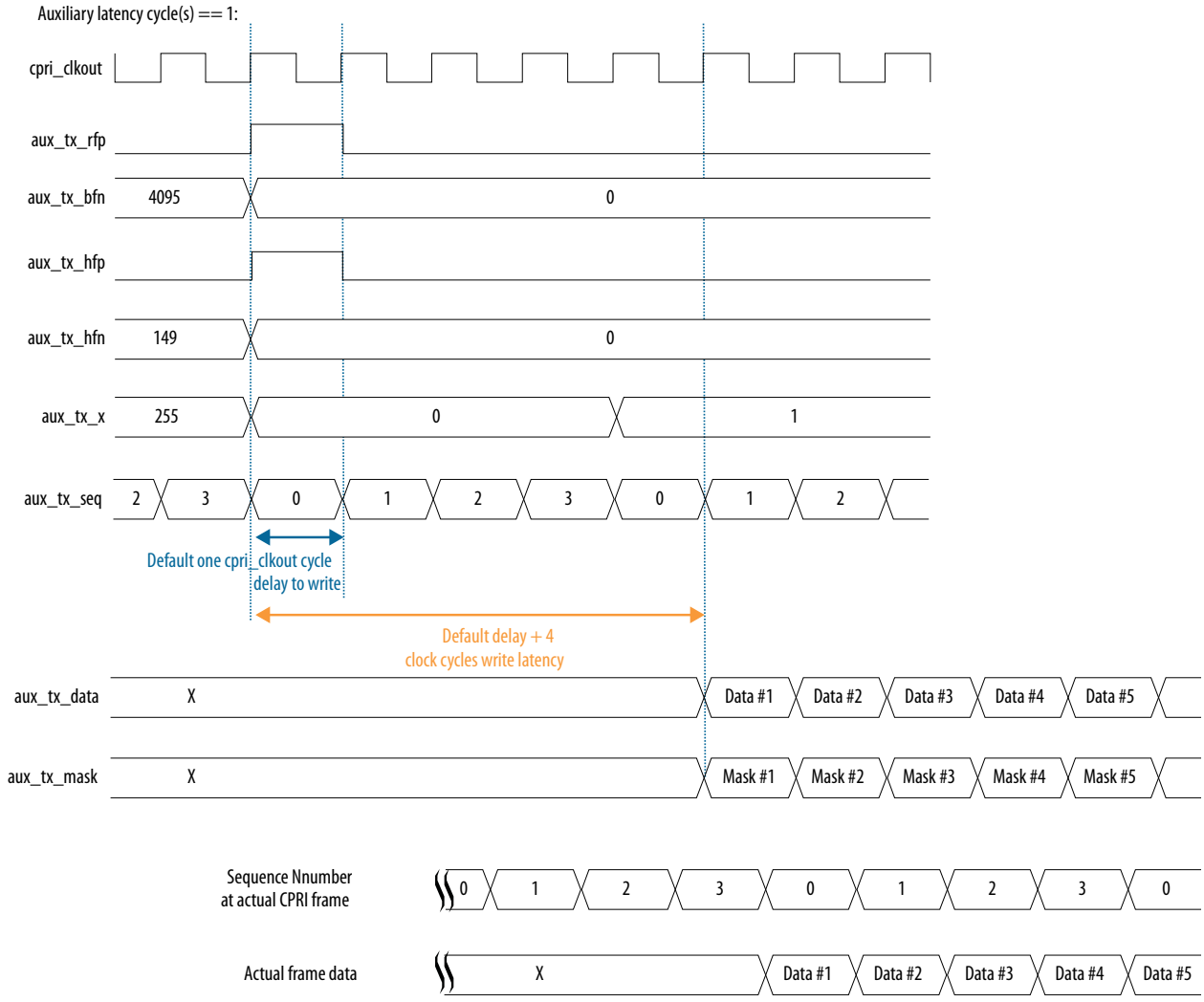
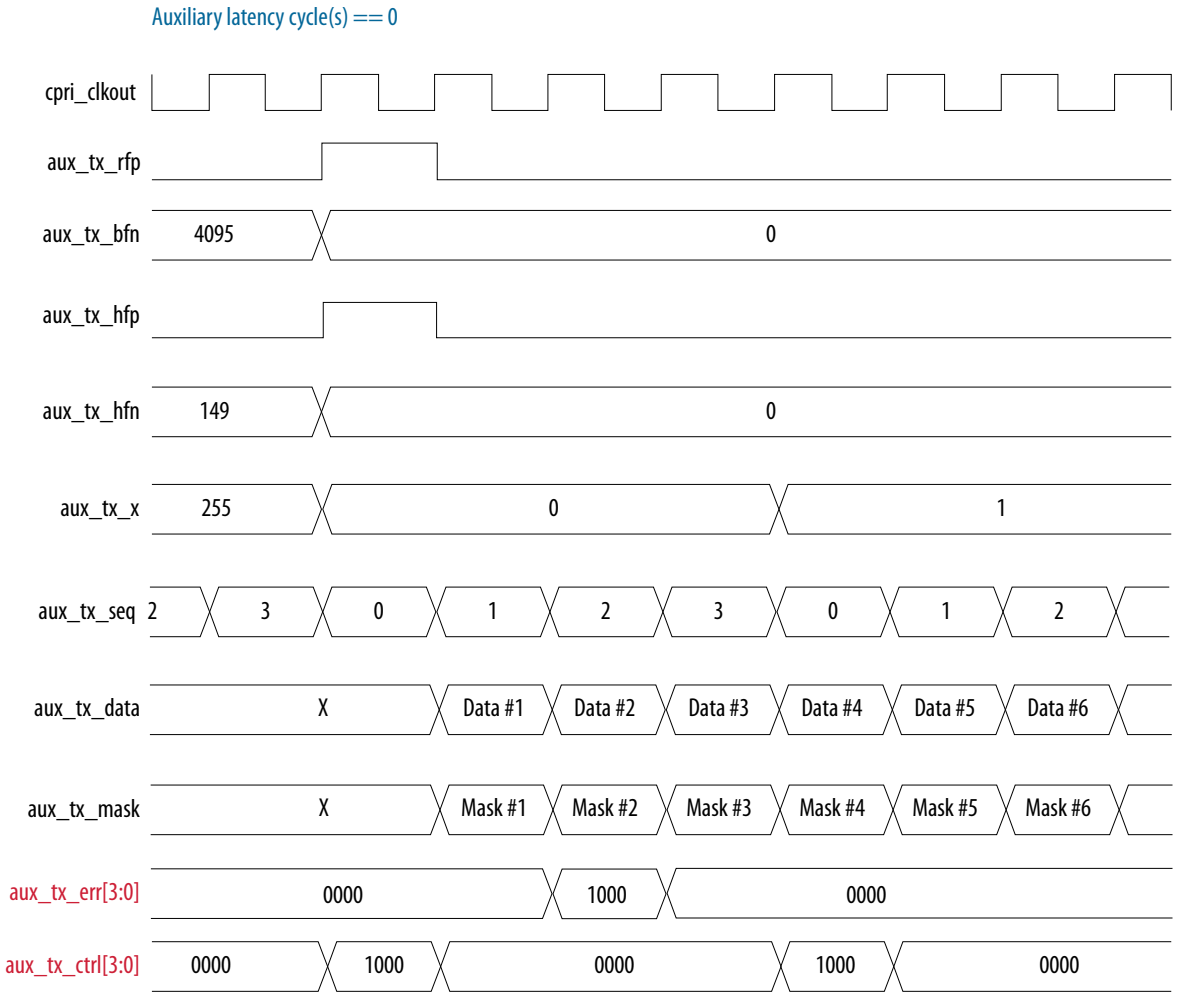


Figure 3-15: AUX TX Timing Diagram with Error

Illustrates the behavior of the aux_tx_err signal on the AUX TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps. The aux_tx_ctrl signal shows that when aux_tx_seq has the value of zero, the first byte at the corresponding position in the target CPRI frame is a control byte. The value of the **Auxiliary and direct interfaces write latency cycle(s)** parameter is zero. Therefore, the data on aux_tx_data is delayed by one clock cycle from the value on aux_tx_seq. The data that appears on aux_tx_data when aux_tx_seq has the value of 1 is the data that targets position X.Y.Z.0 in the target CPRI frame.

The value of Mask #1 is presumably 0xFFXXXXXX, indicating that the incoming data on aux_tx_data is intended to overwrite this control byte in the target CPRI frame. Therefore, in the following cpri_clkout cycle, the IP core asserts the aux_tx_err signal.



Related Information

- **AUX Interface Synchronization** on page 3-26
  Illustrates the relationship between the AUX synchronization signals.

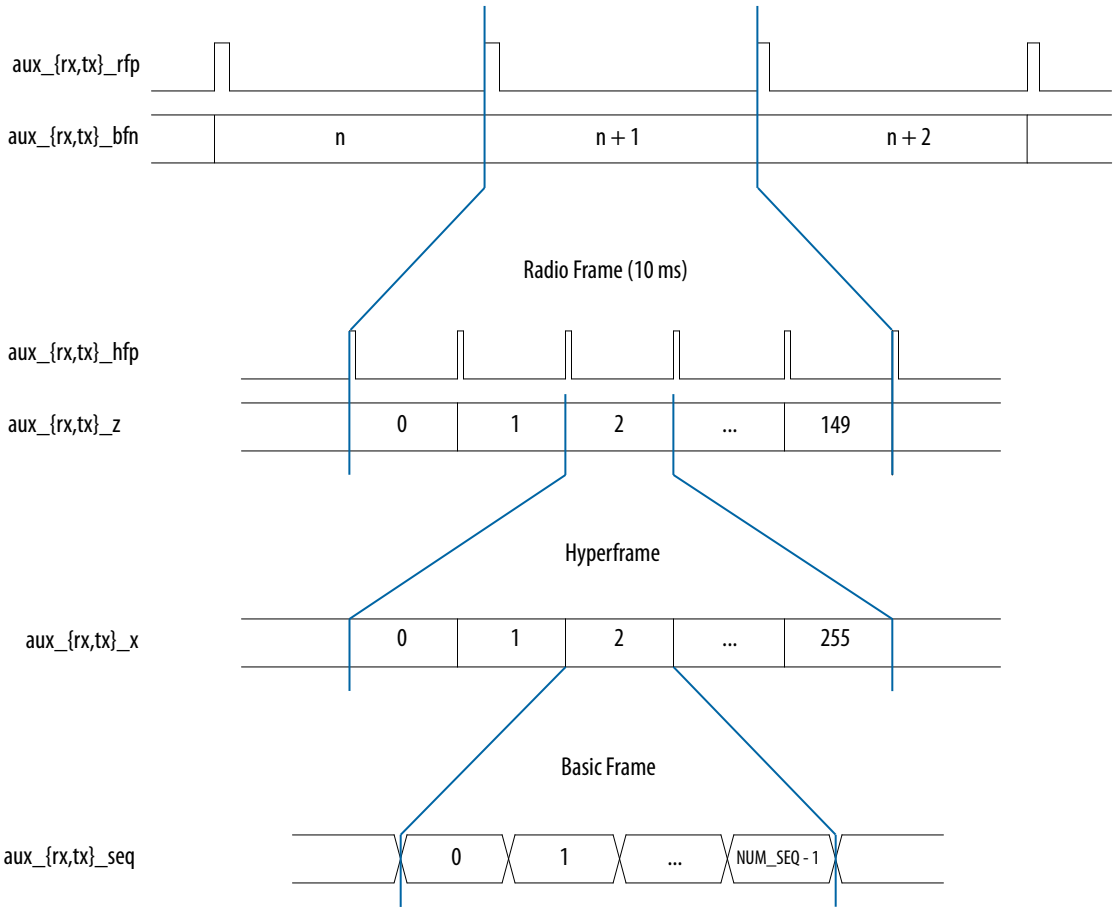
- **Auxiliary Latency Cycles** on page 3-26
  The **Auxiliary and direct interfaces write latency cycle(s)** parameter affects the relative timing of the data on the aux_tx_data and aux_tx_mask busses. This section provides an explanation of the parameter's effect and purpose.

AUX Interface Synchronization

Figure 3-16: Relationship Between Synchronization Pulses and Numbers on the AUX Interface

The output synchronization signals are useful for custom user logic, including frame synchronization across hops in multi-hop configurations.

The output synchronization signals are derived from the CPRI frame synchronization state machine.



Related Information

AUX Interface Signals on page 3-18
Describes the AUX interface signals and provides AUX interface timing diagrams.

Auxiliary Latency Cycles

Intel provides configurable write latency on the AUX TX interface and other direct TX interfaces to support user logic with sufficient advance notice of the position in the CPRI frame. The processing time

that user logic requires after determining the current position in the CPRI frame is implementation specific, and the default write latency of a single `cpri_clkout` cycle might not be adequate. Using the **Auxiliary and direct interfaces write latency cycle(s)** parameter, you can set the write latency to the number of clock cycles required for your system to process data before sending it on the AUX TX interface or other direct TX interface.

In the CPRI v6.0 parameter editor, you can specify a non-zero number of **Auxiliary and direct interfaces write latency cycle(s)** to increase the write latency on the AUX TX interface and other direct TX interfaces to the CPRI v6.0 IP core.

The write latency is the number of `cpri_clkout` cycles from when the `aux_tx_seq` output signal has the value of n to when user logic must write data to the AUX TX interface to target the corresponding position in the CPRI frame. For other direct interfaces, the IP core notifies user logic when it is ready for input and the user does not need to monitor the `aux_tx_seq` signal. However, the **Auxiliary latency cycle(s)** value does apply to all of the direct interfaces.

When **Auxiliary and direct interfaces write latency cycle(s)** has the default value of zero, the write latency on the direct TX interfaces is one `cpri_clkout` cycle. When **Auxiliary and direct interfaces write latency cycle(s)** has the value of N, the write latency is (1+N) `cpri_clkout` cycles.

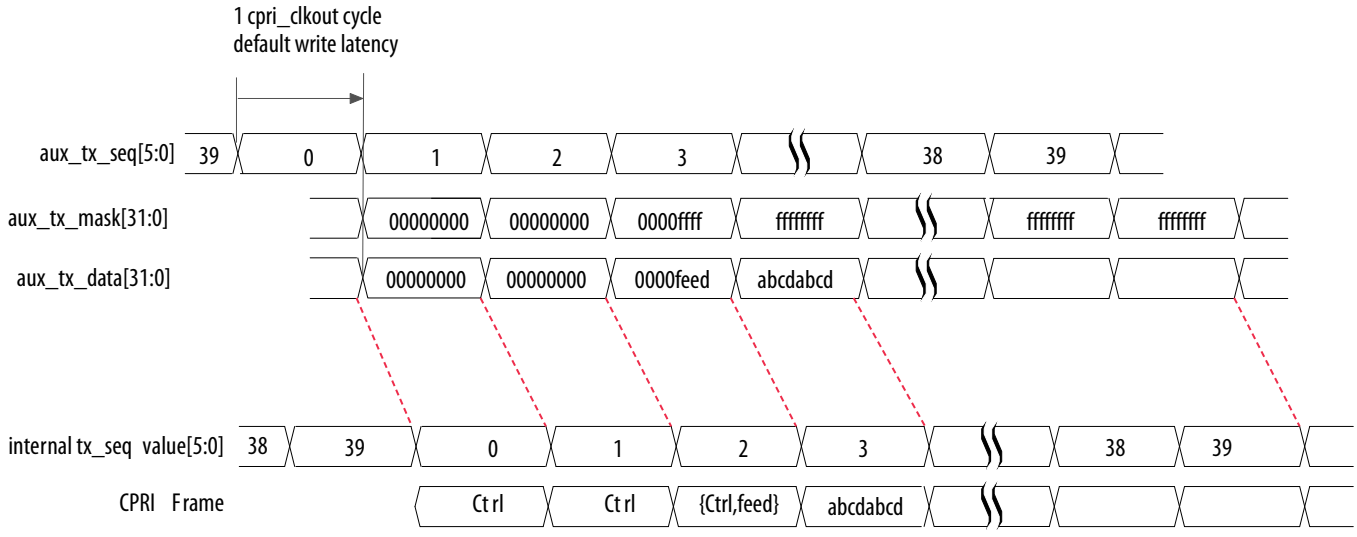
User logic is responsible to ensure that the data presented to the IP core on the AUX TX interface is presented at the correct write latency relative to the AUX TX interface synchronization signals.

Note: You cannot simply write to the AUX TX interface with a consistent write latency that you determine after configuring your IP core. If you do not specify the correct write latency in the CPRI v6.0 parameter editor, the data you present on the AUX TX interface will not fill the correct position in the target CPRI frame. To ensure the write latency offset is implemented correctly in the IP core, you must set the parameter.

Figure 3-17: AUX Interface Transmit Write Latency

Illustrates the transmit write latency on the AUX interface when **Auxiliary and direct interfaces write latency cycle(s)** has the value of 0. If you specify a non-zero value for this parameter, the latency increases from the default latency of one `cpri_clkout` cycle to 1 plus the number of cycles you specify.

In this example, the CPRI line bit rate is 6.144 Gbps, so that the control word is 10 bytes. User logic masks the control word, so that the IP core does not receive the control words from the AUX interface.



Direct Interface CPRI Frame Data Format

The information on the AUX interface and all of the other direct interfaces except the L1 CSR interface, appears in the relevant data bus in 32-bit words. The CPRI v6.0 IP core converts the contents of the incoming CPRI frame to a 32-bit format internally. Similarly, the IP core expects to receive data on the various direct interfaces in this format. The only exception is the L1 CSR interface, which transmits and receives information in individual bits.

Figure 3-18: AUX Interface Data at Different CPRI Line Bit Rates

The AUX interface presents and expects data in fixed 32-bit words. The mapping of the CPRI frame to and from 32-bit words depends on the CPRI v6.0 IP core bit rate. This figure illustrates how CPRI frame words are mapped to 32-bit words on the AUX interface 32-bit data bus.

0.6144 Gbps Line Rate:

	Sequence number on AUX interface 0	1	2	3
[31:24]	#Z.X.0.0 (1)	#Z.X.4.0	#Z.X.8.0	#Z.X.12.0
[23:16]	#Z.X.1.0	#Z.X.5.0	#Z.X.9.0	#Z.X.13.0
[15:8]	#Z.X.2.0	#Z.X.6.0	#Z.X.10.0	#Z.X.14.0
[7:0]	#Z.X.3.0	#Z.X.7.0	#Z.X.11.0	#Z.X.15.0

1.2288 Gbps Line Rate:

	Sequence number on AUX interface 0	1	2	...	7
[31:24]	#Z.X. 0.0 (1)	#Z.X.2.0	#Z.X.4.0	...	#Z.X.14.0
[23:16]	#Z.X. 0.1 (1)	#Z.X.2.1	#Z.X.4.1	...	#Z.X.14.1
[15:8]	#Z.X.1.0	#Z.X.3.0	#Z.X.5.0	...	#Z.X.15.0
[7:0]	#Z.X.1.1	#Z.X.3.1	#Z.X.5.1	...	#Z.X.15.1

2.4576 Gbps Line Rate:

	Sequence number on AUX interface 0	1	2	...	15
[31:24]	#Z.X.0.0 (1)	#Z.X.1.0	#Z.X.2.0	...	#Z.X.15.0
[23:16]	#Z.X.0.1 (1)	#Z.X.1.1	#Z.X.2.1	...	#Z.X.15.1
[15:8]	#Z.X.0.2 (1)	#Z.X.1.2	#Z.X.2.2	...	#Z.X.15.2
[7:0]	#Z.X.0.3 (1)	#Z.X.1.3	#Z.X.2.3	...	#Z.X.15.3

3.072 Gbps Line Rate:

	Sequence number on AUX interface 0	1	2	...	18	19
[31:24]	#Z.X.0.0 (1)	#Z.X.0.4 (1)	#Z.X.1.3	...	#Z.X.14.2	#Z.X.15.1
[23:16]	#Z.X.0.1 (1)	#Z.X.1.0	#Z.X.1.4	...	#Z.X.14.3	#Z.X.15.2
[15:8]	#Z.X.0.2 (1)	#Z.X.1.1	#Z.X.2.0	...	#Z.X.14.4	#Z.X.15.3
[7:0]	#Z.X.0.3 (1)	#Z.X.1.2	#Z.X.2.1	...	#Z.X.15.0	#Z.X.15.4

4.952 Gbps Line Rate:

	Sequence number on AUX interface					
	0	1	2	...	30	31
[31:24]	#Z.X.0.0(1)	#Z.X.0.4(1)	#Z.X.1.0	...	#Z.X.14.0	#Z.X.15.4
[23:16]	#Z.X.0.1(1)	#Z.X.0.5(1)	#Z.X.1.1	...	#Z.X.14.1	#Z.X.15.5
[15:8]	#Z.X.0.2(1)	#Z.X.0.6(1)	#Z.X.2.2	...	#Z.X.14.2	#Z.X.15.6
[7:0]	#Z.X.0.3(1)	#Z.X.0.7(1)	#Z.X.2.3	...	#Z.X.15.3	#Z.X.15.7

6.144 Gbps Line Rate:

	Sequence number on AUX interface					
	0	1	2	...	38	39
[31:24]	#Z.X.0.0(1)	#Z.X.0.4(1)	#Z.X.0.8(1)	...	#Z.X.15.2	#Z.X.15.6
[23:16]	#Z.X.0.1(1)	#Z.X.0.5(1)	#Z.X.0.9(1)	...	#Z.X.15.3	#Z.X.15.7
[15:8]	#Z.X.0.2(1)	#Z.X.0.6(1)	#Z.X.1.0	...	#Z.X.15.4	#Z.X.15.8
[7:0]	#Z.X.0.3(1)	#Z.X.0.7(1)	#Z.X.1.1	...	#Z.X.15.5	#Z.X.15.9

8.11008, 9.8304 Gbps Line Rate:

	Sequence number on AUX interface						
	0	1	2	3	...	62	63
[31:24]	#Z.X.0.0(1)	#Z.X.0.4(1)	#Z.X.0.8 (1)	#Z.X.0.12 (1)	...	#Z.X.15.8	#Z.X.15.12
[23:16]	#Z.X.0.1(1)	#Z.X.0.5(1)	#Z.X.0.9 (1)	#Z.X.0.13 (1)	...	#Z.X.15.9	#Z.X.15.13
[15:8]	#Z.X.0.2(1)	#Z.X.0.6(1)	#Z.X.0.10 (1)	#Z.X.0.14 (1)	...	#Z.X.15.10	#Z.X.15.14
[7:0]	#Z.X.0.3(1)	#Z.X.0.7(1)	#Z.X. 0.11(1)	#Z.X.0.15 (1)	...	#Z.X.15.11	#Z.X.15.15

10.1376 Gbps Line Rate:

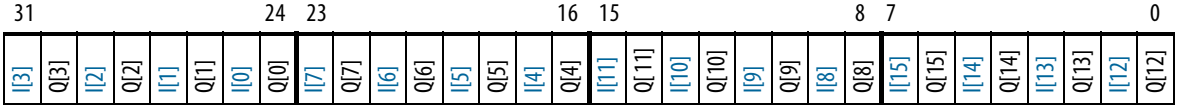
	Sequence number on AUX interface								
	0	1	2	3	4	5	...	62	79
[31:24]	#Z.X.0.0(1)	#Z.X.0.4(1)	#Z.X.0.8 (1)	#Z.X.0.12 (1)	#Z.X.0.16	#Z.X.1.0	...	#Z.X.15.12	#Z.X.15.16
[23:16]	#Z.X.0.1(1)	#Z.X.0.5(1)	#Z.X.0.9 (1)	#Z.X.0.13 (1)	#Z.X.0.17	#Z.X.1.1	...	#Z.X.15.13	#Z.X.15.17
[15:8]	#Z.X.0.2(1)	#Z.X.0.6(1)	#Z.X.0.10 (1)	#Z.X.0.14 (1)	#Z.X.0.18	#Z.X.1.2	...	#Z.X.15.14	#Z.X.15.18
[7:0]	#Z.X.0.3(1)	#Z.X.0.7(1)	#Z.X. 0.11(1)	#Z.X.0.15 (1)	#Z.X.0.19	#Z.X.1.3	...	#Z.X.15.15	#Z.X.15.19

Note:
(1) Light blue table cells indicate control word bytes. Yellow table cells indicate real-time vendor specific bytes. White table cells indicate data word bytes.

The CPRI IP core passes the incoming AUX data through to the CPRI link unmodified. You must ensure that the incoming AUX data bits already include any CRC values expected by the application at the other end of the CPRI link.

Figure 3-19: Data Sample Order on aux_tx_data and aux_rx_data Buses

Illustrates how CPRI frame data is ordered in each 32-bit word.



Direct IQ Interface

If you turn on **Enable direct IQ mapping interface** in the CPRI v6.0 parameter editor, the direct IQ interface is available. This interface allows direct access to the I/Q data time slots in the CPRI frame. You can connect this interface to any user-defined air standard I/Q mapping module.

This interface is Avalon-ST compliant with a read latency value of 1.

You can alter the transmit latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter.

Table 3-6: Direct IQ Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Direct IQ RX Interface		
Signal Name	**Direction**	**Description**
`iq_rx_valid[3:0]`	Output	Each asserted bit indicates the corresponding byte on the current `iq_rx_data` bus is valid I/Q data.
`iq_rx_data[31:0]`	Output	I/Q data received from the CPRI frame. The `iq_rx_valid` signal indicates which bytes are valid I/Q data bytes.
Direct IQ TX Interface		
Signal Name	**Direction**	**Description**
`iq_tx_ready[3:0]`	Output	Each asserted bit indicates the IP core is ready to read I/Q data from the corresponding byte of `iq_tx_data` on the next clock cycle.
`iq_tx_valid[3:0]`	Input	Write valid for `iq_tx_data`. Assert bit [n] to indicate that the corresponding byte on the current `iq_tx_data` bus is valid I/Q data.
`iq_tx_data[31:0]`	Input	I/Q data to be written to the CPRI frame. The IP core writes the individual bytes of the current value on the `iq_tx_data` bus to the CPRI frame based on the `iq_tx_ready` signal from the previous cycle, and the `iq_tx_valid` signal in the current cycle.

Figure 3-20: Direct IQ RX Interface Timing Diagram

Direct IQ RX interface behavior in a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_rx_x` and `aux_rx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `iq_rx_valid` output signal that you use to identify the clock cycles with valid I/Q data.

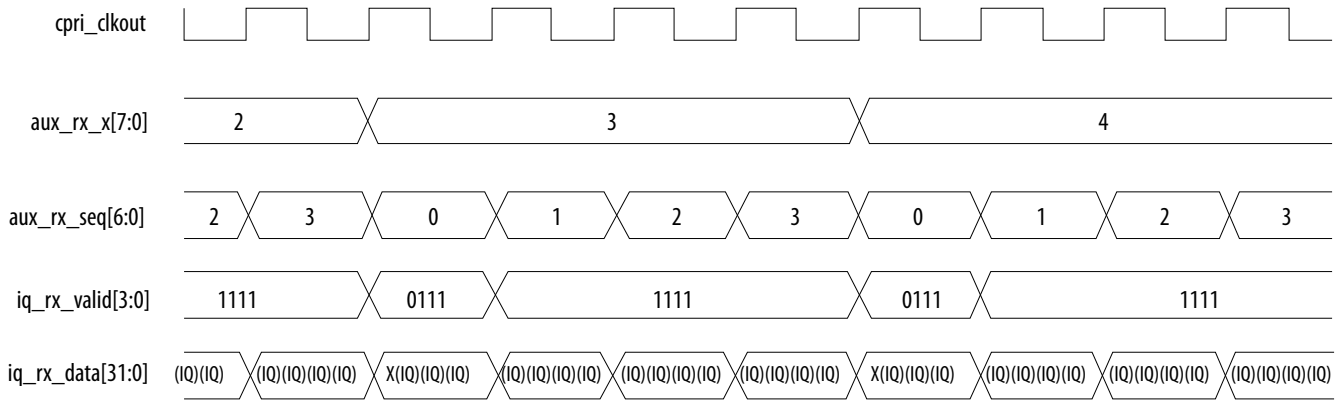
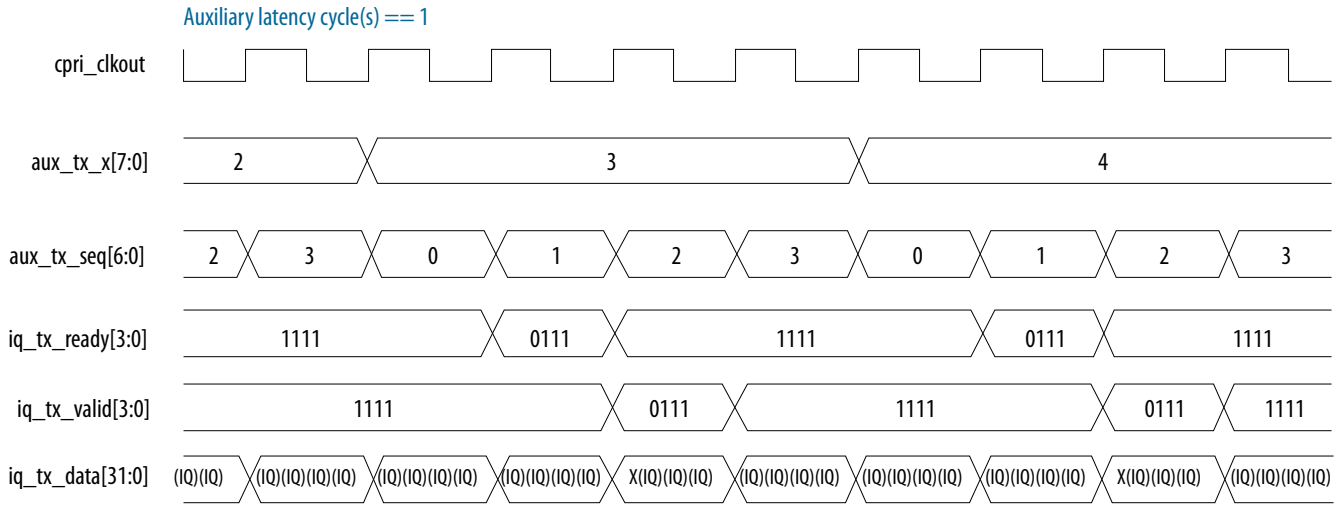


Figure 3-21: Direct IQ TX Interface Timing Diagram

Expected behavior on the direct IQ TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_tx_x` and `aux_tx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `iq_tx_ready` output signal that you use to identify the clock cycles when you can write I/Q data to the CPRI frame. Note that the write latency is two `cpri_clkout` clock cycles in this example.



Related Information

Avalon Interface Specifications

For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Streaming Interfaces* chapter.

Ctrl_AxC Interface

If you turn on **Enable direct ctrl_axc access interface** in the CPRI v6.0 parameter editor, the Ctrl_AxC interface is available. This interface allows direct access to the Ctrl_AxC subchannels in the CPRI frame, which are subchannels 4, 5, 6, and 7 in each hyperframe.

This interface is Avalon-ST compliant with a read latency value of 1.

You can alter the transmit latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter.

Table 3-7: Ctrl_AxC Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Ctrl_AxC RX Interface		
Signal Name	**Direction**	**Description**
`ctrl_axc_rx_valid[3:0]`	Output	Each asserted bit indicates the corresponding byte on the current `ctrl_axc_rx_data` bus is valid Ctrl_Axc data.
`ctrl_axc_rx_data[31:0]`	Output	Ctrl_Axc data received from the CPRI frame. The `ctrl_axc_rx_valid` signal indicates which bytes are valid Ctrl_Axc data bytes.
Ctrl_AxC TX Interface		
Signal Name	**Direction**	**Description**
`ctrl_axc_tx_ready[3:0]`	Output	Each asserted bit indicates the IP core is ready to read Ctrl_Axc data from the corresponding byte of `ctrl_axc_tx_data` on the next clock cycle.
`ctrl_axc_tx_valid[3:0]`	Input	Write valid for `ctrl_axc_tx_data`. Assert bit [n] to indicate that the corresponding byte on the current `ctrl_axc_tx_data` bus is valid Ctrl_Axc data.
`ctrl_axc_tx_data[31:0]`	Input	Ctrl_Axc data to be written to the CPRI frame. The IP core writes the individual bytes of the current value on the `ctrl_axc_tx_data` bus to the CPRI frame based on the `ctrl_axc_tx_ready` signal from the previous cycle, and the `ctrl_axc_tx_valid` signal in the current cycle.

Figure 3-22: Ctrl_Axc RX Interface Timing Diagram

Ctrl_Axc RX interface behavior in a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_rx_x` and `aux_rx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `ctrl_axc_rx_valid` output signal that you use to identify the clock cycles with valid Ctrl_Axc data.

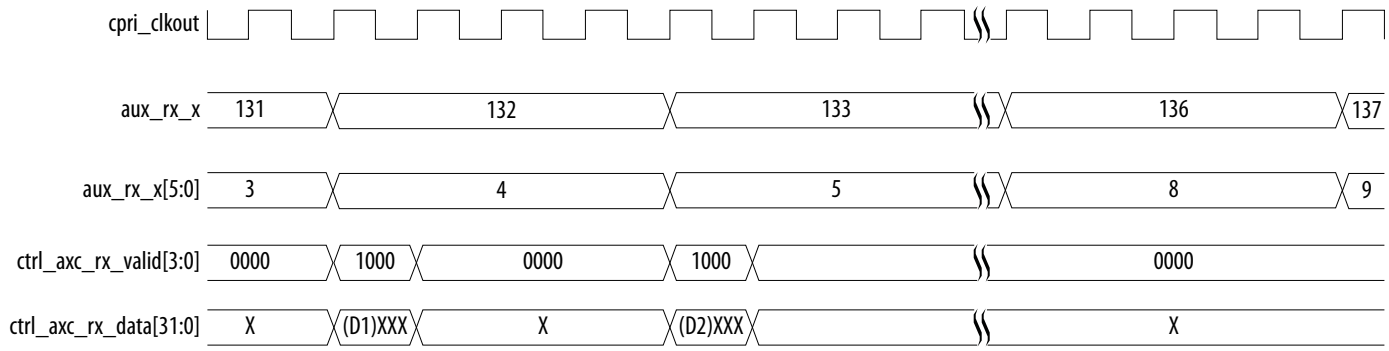
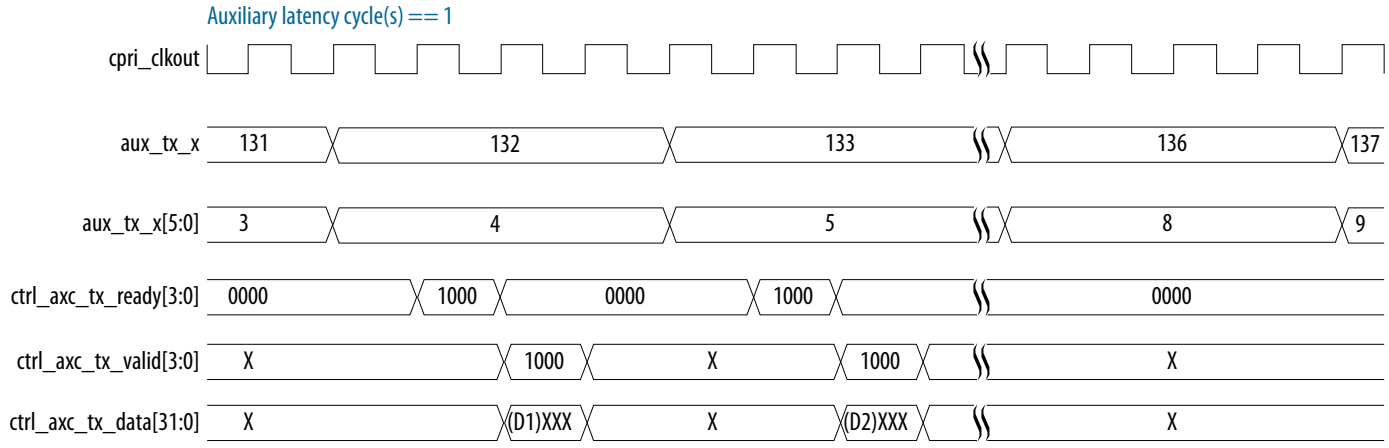


Figure 3-23: Ctrl_Axc TX Interface Timing Diagram

Expected behavior on the Ctrl_Axc TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_tx_x` and `aux_tx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `ctrl_axc_tx_ready` output signal that you use to identify the clock cycles when you can write Ctrl_Axc data to the CPRI frame. Note that the write latency is two `cpri_clkout` clock cycles in this example.



Related Information

Avalon Interface Specifications

For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Streaming Interfaces* chapter.

Direct Vendor Specific Access Interface

If you turn on **Enable direct vendor specific access interface** in the CPRI v6.0 parameter editor, the direct vendor specific access interface is available. This interface allows direct access to the Vendor Specific subchannels in the CPRI hyperframe. The Vendor Specific information is present only in subchannels 16 through (P-1) of the CPRI hyperframe, where P is the Fast C&M pointer value. Check the `vs_rx_valid` and `vs_tx_ready` signals to ensure you read and write this interface at the time that corresponds to the correct position in the CPRI frame. If you implement the AUX interface, you can read the value on the `aux_rx_x` or `aux_tx_x` output signal to identify the current position in the frame.

This interface is Avalon-ST compliant with a read latency value of 1.

You can alter the transmit latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter.

Table 3-8: Direct Vendor Specific Access Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Direct Vendor Specific RX Interface		
Signal Name	**Direction**	**Description**
`vs_rx_valid[3:0]`	Output	Each asserted bit indicates the corresponding byte on the current `vs_rx_data` bus is a valid vendor-specific byte.
`vs_rx_data[31:0]`	Output	Vendor-specific word received from the CPRI frame. The `vs_rx_valid` signal indicates which bytes are valid vendor-specific bytes.
Direct Vendor Specific TX Interface		
Signal Name	**Direction**	**Description**
`vs_tx_ready[3:0]`	Output	Each asserted bit indicates the IP core is ready to receive a vendor-specific byte from the corresponding byte of `vs_tx_data` on the next clock cycle.
`vs_tx_valid[3:0]`	Input	Write valid for `vs_tx_data`. Assert bit [n] of `vs_tx_valid` to indicate that byte [n] on the `vs_tx_data` bus holds a valid value in the current clock cycle.
`vs_tx_data[31:0]`	Input	Vendor-specific word to be written to the CPRI frame. The IP core writes the individual bytes of the current value on the `vs_tx_data` bus to the CPRI frame based on the `vs_tx_ready` signal from the previous cycle, and the `vs_tx_valid` signal in the current cycle.

Figure 3-24: Direct VS RX Timing Diagram

Direct VS RX interface behavior in a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_rx_x` signal is not part of this interface and is available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, its presence in the timing diagram explains the timing of the `vs_rx_valid` output signal that you use to identify the clock cycles with valid VS data.

The `aux_rx_x[7:0]` signal (labelled simply `aux_rx_x`) holds the eight-bit index of the basic frame in the hyperframe, from the perspective of the AUX interface. The subchannel index is the control word index modulo 64, available in `aux_rx_x[5:0]` if you turn on the AUX interface in your CPRI IP core.

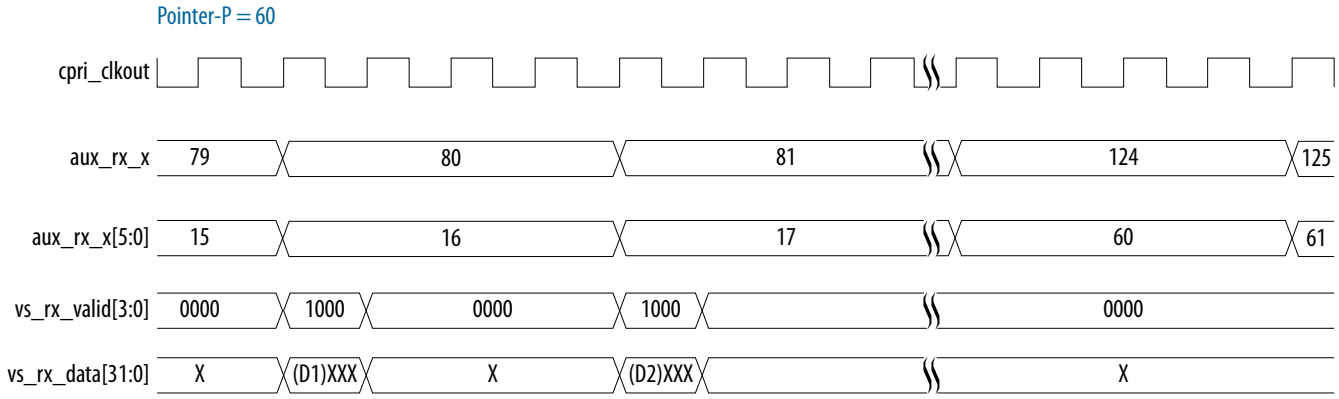


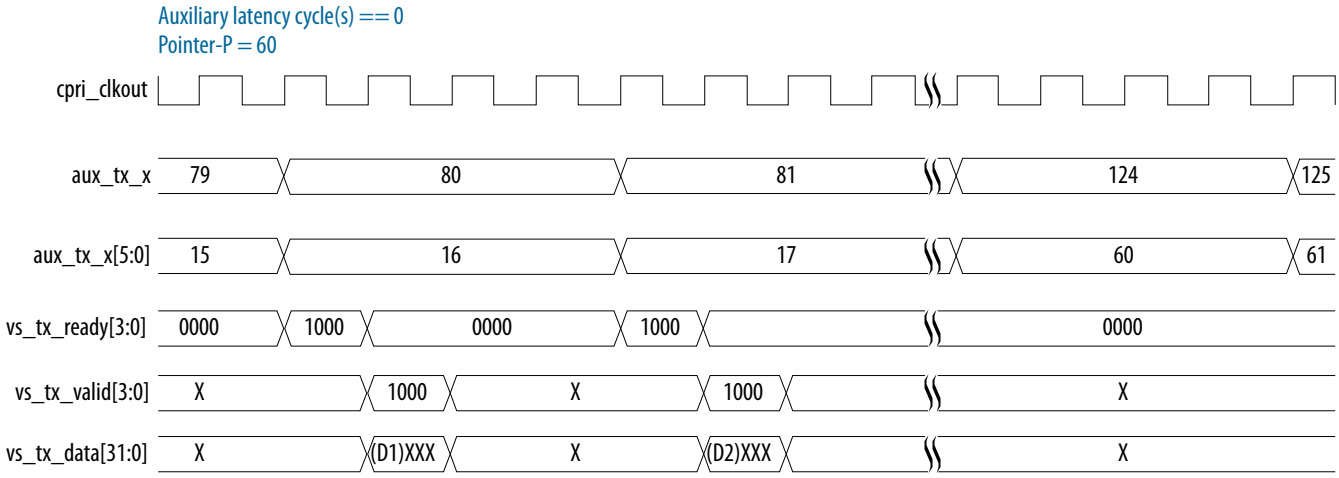
Figure 3-25: Direct VS TX Timing Diagram

Expected behavior on the direct VS TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps.

The `aux_tx_x` signal is not part of this interface and is available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, its presence in the timing diagram explains the timing of the `vs_tx_ready` output signal that you use to identify the clock cycles when you can write VS data to the CPRI frame.

The `aux_tx_x[7:0]` signal (labelled simply `aux_tx_x`) holds the eight-bit index of the basic frame in the hyperframe, from the perspective of the AUX interface. The subchannel index is the control word index modulo 64, available in `aux_tx_x[5:0]` if you turn on the AUX interface in your CPRI IP core.

Note that the write latency is one `cpri_clkout` clock cycle in this example.



Related Information

Avalon Interface Specifications

For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Streaming Interfaces* chapter.

Real-Time Vendor Specific Interface

If you turn on **Enable direct real-time vendor specific interface** in the CPRI v6.0 parameter editor, the real-time vendor specific interface is available. This interface allows direct access to the Real Time Vendor Specific words in the CPRI hyperframe. Check the `rtvs_rx_valid` and `rtvs_tx_ready` signals to ensure you read and write this interface at the time that corresponds to the correct position in the CPRI frame. If you implement the AUX interface, you can read the value on the `aux_rx_seq` or `aux_tx_seq` output signal to identify the current position in the frame.

This option is only available if you specify a CPRI line bit rate of 10.1376 Gbps for your IP core.

This interface is Avalon-ST compliant with a read latency value of 1.

You can alter the transmit write latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter.

Table 3-9: Real-Time Vendor Specific Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Real-Time Vendor Specific RX Interface		
Signal Name	**Direction**	**Description**
`rtvs_rx_valid`	Output	Each asserted bit indicates the corresponding byte on the current `rtvs_rx_data` bus is a valid real-time vendor-specific byte.
`rtvs_rx_data[31:0]`	Output	Real-time vendor-specific word received from the CPRI frame. The `rtvs_rx_valid` signal indicates which bytes are valid real-time vendor-specific bytes.
Real-Time Vendor Specific TX Interface		
Signal Name	**Direction**	**Description**
`rtvs_tx_ready`	Output	Indicates the IP core is ready to read a real-time vendor-specific byte from `rtvs_tx_data` on the next clock cycle.
`rtvs_tx_valid`	Input	Write valid for `rtvs_tx_data`. Assert this signal to indicate `rtvs_tx_data` holds a valid value in the current clock cycle.
`rtvs_tx_data[31:0]`	Input	Real-time vendor-specific word to be written to the CPRI frame. The IP core writes the current value of the `rtvs_tx_data` bus to the CPRI frame based on the `rtvs_tx_ready` signal from the previous cycle, and the `rtvs_tx_valid` signal in the current cycle.

Figure 3-26: Direct RTVS RX Timing Diagram

Direct RTVS RX interface behavior in a CPRI v6.0 IP core running at 10.1376 Gbps.

The `aux_rx_x` and `aux_rx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `rtvs_rx_valid` output signal that you use to identify the clock cycles with valid RTVS data.

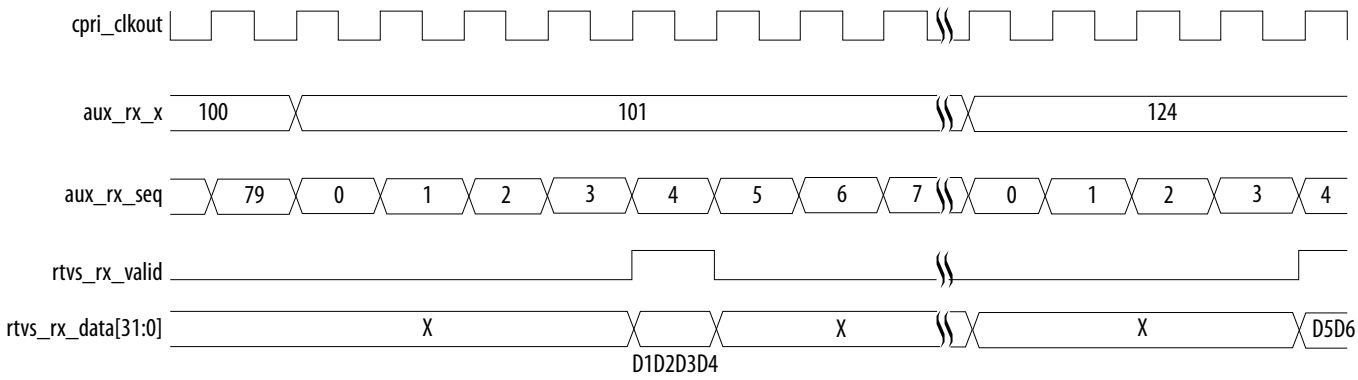
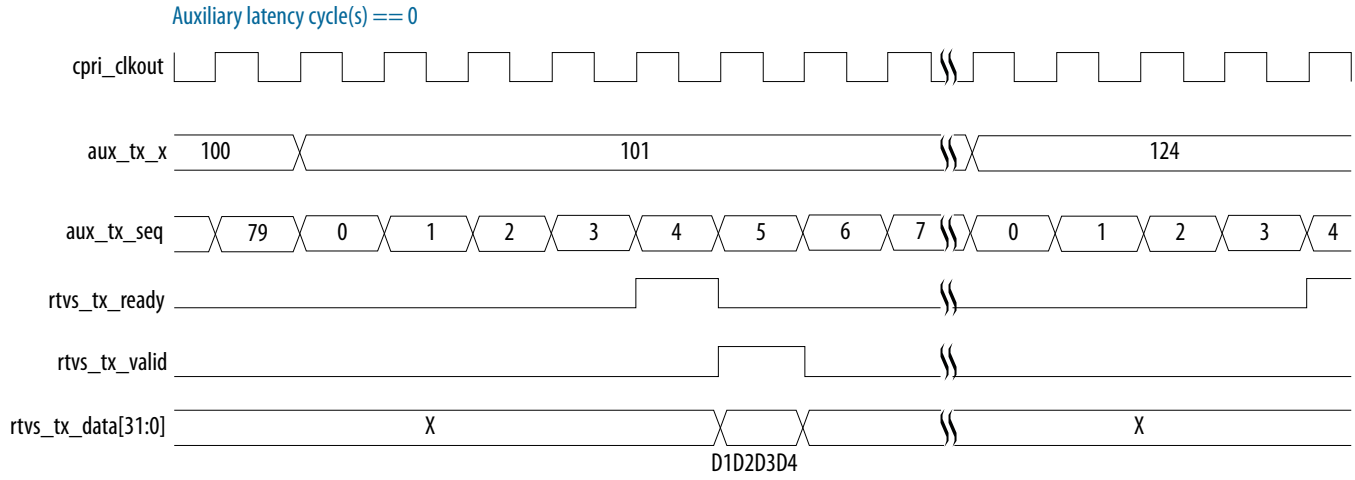


Figure 3-27: Direct RTVS TX Timing Diagram

Expected behavior on the direct RTVS TX interface of a CPRI v6.0 IP core running at 10.1376 Gbps.

The `aux_tx_x` and `aux_tx_seq` signals are not part of this interface and are available only if you turn on the AUX interface in your CPRI v6.0 IP core variation. However, their presence in the timing diagram explains the timing of the `rtvs_tx_ready` output signal that you use to identify the clock cycles when you can write RTVS data to the CPRI frame.

Note that the write latency is one `cpri_clkout` clock cycle in this example.



Related Information

Avalon Interface Specifications

For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Streaming Interfaces* chapter.

Direct HDLC Serial Interface

If you turn on **Enable direct HDLC serial interface** in the CPRI v6.0 parameter editor, the direct HDLC serial interface is available. This interface allows direct access to the slow control and management data in the CPRI frame. You can connect this interface to a user-defined HDLC PCS and MAC.

This interface is Avalon-ST compliant with a read latency value of 1.

You can alter the transmit write latency with the **Auxiliary and direct interfaces write latency cycle(s)** parameter. However, you do not need to view the `aux_tx_seq` signal for correct alignment. You can monitor the `hdlc_rx_valid` and `hdlc_tx_ready` signals to discover the correct times to read and write data on this interface.

Table 3-10: Direct HDLC Serial Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Direct HDLC Serial RX Interface		
Signal Name	**Direction**	**Description**
hdlc_rx_valid	Output	When asserted, indicates hdlc_rx_data holds a valid HDLC bit in the current clock cycle.
hdlc_rx_data	Output	HDLC data stream received from the CPRI frame. The hdlc_rx_valid signal indicates which bits are valid HDLC bytes.
Direct HDLC Serial TX Interface		
Signal Name	**Direction**	**Description**
hdlc_tx_ready	Output	When asserted, indicates the IP core is ready to receive HDLC data from hdlc_tx_data on the next clock cycle.
hdlc_tx_valid	Input	Write valid for hdlc_tx_data. Assert this signal to indicate that hdlc_tx_data holds a valid HDLC bit in the current clock cycle.
hdlc_tx_data	Input	HDLC data stream to be written to the CPRI frame directly. The IP core writes the current value on hdlc_tx_data to the CPRI frame based on the hdlc_tx_ready signal from the previous cycle, and the hdlc_tx_valid signal in the current cycle.

Figure 3-28: Direct HDLC Serial RX Timing Diagram

HDLC Serial RX interface behavior in a CPRI v6.0 IP core running at 0.6144 Gbps.

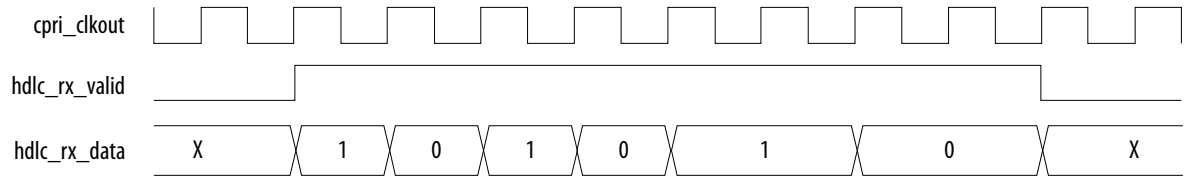
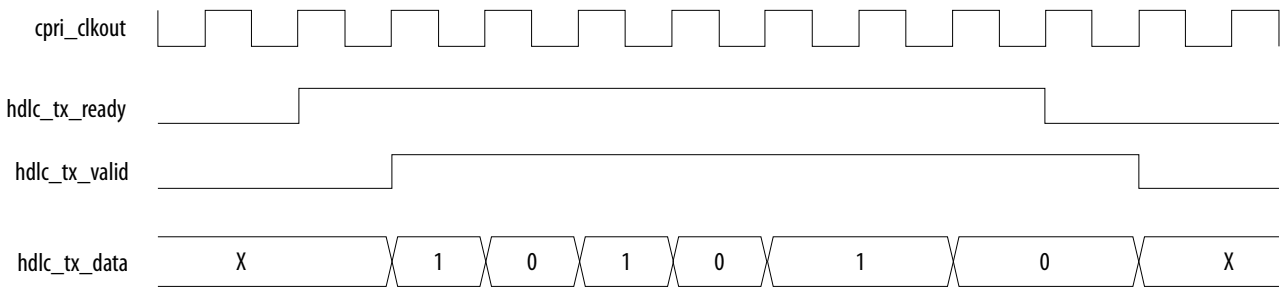


Figure 3-29: Direct HDLC Serial TX Timing Diagram

Expected behavior on the HDLC Serial TX interface of a CPRI v6.0 IP core running at 0.6144 Gbps.



Related Information

Avalon Interface Specifications

For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Streaming Interfaces* chapter.

Direct L1 Control and Status Interface

If you turn on **Enable direct Z.130.0 alarm bits access interface** in the CPRI v6.0 parameter editor, the direct L1 control and status interface is available. This interface provides direct access to the Z.130.0 alarm and reset signals (loss of frame (LOF), loss of signal (LOS), service access point (SAP) defect indication (SDI), remote alarm indication (RAI), and reset request or acknowledge) in the CPRI hyperframe.

If you connect the AUX interface of your RE slave IP core to a network switch or other routing layer rather than directly to the downstream RE master, or if you do not fully connect the AUX interface to the downstream RE master, you can use this Z.130.0 access interface to streamline the transfer of reset requests and SDI alarms across hops.

This interface has higher transmit priority than access through the CPRI v6.0 IP core registers.

This interface has the following types of signals:

- `_local_` signals are output signals from the IP core about the state of this IP core, and also indicate the IP core will assert the relevant outgoing Z.130.0 bit in the next CPRI hyperframe according to the transmit priority of this interface.
- `_assert` signals are input signals the application can use to request that the IP core assert the relevant outgoing Z.130.0 bit in the next CPRI hyperframe according to the transmit priority of this interface. You can also connect these signals in an RE master to the corresponding `_req` output signals of the upstream RE slave in a multi-hop configuration, to support efficient transfer of reset requests and SDI alarms to the IP core.
- `_remote` signals are output signals from the IP core that indicate the IP core received a Z.130.0 byte on the CPRI link with the relevant bit asserted by the CPRI link partner.
- `_req` signals are also output signals from the IP core that indicate the IP core received a Z.130.0 byte on the CPRI link with the relevant bit asserted by the CPRI link partner. However, these signals are intended to be passed downstream in a multi-hop configuration. If the IP core is an RE slave, and it connects to an RE master through the Z.130.0 alarm and reset interface in a multi-hop configuration, you can connect the RE slave IP core `_req` output signal directly to the corresponding `_assert` input signal on the downstream RE master for efficient communication of the reset request or SDI alarm.

Table 3-11: Direct L1 Control and Status Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Signal Name	Direction	Description
`z130_local_lof`	Output	Indicates the IP core has detected a local loss of frame. In this case, the `state_l1_synch` output signal indicates the L1 synchronization state machine is in state XACQ1 or XACQ2. In this case the IP core also asserts the `local_lof` bit in the `FLSAR` register at offset 0x2C.

Signal Name	Direction	Description
z130_local_los	Output	Indicates the IP core has detected a local loss of signal. The IP core asserts this flag if it detects excessive 8B/10B errors that trigger the assertion of the optional L1 debug rx_lcv output signal or the xcvr_los output signal and the rx_los field of the L1_CONFIG register. In this case the IP core also asserts the local_los bit in the FLSAR register at offset 0x2C.
z130_sdi_assert	Input	Indicates that the master service access point (SAP) is not available. Possible causes for this situation are equipment error or that the connected slave IP core is forwarding an SDI request it detected to the current RE CPRI master IP core through a direct connection.
z130_local_rai	Output	Indicates that either the z130_local_lof or the z130_local_los signal is high; clears when both of those two signals are low. Logical OR of two output signals z130_local_lof and z130_local_los.
z130_reset_assert	Input	Reset request from the application or from an RE slave to the current RE CPRI master IP core through a direct connection.
z130_remote_lof	Output	Indicates LOF received in Z.130.0 control byte from remote CPRI link partner. In this case the IP core also asserts the remote_lof bit in the FLSAR register at offset 0x2C.
z130_remote_los	Output	Indicates LOS received in Z.130.0 control byte from remote CPRI link partner. In this case the IP core also asserts the remote_los bit in the FLSAR register at offset 0x2C.
z130_sdi_req	Output	Indicates remote SAP defect indication received in Z.130.0 control byte from remote CPRI link master. If the current CPRI IP core is an RE slave in a multi-hop configuration, you should connect this output signal directly to the z130_sdi_assert input signal of the downstream RE master.
z130_remote_rai	Output	Asserts when either z130_remote_lof or z130_remote_los is asserted, and clears when both z130_remote_lof and z130_remote_los have the value of 0. In this case the IP core also asserts the rai_detected bit in the FLSAR register at offset 0x2C.

Signal Name	Direction	Description
z130_reset_req	Output	If the current IP core is a CPRI link slave, indicates the IP core received a reset request in the Z.130.0 control byte from the remote CPRI link master. If the current IP core is a CPRI link master, indicates the IP core received a reset acknowledgement in the Z.130.0 control byte from the remote CPRI link slave.

Figure 3-30: sdi_assert to sdi_req on Direct L1 Control and Status Interface

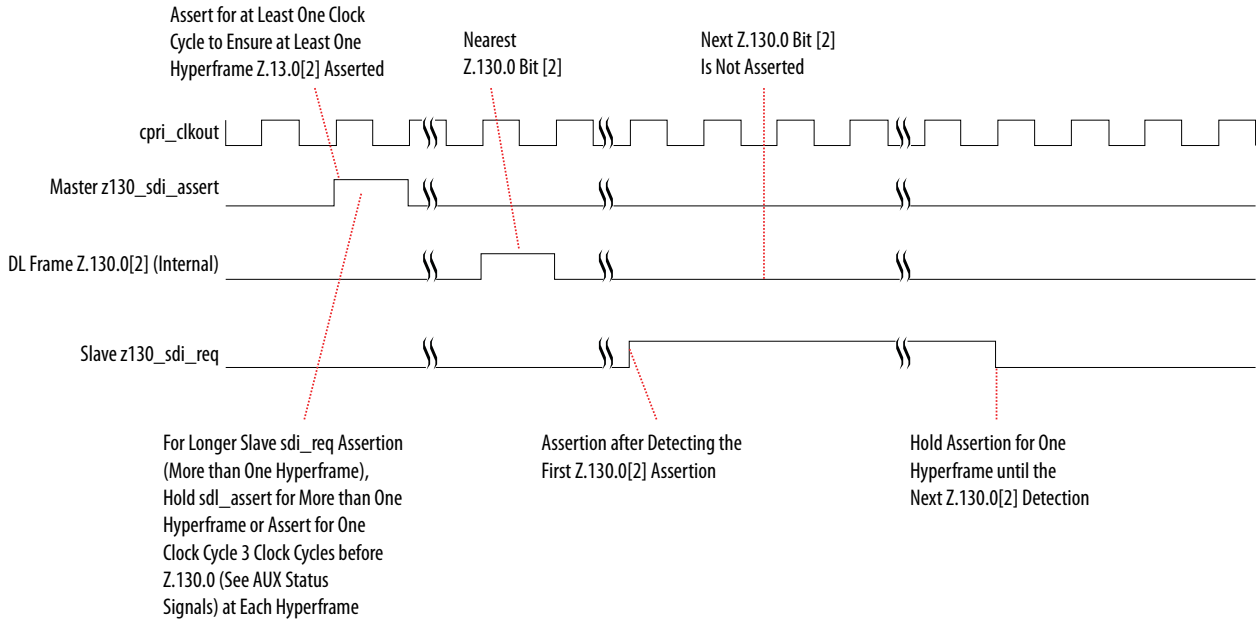


Figure 3-31: reset_assert to reset_req on Direct L1 Control and Status Interface

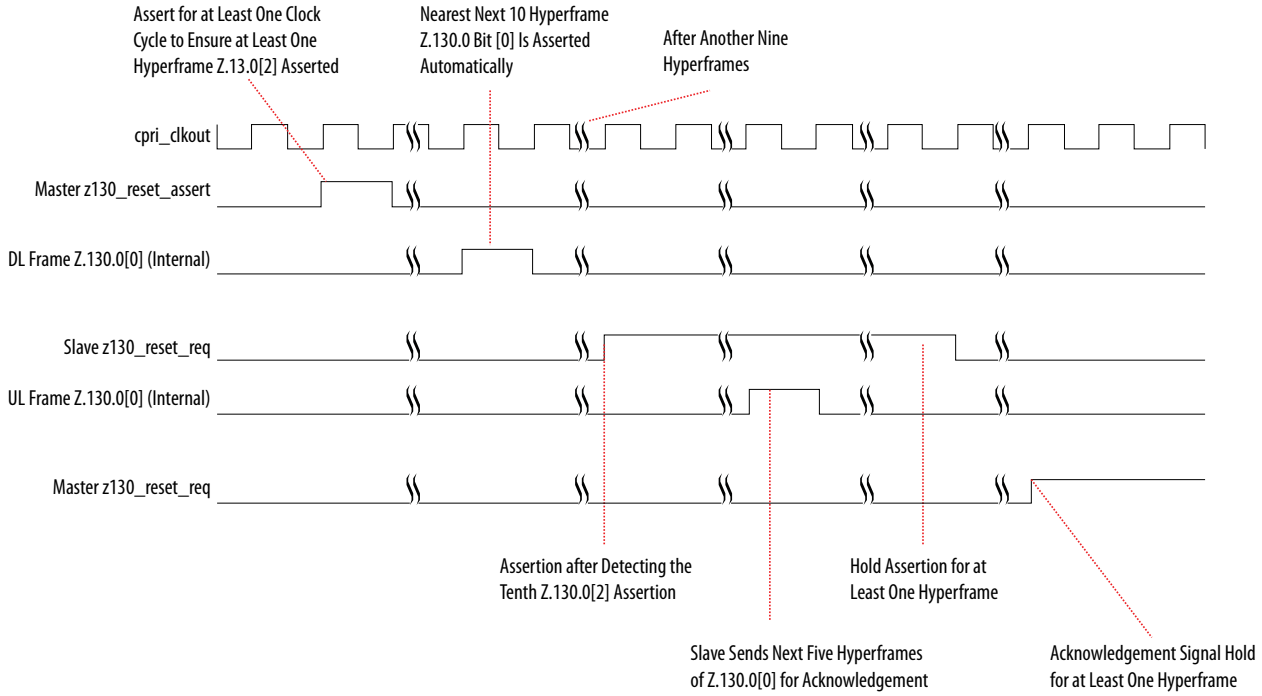
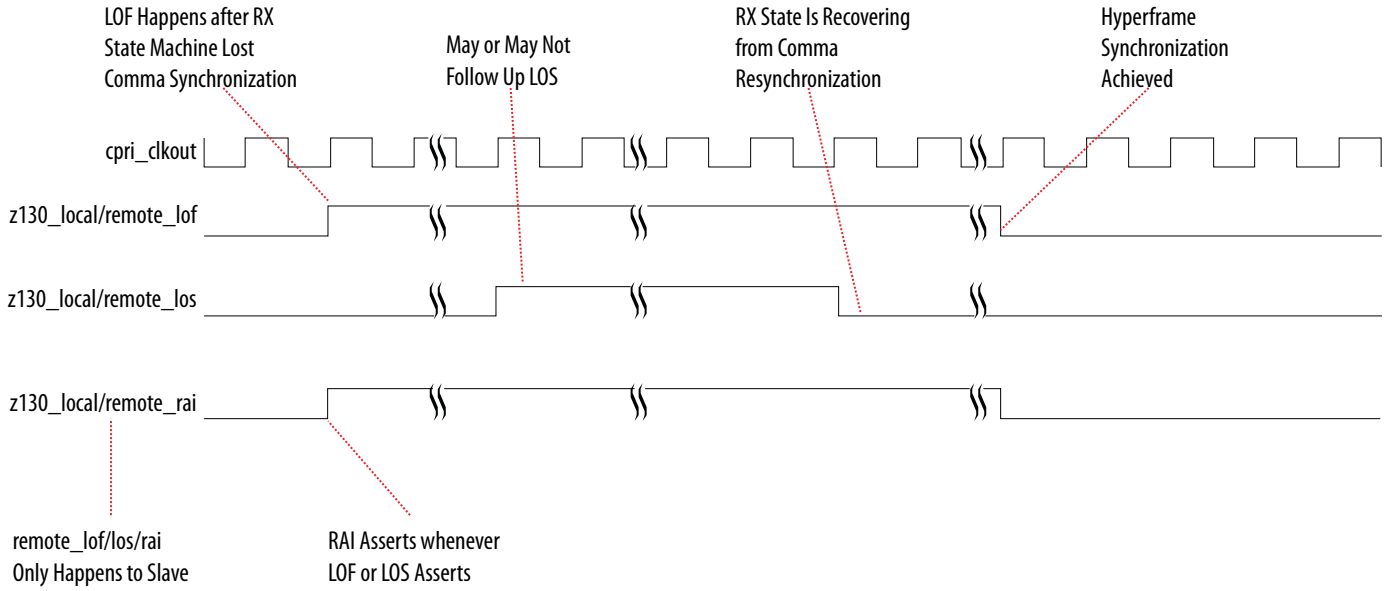


Figure 3-32: LOF, LOS, and RAI on Direct L1 Control and Status Interface



Related Information

FLSAR Register on page 5-11

L1 Debug Interface

If you turn on **Enable L1 debug interfaces** in the CPRI v6.0 parameter editor, the L1 debug interface is available.

Table 3-12: Direct L1 Control and Status Interface Signals

All of the L1 debug signals are asynchronous.

Signal Name	Direction	Description
rx_lcv	Output	Indicates the IP core has detected excessive 8B10B errors received. The IP core asserts this signal when it detects more than 15 bits of error.
rx_freq_alarm	Output	Indicates the CPRI receive clock (receiver CDR recovered clock) and the main IP core clock (cpri_clkout) have a PPM difference. The IP core asserts this alarm each time it detects a mismatch.

Media Independent Interface (MII) to External Ethernet Block

The media independent interface (MII) allows the CPRI v6.0 IP core to communicate directly with an external Ethernet MAC block. If you set the value of the **Ethernet PCS interface** parameter in the CPRI v6.0 parameter editor to **MII**, your IP core includes this interface.

The MII supports the bandwidth described in the CPRI v6.0 Specification in Table 12, Achievable Ethernet bit rates.

Table 3-13: MII Signals

These signals are available if you set the value of the **Ethernet PCS interface** parameter in the CPRI v6.0 parameter editor to **MII**. You can connect a user-defined Ethernet MAC to this interface.

The interface is fully compliant to the IEEE 802.3 100BASE-X 100Mbps MII specification. An Ethernet PCS block in the CPRI v6.0 IP core ensures the interface bandwidth matches the current CPRI line bit rate and accesses data at the correct CPRI frame positions according to the Z.194.0 pointer value.

You must monitor the MII FIFO status signals and ensure you do not overflow or underflow the FIFO.

The interface signals are clocked by the mii_rxclk or mii_txclk clock.

RX MII Signals		
Signal Name	**Direction**	**Description**
mii_rxclk	Input	Clocks the MII receiver interface. You must drive this clock at the frequency of 25 MHz to achieve the 100 Mbps bandwidth required for this interface.
mii_rxreset_n	Input	Resets the MII receiver interface and FIFO read logic. This reset signal is active low.

RX MII Signals		
Signal Name	**Direction**	**Description**
mii_rxdv	Output	Ethernet receive data valid. Indicates the presence of valid data or initial K nibble on mii_rxd[3:0].
mii_rxer	Output	Ethernet receive error. Indicates an error in the current nibble of mii_rxd. This signal is de-asserted at reset, and remains de-asserted while the CPRI v6.0 IP core is resetting and until link initialization completes.
mii_rxd[3:0]	Output	Ethernet receive nibble data. Data bus for data from the CPRI v6.0 IP core to the external Ethernet block. All bits are de-asserted during reset, and all bits are asserted after reset until the CPRI v6.0 IP core achieves frame synchronization.
TX MII Signals		
Signal Name	**Direction**	**Description**
mii_txclk	Input	Clocks the MII transmitter interface. You must drive this clock at the frequency of 25 MHz to achieve the 100 Mbps bandwidth required for this interface.
mii_txreset_n	Input	Resets the MII transmitter interface and FIFO write logic. This signal is active low.
mii_txen	Input	Valid signal from the external Ethernet block, indicating the presence of valid data on mii_txd[3:0]. The external Ethernet block must also assert this signal two cycles before initial valid data, while the IP core inserts /J/ and /K/ nibbles in the data stream to form the start-of-packet symbol.
mii_txer	Input	Ethernet transmit coding error. When this signal is asserted, the CPRI v6.0 IP core inserts an Ethernet HALT symbol in the data it passes to the CPRI link.
mii_txd[3:0]	Input	Ethernet transmit nibble data. The data transmitted from the external Ethernet block to the CPRI v6.0 IP core, for transmission on the CPRI link. This input bus is synchronous to the rising edge of the mii_txclk clock.
MII Status Signals		
Signal Name	**Direction**	**Description**
mii_tx_fifo_status[3:0]	Output	Ethernet Tx PCS FIFO fill level status. The individual bits have the following meanings: • Bit [3]: Empty • Bit [2]: Almost empty • Bit [1]: Full • Bit [0]: Almost full

MII Status Signals		
Signal Name	**Direction**	**Description**
`mii_rx_fifo_status[3:0]`	Output	Ethernet Rx PCS FIFO fill level status. The individual bits have the following meanings: • Bit [3]: Empty • Bit [2]: Almost empty • Bit [1]: Full • Bit [0]: Almost full

Figure 3-33: RX MII Timing Diagram

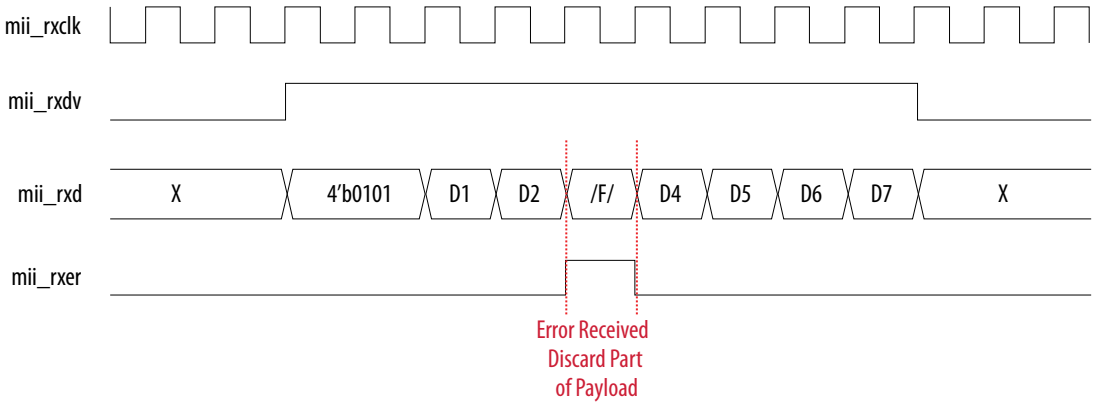
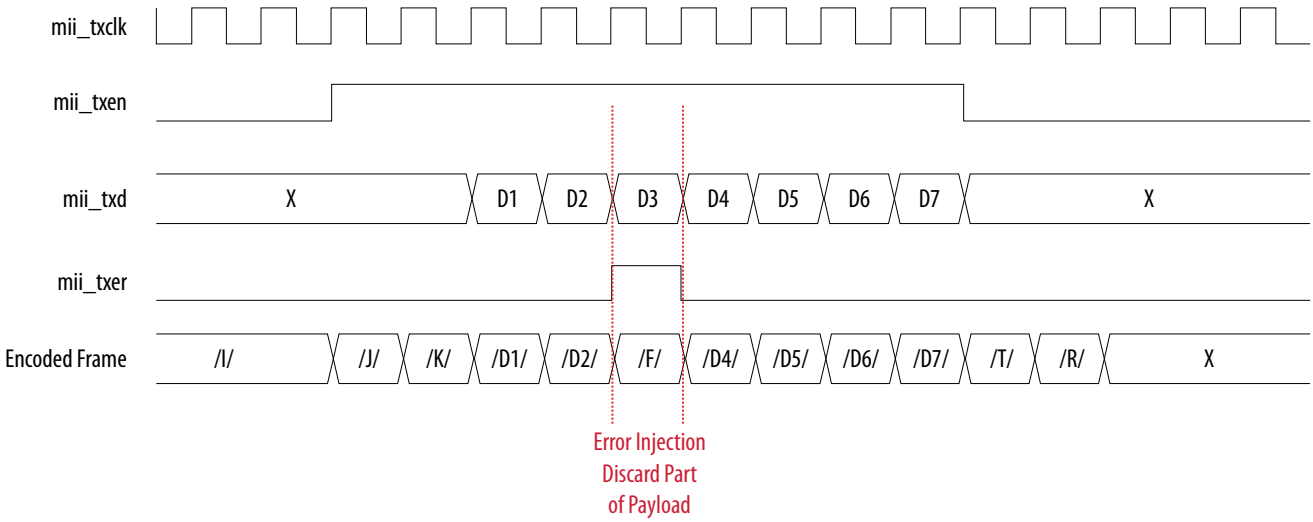


Figure 3-34: TX MII Timing Diagram



Related Information

CPRI v6.0 IP Core L2 Interface on page 4-1

Gigabit Media Independent Interface (GMII) to External Ethernet Block

The gigabit media independent interface (GMII) allows the CPRI v6.0 IP core to communicate directly with an external Ethernet MAC block. If you set the value of the **Ethernet PCS interface** parameter in the CPRI v6.0 parameter editor to **GMII**, your IP core includes this interface.

The GMII supports the bandwidth described in the CPRI v6.0 Specification in Table 12, Achievable Ethernet bit rates.

Table 3-14: GMII Signals

These signals are available if you set the value of the **Ethernet PCS interface** parameter in the CPRI v6.0 parameter editor to **GMII**. You can connect a user-defined Ethernet MAC to this interface.

The interface is fully compliant to the IEEE 802.3 1000BASE-X 1Gbps MII specification. An Ethernet PCS block in the CPRI v6.0 IP core ensures the interface bandwidth matches the current CPRI line bit rate and accesses data at the correct CPRI frame positions according to the Z.194.0 pointer value.

You must monitor the GMII FIFO status signals and ensure you do not overflow or underflow the FIFO.

The interface signals are clocked by the `gmii_rxclk` or `gmii_txclk` clock.

RX GMII Signals		
Signal Name	**Direction**	**Description**
`gmii_rxclk`	Input	Clocks the GMII receiver interface. You must drive this clock at the frequency of 125 MHz.
`gmii_rxreset_n`	Input	Resets the GMII receiver interface and FIFO read logic. This reset signal is active low.
`gmii_rxdv`	Output	Ethernet receive data valid. Indicates the presence of valid data or initial start-of-packet control character on `gmii_rxd[7:0]`.
`gmii_rxer`	Output	Ethernet receive error. Indicates an error on `gmii_rxd`. When this signal is asserted, the value on gmii_rxd[7:0] is 0x0E.
`gmii_rxd[7:0]`	Output	Ethernet receive data. Data bus for data from the CPRI v6.0 IP core to the external Ethernet block. All bits are de-asserted during reset, and all bits are asserted after reset until the CPRI v6.0 IP core achieves frame synchronization.
TX GMII Signals		
Signal Name	**Direction**	**Description**
`gmii_txclk`	Input	Clocks the GMII transmitter interface. You must drive this clock at the frequency of 125 MHz.
`gmii_txreset_n`	Input	Resets the GMII transmitter interface and FIFO write logic. This signal is active low.

TX GMII Signals		
Signal Name	**Direction**	**Description**
`gmii_txen`	Input	Valid signal from the external Ethernet block, indicating the presence of valid data on `gmii_txd[7:0]`. This signal must be asserted two cycles before data is actually valid. This advance notice provides time for the CPRI v6.0 GMII transmitter block to insert an S character in the data stream to form the start-of-packet symbol. Deasserting this signal triggers the IP core to insert T and R characters in the data stream to form the end-of-packet symbol.
`gmii_txer`	Input	Ethernet transmit coding error. When this signal is asserted, the CPRI v6.0 IP core inserts an Ethernet Error Propogation symbol /V/ in the data it passes to the CPRI link.
`gmii_txd[7:0]`	Input	Ethernet transmit data. The data transmitted from the external Ethernet block to the CPRI v6.0 IP core, for transmission on the CPRI link. This input bus is synchronous to the rising edge of the `gmii_txclk` clock.
MII Status Signals		
Signal Name	**Direction**	**Description**
`gmii_txfifo_status[3:0]`	Output	Ethernet Tx PCS FIFO fill level status. The individual bits have the following meanings: • Bit [3]: Empty • Bit [2]: Almost empty • Bit [1]: Full • Bit [0]: Almost full
`gmii_rxfifo_status[3:0]`	Output	Ethernet Rx PCS FIFO fill level status. The individual bits have the following meanings: • Bit [3]: Empty • Bit [2]: Almost empty • Bit [1]: Full • Bit [0]: Almost full

Figure 3-35: RX GMII Timing Diagram

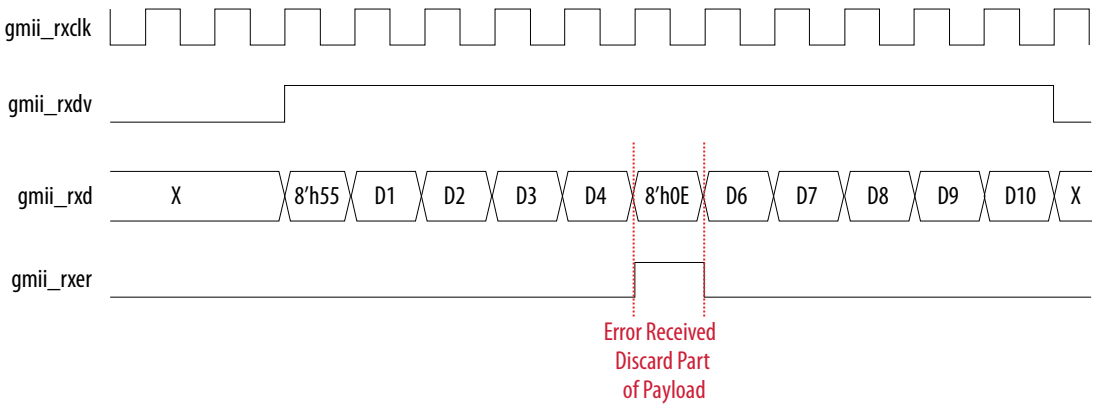
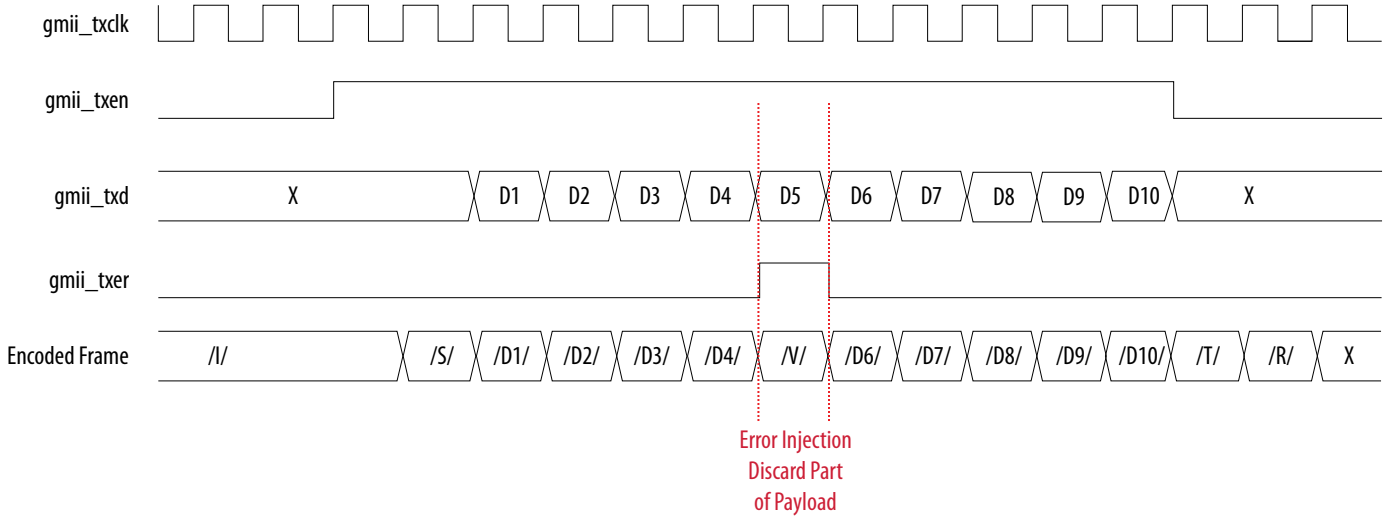


Figure 3-36: TX GMII Timing Diagram



Related Information

CPRI v6.0 IP Core L2 Interface on page 4-1

CPU Interface to CPRI v6.0 IP Core Registers

Use the CPU interface to access the CPRI v6.0 IP core status and configuration registers. This interface does not provide access to the hard transceiver configuration registers on the Intel Arria 10 or Intel Stratix 10 device.

If you turn on **Enable all control word access via management interface** in the CPRI v6.0 parameter editor, you can access all CPRI hyperframe control words through this interface.

The control and status interface is an Avalon-MM slave interface. Depending on the value you specify for **Avalon-MM interface addressing type** in the CPRI v6.0 parameter editor, the interface implements word

addressing or byte addressing. If you specify word addressing, you must connect other design components correctly to the interface to ensure the Avalon-MM byte addresses appear on the CPRI v6.0 IP core CPU interface as word addresses.

An on-chip processor such as the Nios II processor, or an external processor, can access the CPRI v6.0 configuration address space using this Avalon-MM interface.

Related Information

Avalon Interface Specifications
For more information about the Avalon-MM protocol, including timing diagrams, refer to the *Avalon Memory-Mapped Interfaces* chapter.

CPU Interface Signals

Table 3-15: CPRI v6.0 IP Core CPU Interface Signals

The CPRI v6.0 IP core CPU interface has the following features:

- Avalon-MM slave interface compliant.
- Provides support for single cycle read and write operations: you can read or write a single register in a single access operation.
- Supports a single `cpu_clk` clock cycle read latency and a zero `cpu_clk` clock cycle write latency for most registers.

Signal Name	Direction	Description
`cpu_clk`	Input	Clocks the signals on the CPRI v6.0 CPU interface. Supports any frequency that the device fabric supports.
`cpu_reset_n`	Input	Active low reset signal. Resets the CPRI v6.0 CPU interface and all of the registers to which it provides access. You should hold this signal asserted for one full `cpu_clk` cycle to ensure it is captured by the IP core.
`cpu_address[15:0]`	Input	Address for reads and writes. All CPRI v6.0 control and status registers are 32 bits wide. By default, this address is a word address (addresses a 4-byte (32-bit) word), not a byte address. However, if you set **Avalon-MM interface addressing type** to **Byte** in the CPRI v6.0 parameter editor, this address is a byte address.
`cpu_byteenable[3:0]`	Input	Data-byte enable signal
`cpu_read`	Input	You must assert this signal to request a read transfer
`cpu_write`	Input	You must assert this signal to request a write transfer
`cpu_writedata[31:0]`	Input	Write data
`cpu_readdata[31:0]`	Output	Read data

Signal Name	Direction	Description
cpu_waitrequest	Output	Indicates that the control and status interface is busy executing an operation. When the IP core deasserts this signal, the operation is complete and the read data is valid.
cpu_irq	Output	Interrupt request. All interrupts that you enable in the relevant register fields, assert this interrupt signal when they are triggered. You must check the relevant register fields to determine the cause or causes of the interrupt assertion.

Figure 3-37: Read Transaction on CPU Interface

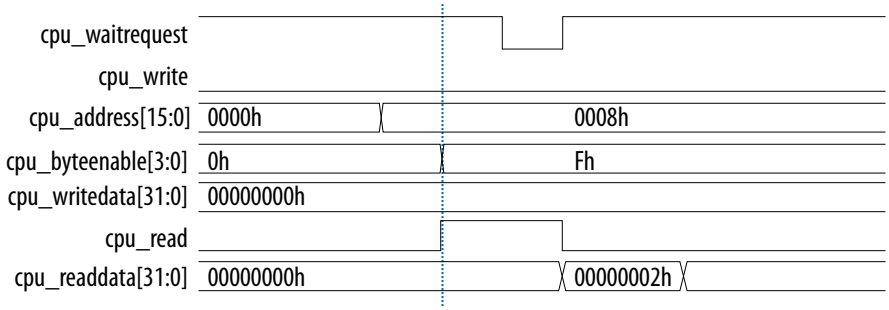
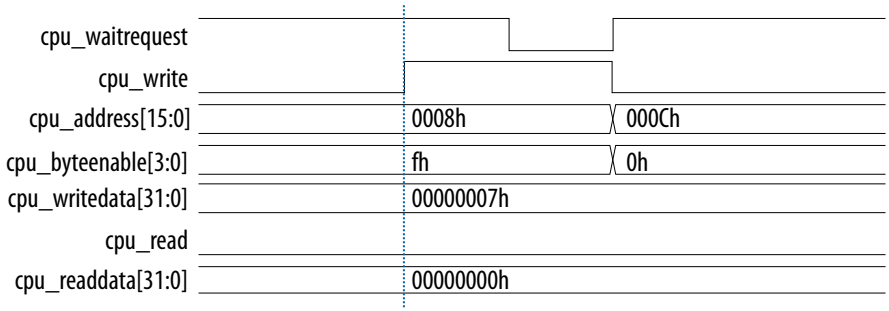


Figure 3-38: Write Transaction on CPU Interface



Related Information

Avalon Interface Specifications

For more information about the Avalon-MM protocol, refer to the *Avalon Memory-Mapped Interfaces* chapter.

Accessing the Hyperframe Control Words

When you turn on **Enable all control word access via management interface** in the CPRI v6.0 parameter editor, you can access the 256 control words in a hyperframe through the CPRI v6.0 IP core CPU interface. The `CTRL_INDEX` register and the `RX_CTRL` register support your application in reading the incoming control words, and the `L1_CONFIG` register, `CTRL_INDEX` register, and `TX_CTRL` register support the application in writing to outgoing control words.

Register support provides you access to the full control word. Alternatively, in timing-critical applications, you can access the full control words through the CPRI v6.0 IP core AUX interface or other dedicated direct interfaces.

Note: Intel recommends that you use the CPU interface to access the hyperframe control words only in applications that are not timing-critical.

Specifying the Control Word

Figure 3-39: Subchannels in a Hyperframe

Illustrates how the 256 control words in the hyperframe are organized as 64 subchannels of four control words each. The figure illustratres why the index X of a control word is Ns + 64 + Xs, where Ns is the subchannel index and Xs is the index of the control word within the subchannel.

	Xs == 0	1	2	3
Ns == 0	0: K28.5	Synchronization and Timing		
1	1: HDLC link	65: HDLC	129: HDLC	193: HDLC
2	2: L1 In-band	66: L1 in-band	130: L1 in-band	194: P (20 = 0x14)
3	3: Reserved	67: Reserved		
4	4: Ctrl_AxC	...	...	...
	...			
7	7: Ctrl_AxC	71: Ctrl_AxC	135: Ctrl_AxC	199: Ctrl_AxC
	...			
14	14: Reserved			
15	15: Reserved	79: Reserved	143: Reserved	207: Reserved
16	Vendor-specific			
	...			
19				
20 Pointer P --->	20: Ethernet			
	...			
62	62	126	190	254
63	63	127	191	255

The `rx_ctrl_x` and `tx_ctrl_x` fields of the `CTRL_INDEX` register hold the X value of the control word you want to access through the control and status interface.

Specifying the Position in the Control Word

You can access only 32 bits in a single register access. Depending on the CPRI line bit rate, a control word may have multiple 32-bit sections. Therefore, in addition to specifying the control word location in the CPRI frame, you must also specify a 32-bit aligned position in the control word.

Table 3-16: Control Word Byte Positions in RX_CTRL and TX_CTRL Registers

In this table, each control word nibble is indicated with 0xF. The presence of 0xF or 0x0 indicates whether the nibble within the register is populated with a valid control word nibble.

CPRI Bit Rate (Gbps)	Register Access Sequence Number ({rx,tx}_ctrl_seq)				
	0 (first access)	1 (2nd access)	2 (3rd access)	3 4th access)	4 (5th access)
0.6144	FF000000	0	0	0	0
1.2288	FFFF0000	0	0	0	0
2.4576	FFFFFFFF	0	0	0	0
3.072	FFFFFFFF	FF000000	0	0	0
4.9152	FFFFFFFF	FFFFFFFF	0	0	0
6.144	FFFFFFFF	FFFFFFFF	FFFF0000	0	0
8.11008, 9.8034	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	0
10.1376	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	FF000000

However, the table does not clarify which control word byte occupies which position in the register. The following examples indicate the correspondence between register bytes and control word bytes:

- At the CPRI line bit rate of 0.6144 Gbps, when you access hyperframe control word X, the 8-bit control word from hyperframe position #Z.X.0 is in bits [31:24] of the register.
- At the CPRI line bit rate of 1.2288 Gbps, the byte from position #Z.X.0.0 is in bits [31:24] of the register and the byte from position #Z.X.0.1 is in bits [23:16] of the register.
- At the CPRI line bit rate of 3.072 Gbps, you must access the register twice to retrieve or write the full control word. In the first access operation, you access the 32 bits of the control word in positions #Z.X.0.0 (in register bits [31:24]), #Z.X.0.1 (in register bits [23:16]), #Z.X.0.2 (in register bits [15:8]), and #Z.X.0.3 (in register bits [7:0]). In the second access operation, you access the eight bits of the control word in position #Z.X.0.4 in bits [31:24] of the register.

Retrieving the Hyperframe Control Words

A control receive table contains one entry for each of the 256 control words in the current hyperframe. To read a control word, your application must write the control word number X to the `rx_ctrl_x` field of the `CTRL_INDEX` register and then read the last received #Z.X control word from the `RX_CTRL` register. Because the register can hold only 32 bits at a time, depending on the CPRI line bit rate, reading the full control word may require multiple register accesses. Increment the value in the `rx_ctrl_seq` field of the `CTRL_INDEX` register from zero to four to access the full control word when the CPRI line bit rate is 10.1376 Gbps, or from zero to two when the CPRi line bit rate is 6.144 Gbps, for example.

Example 3-1: Control Word Retrieval Example

To retrieve the vendor-specific portion of a control word in the most recent received hyperframe, perform the following steps:

1. Identify the indices for the vendor-specific portion of the transmit control table, using the formula X = Ns + 64 + Xs.

In the example, Ns = 16 and Xs = 0, 1, 2, and 3. Therefore, the indices to be read are 16, 80, 144, and 208.

2. For each value X in 16, 80, 144, and 208, perform the following steps:
   a. Write the value X to the `rx_ctrl_x` field of the `CTRL_INDEX` register.
   b. Reset the `rx_ctrl_seq` field of the `CTRL_INDEX` register to the value of zero.
   c. In the following `cpu_clk` cycle, read the first 32-bit section of the control word from the `RX_CTRL` register.
   d. If the CPRI line bit rate is greater than 2.4576 Gbps, increment the `rx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 1 and in the following `cpu_clk` cycle, read the second 32-bit section of the #Z.X control word from the `RX_CTRL` register.
   e. If the CPRI line bit rate is greater than 4.9152 Gbps, increment the `rx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 2 and in the following `cpu_clk` cycle, read the third 32-bit section of the #Z.X control word from the `RX_CTRL` register.
   f. If the CPRI line bit rate is greater than 6.144 Gbps, increment the `rx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 3 and in the following `cpu_clk` cycle, read the fourth 32-bit section of the #Z.X control word from the `RX_CTRL` register.
   g. If the CPRI line bit rate is 10.1376 Gbps, increment the `rx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 4 and in the following `cpu_clk` cycle, read the fifth 32-bit section of the #Z.X control word (the real-time vendor specific bytes) from the `RX_CTRL` register.

Writing the Hyperframe Control Words

A control transmit table contains one entry for each of the 256 control words in the current hyperframe. Each control transmit table entry contains a control word and an enable bit. As the frame is created, if a control word entry is enabled, and the global `tx_ctrl_insert_en` bit in the `L1_CONFIG` register is set, the IP core writes the appropriate control transmit table entry to the CPRI frame's control word.

You write to a control transmit table entry through the `TX_CTRL` register. This register access method requires that you write the control word in 32-bit sections. Use the `tx_ctrl_seq` field of the `CTRL_INDEX` register to specify the 32-bit section you are currently writing to the `TX_CTRL` register.

To write a control word in the control transmit table, perform the following steps:

1. Write the control word number X to the `tx_ctrl_x` field of the `CTRL_INDEX` register.
2. Reset the `tx_ctrl_seq` field of the `CTRL_INDEX` register to the value of zero.
3. Write the first 32-bit section of the next intended #Z.X control word to the `TX_CTRL` register.
4. If the CPRI line bit rate is greater than 2.4576 Gbps, increment the `tx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 1 and write the second 32-bit section of the next intended #Z.X control word to the `TX_CTRL` register.
5. If the CPRI line bit rate is greater than 4.9152 Gbps, increment the `tx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 2 and write the third 32-bit section of the next intended #Z.X control word to the `TX_CTRL` register.
6. If the CPRI line bit rate is greater than 6.144 Gbps, increment the `tx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 3 and write the fourth 32-bit section of the next intended #Z.X control word to the `TX_CTRL` register.

7. If the CPRI line bit rate is 10.1376 Gbps, increment the `tx_ctrl_seq` field of the `CTRL_INDEX` register to the value of 4 and write the fifth 32-bit section of the next intended #Z.X control word (the real-time vendor specific bytes) to the `TX_CTRL` register.
8. Set the `tx_ctrl_insert` bit of the `CTRL_INDEX` register to the value of 1.
9. After you update the control transmit table, set the `tx_ctrl_insert_en` bit of the `L1_CONFIG` register to enable the CPRI v6.0 IP core to write the values from the control transmit table to the control words in the outgoing CPRI frame.

The `tx_control_insert` bit of the `CTRL_INDEX` register enables or disables the transmission of the corresponding control transmit table entry in the CPRI frame. The `tx_ctrl_insert_en` bit of the `L1_CONFIG` register is the master enable: when it is set, the CPRI v6.0 IP core writes all table entries with the `tx_ctrl_insert` bit set into the CPRI frame.

Example 3-2: Control Word Transmission Example

To write the vendor-specific portion of the control word in a transmitted hyperframe, perform the following steps:

1. Identify the indices for the vendor-specific portion of the transmit control table, using the formula X = Ns + 64 + Xs.

   In the example, Ns = 16 and Xs = 0, 1, 2, and 3. Therefore, the indices to be read are 16, 80, 144, and 208.
2. For each value X in 16, 80, 144, and 208, perform the sequence of steps listed above.

   After you update the control transmit table with the control bytes, to insert the data in the next outgoing CPRI frame, make sure that you set the`tx_ctrl_insert_en` bit of the `L1_CONFIG` register to the value of 1 as specified in the instructions.

Auto-Rate Negotiation

If you turn on **Enable line bit rate auto-negotiation** in the CPRI v6.0 parameter editor, the auto-rate negotiation control and status interface is available. The CPRI v6.0 IP core provides support for dynamically changing the CPRI line bit rate, but requires that you implement user logic to control the auto-rate negotiation process. You control the process through the auto-rate negotiation control and status interface or the `BIT_RATE_CONFIG` register at offset 0x0C.

Table 3-17: Auto-Rate Negotiation Control and Status Interface Signals

All interface signals are clocked by the `cpri_clkout` clock.

Signal Name	Direction	Description
nego_bitrate_in[4:0]	Input	CPRI line bit rate to be used in next attempt to achieve frame synchronization, encoded according to the following valid values: • 5'b00001: 0.6144 Gbps • 5'b00010: 1.2288 Gbps • 5'b00100: 2.4576 Gbps • 5'b00101: 3.0720 Gbps • 5'b01000: 4.9150 Gbps • 5'b01010: 6.1440 Gbps • 5'b01100: 8.11008 Gbps • 5'b10000: 9.8304 Gbps • 5'b10100: 10.1376 Gbps This signal has higher priority than the bit_rate field in the BIT_RATE_CONFIG register at offset 0x0C. When this signal has the value of 5'b00000, the CPRI v6.0 IP core responds to the register field.
nego_bitrate_out[4:0]	Output	Reflects the current actual CPRI line bit rate.

Related Information

- **BIT_RATE_CONFIG Register** on page 5-5
- **Altera wiki CPRI v6.0 IP core information**
  Includes links to a reference design that implements auto-rate negotiation.

Extended Delay Measurement

The CPRI v6.0 IP core employs an additional mechanism to measure the delay through the IP core FIFOs to your desired precision. Separate dedicated clocks support this measurement for the RX and TX internal buffers in all variations and for the hard FIFOs present only in Intel Stratix 10 variations.

Related Information

- **TX_EX_DELAY Register** on page 5-18
- **RX_EX_DELAY Register** on page 5-18
- **Extended Delay Measurement Interface** on page 3-59

Extended Delay Measurement for Soft Internal Buffers

The CPRI v6.0 IP core uses a dedicated clock, ex_delay_clk, to measure the delay through the RX and TX internal buffers to your desired precision. The extended delay process is identical for the two directions of flow through the IP core; the TX_EX_DELAY and RX_EX_DELAY registers hold the same information for the two directions.

The tx_msrm_period field of the TX_EX_DELAY register contains the value N, such that N clock periods of the ex_delay_clk clock are equal to some whole number M of cpri_clkout periods. For example, N may be a multiple of M, or the M/N frequency ratio may be slightly greater than 1, such as 64/63 or 128/127.

The application layer specifies N to ensure the accuracy your application requires. The accuracy of the Tx buffer delay measurement is N/*least_common_multiple*(N,M) cpri_clkout periods.

Similarly, the rx_msrm_period field of the RX_EX_DELAY register contains the value N, such that N clock periods of the ex_delay_clk clock are equal to some whole number M of cpri_clkout periods.

If your application does not require this precision, drive the ex_delay_clk input port with the cpri_clkout signal. In this case, the M/N ratio is 1 because the frequencies are the same.

The tx_buf_delay field of the TX_DELAY register indicates the number of 32-bit words currently in the Tx buffer. After you program the tx_msrm_period field of the TX_EX_DELAY register with the value of N, the tx_ex_delay field of the TX_EX_DELAY register holds the current measured delay through the Tx buffer. The unit of measurement is cpri_clkout periods. The tx_ex_delay_valid field indicates that a new measurement has been written to the tx_ex_delay field since the previous register read. The following sections explain how you set and use these register values to derive the extended Tx delay measurement information.

M/N Ratio Selection

As your selected M/N ratio approaches 1, the accuracy provided by the extended delay measurement increases.

Table 3-18: Resolution as a Function of M/N Ratio at 3.072 Gbps

M	N	cpri_clkout Period	ex_delay_clk Period	Resolution
128	127	13.02 ns (1/76.80 MHz)	13.12 ns	±100 ps
64	63		13.22 ns	±200 ps
1	4		3.25 ns	±3.25 ns

Example 3-3: Extended Delay Measurement Calculation Example

This section walks you through an example that shows you how to calculate the frequency at which to run ex_delay_clk, and how to program and use the registers to determine the delay through the CPRI Receive Buffer.

For example, assume your CPRI v6.0 IP core runs at CPRI line bit rate 3.072 Gbps. In this case, the cpri_clkout frequency is 76.80, so a cpri_clkout cycle is 1/76.80 MHz.

If your accuracy resolution requirements are satisfied by an M/N ratio of 128/127, perform the following steps:

1. Program the value N=127 in the rx_msrm_period field of the RX_EX_DELAY register at offset 0x54.
2. Perform the following calculation to determine the ex_delay_clk frequency that supports your desired accuracy resolution:

   ex_delay_clk period = (M/N) cpri_clkout period = (128/127)(1/(76.80 MHz)= 13.123356 ns.

   Based on this calculation, the frequency of ex_delay_clk is 1/(13.123356 ns)

   The following steps assume that you run ex_delay_clk at this frequency.
3. Read the value of the RX_EX_DELAY register at offset 0x54.

If the rx_ex_delay_valid field of the register is set to 1, the value in the rx_ex_delay field has been updated, and you can use it in the following calculations. For this example, assume the value read from the rx_ex_delay field is 0x107D, which is decimal 4221.

4. Perform the following calculation to determine the delay through the Rx buffer:

   Delay through Rx buffer = (rx_ex_delay x cpri_clkout period) / N = (4221 x 13.02083 ns) / 127 = 432.7632 ns.

   This delay comprises (432.7632ns / 13 .02083 ns) = 33.236 cpri_clkout clock cycles.

These numbers provide you the result for this particular example. For illustration, the preceding calculation shows the result in nanoseconds. You can derive the result in cpri_clkout clock cycles by dividing the preceding result by the cpri_clkout clock period. Alternatively, you can calculate the number of cpri_clkout clock cycles of delay through the Rx buffer directly, as rx_msrm_period/N.

Extended Delay Measurement for Intel Stratix 10 Hard FIFOs

The CPRI v6.0 IP core uses a dedicated clock, latency_sclk, to measure the delay through the RX and TX Intel Stratix 10 device hard FIFOs that are configured in the CPRI v6.0 IP core.

The delay calculation process is identical for the two directions of flow through the IP core; the XCVR_TX_FIFO_DELAY and XCVR_RX_FIFO_DELAY registers hold the same information for the two directions.

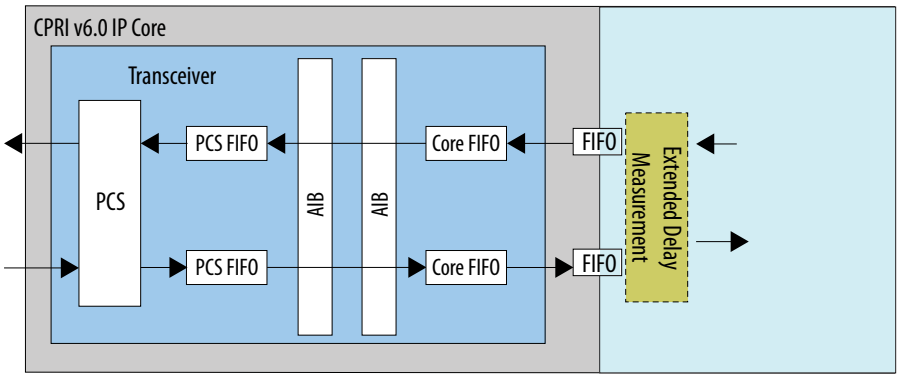
To measure the current Tx delay through the hard FIFOs:

- Check the tx_pcs_fifo_delay_valid and tx_core_fifo_delay_valid fields of the XCVR_TX_FIFO_DELAY register at offset 0x84 to ensure the delay count values are updated.
- Add the delay count values in the tx_pcs_fifo_delay and tx_core_fifo_delay fields of the XCVR_TX_FIFO_DELAY register at offset 0x84.
- Multiply the result by the clock period of the latency_sclk.
- Divide this result by 128.

To measure the current Rx delay through the hard FIFOs:

- Check the rx_pcs_fifo_delay_valid and rx_core_fifo_delay_valid fields of the XCVR_RX_FIFO_DELAY register at offset 0x88 to ensure the delay count values are updated.
- Add the delay count values in the rx_pcs_fifo_delay and rx_core_fifo_delay fields of the XCVR_RX_FIFO_DELAY register at offset 0x88.
- Multiply the result by the clock period of the latency_sclk.
- Divide this result by 128.

Figure 3-40: Additional Delay Through Stratix 10 Hard FIFOs



Extended Delay Measurement Interface

Table 3-19: Extended Delay Measurement Interface Signals

Signal Name	Direction	Description
`ex_delay_clk`	Input	Clock for extended delay measurement.
`ex_delay_reset_n`	Input	Resets the extended delay measurement block. This signal is active low. This reset signal is associated with the `ex_delay_clk` clock.
`latency_sclk`	Input	Clock for extended delay measurement of Intel Stratix 10 hard FIFOs. You can (but need not) drive this clock at the same frequency as `ex_delay_clk`.
`latency_sreset_n`	Input	Resets the extended delay measurement soft logic for the Intel Stratix 10 hard FIFOs. This signal is active low. This reset signal is associated with the `latency_sclk` clock.

Deterministic Latency and Delay Measurement and Calibration

The CPRI v6.0 IP core complies with CPRI V6.0 Specification requirements R-19, R-20, R-20A, R-21, and R-21A.

Delay Measurement and Calibration Features

The CPRI V6.0 Specification measurement and delay requirements support system configuration and correct synchronization.

The CPRI v6.0 IP core provides the following support for accurate delay measurement:

- Provides current Rx delay measurement values in the `RX_DELAY` and `RX_EX_DELAY` registers.
- Provides current Tx delay calibration values in the `XCVR_BITSLIP` register.
- Provides current round-trip delay measurement value in the `ROUND_TRIP_DELAY` register.
- Supports user control over delay measurement accuracy in the `TX_EX_DELAY` and `RX_EX_DELAY` registers.
- If you turn on **Enable round-trip delay calibration** in the CPRI v6.0 parameter editor, supports round-trip delay calibration.
- If you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor, supports single-trip delay calibration.

Delay Requirements

CPRI V6.0 Specification requirements R-17, R-18, and R-18A address jitter and frequency accuracy in the RE core clock for radio transmission. The relevant clock synchronization is performed using an external clean-up PLL that is not included in the CPRI v6.0 IP core.

CPR V6.0 Specification requirement R-20A addresses the maximum allowed delay in switching between receiving and transmitting on the radio interface. The radio interface is implemented outside the IP core based on raw data presented on the AUX interface or other direct interfaces. Because the IP core provides duplex communication on these interfaces, no delay calculation is required.

Requirement R-19 specifies that the link delay accuracy for the downlink between the synchronization master SAP and the synchronization slave SAP, excluding the cable length, be within ±8.138 ns. Requirements R-20 and R-21 extrapolate this requirement to single-hop round-trip delay accuracy. R-20 requires that the accuracy of the round-trip delay, excluding cables, be within ±16.276 ns, and R-21 requires that the round-trip cable delay measurement accuracy be within the same range. Requirement R-21A extrapolates this requirement further, to multihop round-trip delay accuracy. In calculating these delays, Intel assumes that the downlink and uplink cable delays have the same duration.

Figure 3-41: Single-Hop CPRI v6.0 IP Core Configuration Delay Measurement Reference Points

The round-trip cable delay is the sum of the T12 and T34 delays. The two delays are assumed to have the same duration.

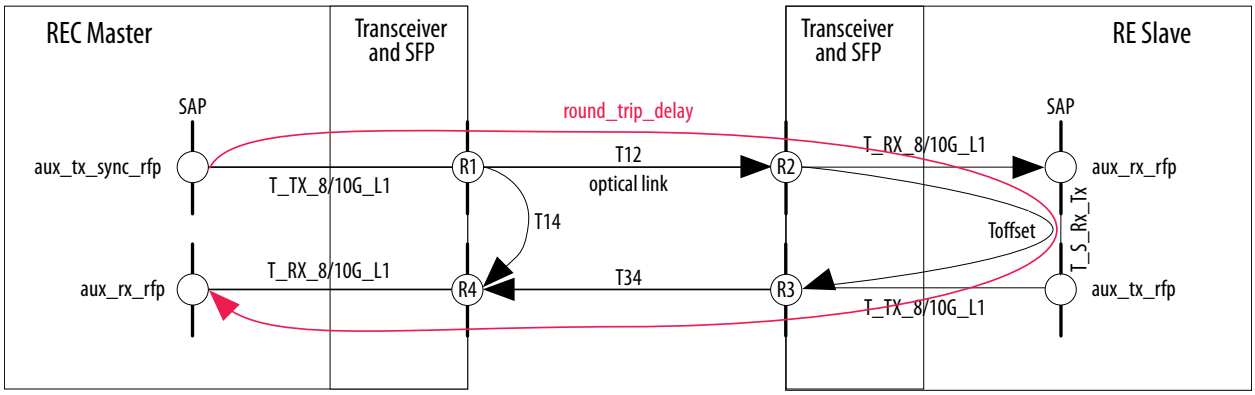
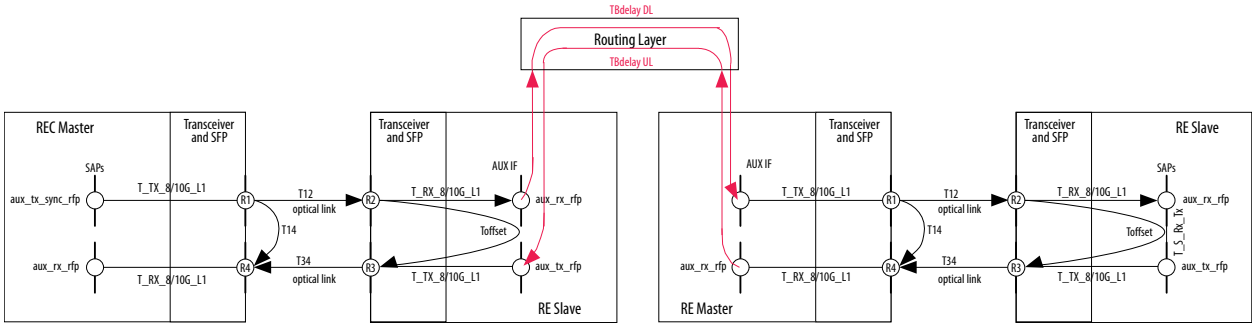


Figure 3-42: Multihop CPRI v6.0 IP Core Configuration Delay Measurement Reference Points

The duration of TBdelay depends on your routing layer implementation.



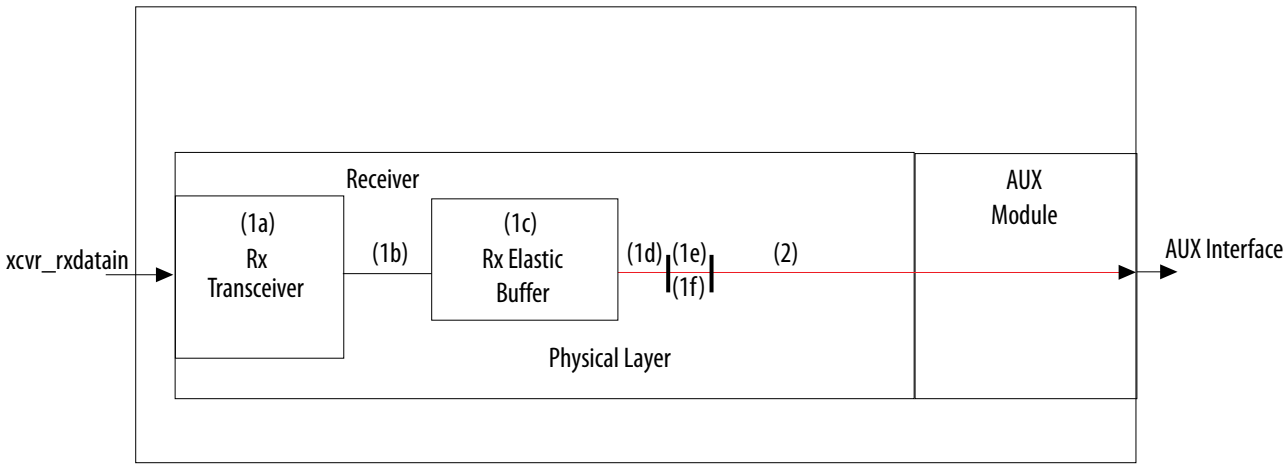
Single-Hop Delay Measurement

The Rx path delay and Tx path delay through the CPRI v6.0 IP core are the main components of the round-trip delay through a single-hop system.

Rx Path Delay

The Rx path delay is the cumulative delay from the arrival of the first bit of a 10 ms radio frame on the CPRI Rx interface to the start of transmission of the radio frame on the AUX interface.

Figure 3-43: Rx Path Delay to AUX Output in CPRI v6.0 IP Core



The Rx path delay to the AUX interface is the sum of the following delays:

1. The link delay is the delay between the arrival of the first bit of a 10 ms radio frame on the CPRI Rx interface and the CPRI v6.0 IP core internal transmission of the radio frame pulse from the CPRI protocol interface receiver.
   a. Rx transceiver latency is a fixed delay through the deterministic latency path of the Rx transceiver. Its duration depends on the device family and the current CPRI line bit rate. This delay includes comma alignment and byte alignment within the transceiver.
   b. Fixed delay from the Rx transceiver to the Rx elastic buffer. This delay depends on the device family and the CPRI line bit rate.

      Note: In IP core variations that target an Intel Stratix 10 device, you must also add the delay through the Stratix 10 hard FIFOs in the Rx path.
   c. Delay through the clock synchronization FIFO, as well as the phase difference between the recovered receive clock and the core clock `cpri_clkout`. The "Extended Delay Measurement" section shows how to calculate the delay in the CPRI v6.0 IP core Rx elastic buffer, which includes the phase difference delay.
   d. Byte alignment delay that can occur as data is shifted out of the receiver. This variable delay appears in the `rx_byte_delay` field of the `RX_DELAY` register. When the value in `rx_byte_delay` is non-zero, a byte alignment delay of one `cpri_clkout` cycle occurs in the Rx path.
   e. Variable delay introduced by the optional single-trip delay calibration feature in CPRI link slave IP cores.
   f. Variable delay introduced by the optional round-trip delay calibration feature in CPRI link master IP cores.
2. Delay from the CPRI low-level receiver block to the AUX interface. This delay depends on the device family and the CPRI line bit rate.

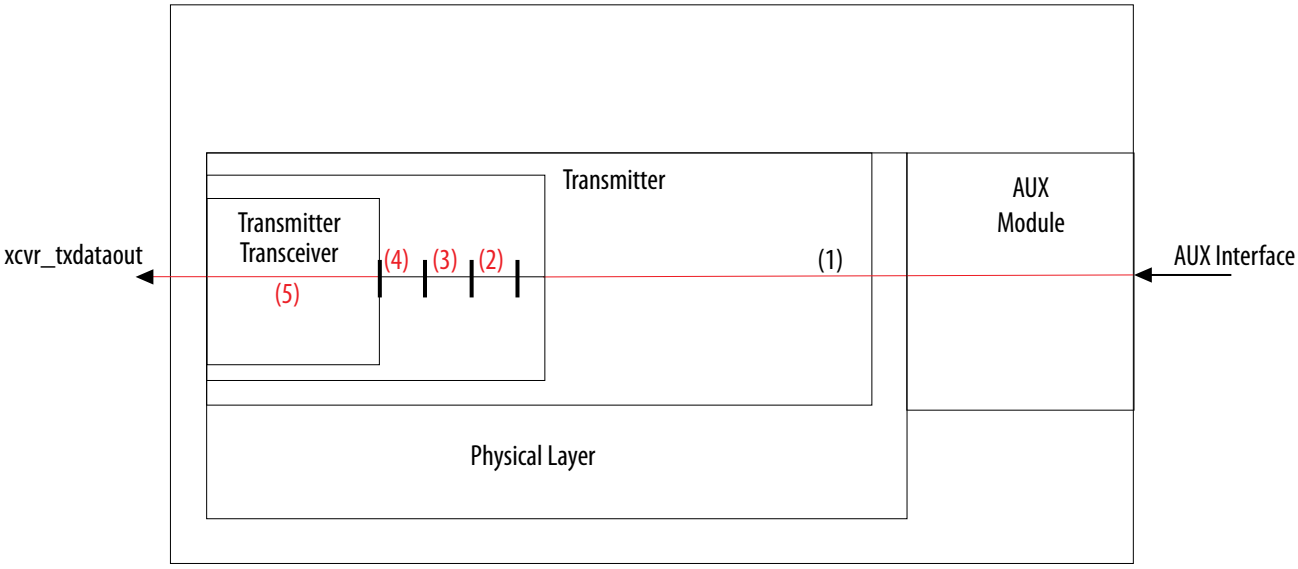
Related Information

RX_DELAY Register on page 5-17

Tx Path Delay

The Tx path delay is the cumulative delay from the arrival of the first bit of a 10 ms radio frame on the CPRI AUX interface (or other direct interface) to the start of transmission of this data on the CPRI link.

Figure 3-44: Tx Path Delay from AUX Interface to CPRI Link in CPRI v6.0 IP Core



The Tx path delay from the AUX interface to the CPRI link is the sum of the following delays:

1. Fixed delay from the AUX interface though the CPRI low-level transmitter to the Tx elastic buffer. This delay depends on the device family and the current CPRI line bit rate.
2. Variable delay through the Tx elastic buffer, as well as the phase difference between the core clock `cpri_clkout` and the transceiver `tx_clkout` clock. The "Extended Delay Measurement" section shows how to calculate the delay in the CPRI v6.0 IP core Tx elastic buffer, which includes the phase difference delay.
3. Variable Tx bitslip delay in CPRI RE slaves. Refer to "Tx Bitslip Delay."
4. Fixed delay from the Tx elastic buffer to the transceiver. This delay depends on the device family and the CPRI line bit rate.

   Note: In IP core variations that target an Intel Stratix 10 device, you must also add the delay through the Stratix 10 hard FIFOs in the Tx path. Refer to "Extended Delay Measurement for Intel Stratix 10 Hard FIFOs" section.
5. Link delay through the transceiver.

Related Information

TX_DELAY Register on page 5-16

Round-Trip Delay

You can read the `ROUND_TRIP_DELAY` register or calculate the round-trip delay from the delay components (Rx path delay, Tx path delay,cable delay, and Rx-to-Tx switching latency). The `round_trip_delay` field of the `ROUND_TRIP_DELAY` register in an REC master records the total round-trip delay from the start of the internal transmit radio frame in the REC to the start of the internal receive radio frame in the REC, that is, from SAP to SAP, in CPRI REC and RE masters.

CPRI V6.0 Specification requirements R-20 and R-21 address the round-trip delay. Requirement R-20 addresses the measurement without including the cable delay, and requirement R-21 is the requirement for the cable delay. Both requirements state that the variation must be no more than ±16.276 ns.

To monitor the round-trip delay, you must check periodically to confirm that the variation in measurements over time is small enough that the requirements are met.

Note: The round trip delay is the sum of the Tx path delays through the REC master and the RE slave, the Rx path delays through the REC master and the RE slave, the cable delay, and the Tx-to-Rx switching delay, sometimes referred to as the loopback delay. The Tx-to-Rx switching delay is labeled T_S_Rx_Tx in the figure. The Tx-to-Rx switching delay depends on the loopback path and the device.

Related Information

ROUND_TRIP_DELAY Register on page 5-19

Multi-Hop Delay Measurement

In a multi-hop system, you must combine the delays between and through the different CPRI masters and CPRI RE slaves to determine the round-trip delay.

To determine the round-trip delay of a full multi-hop system, you must add together the values in the `ROUND_TRIP_DELAY` registers of the REC and RE masters in the system, plus the delays through the external routers, and subtract the loopback delay from all the hops except the final hop.

Round-trip delay = Σ_i round_trip_delay (hop i) + Σ_j (TBdelayUL + TBdelayDL)(j) – Σ_j T_S_Rx_Tx(j)

where the REC and RE masters in the configuration are labeled i = 0, 1, ...,n and the routing layers in the configuration, and their uplink and downlink delays, are labeled j = 0, 1, ...(n-1).

As the equation shows, you must omit the loopback delay from the single-hop calculation for all but the final pair of CPRI link partners. The loopback delay is only relevant at the turnaround point of the full multi-hop path.

Delay Calibration Features

The CPRI v6.0 IP core provides multiple calibration features to support compliance with the CPRI v6.0 Specification.

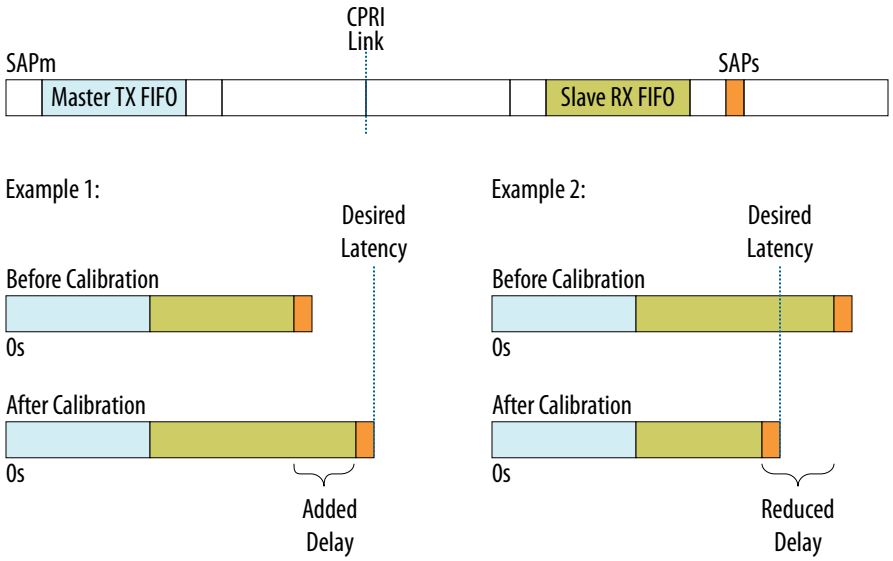
- If you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor, supports single-trip delay calibration using the `DELAY_CAL_STD_CTRL1`, `DELAY_CAL_STD_CTRL2`, `DELAY_CAL_STD_CTRL3`, `DELAY_CAL_STD_CTRL3`, `DELAY_CAL_STD_CTRL4`, `DELAY_CAL_STD_CTRL5`, and `DELAY_CAL_STD_STATUS` registers. You can connect the IOPLL and DPCU blocks that Intel provides with the CPRI v6.0 IP core to ensure correct calibration results.
- If you turn on **Enable round-trip delay calibration** in the CPRI v6.0 parameter editor, supports round-trip delay calibration using the `DELAY_CAL_RTD` register.
- In all supported device families, increases the consistency of the round-trip delay using the `XCVR_BITSLIP` register.

Single-Trip Delay Calibration

The CPRI v6.0 IP core provides an optional mechanism to support calibrating the total delay through a CPRI master and slave on the downlink in a single hop.

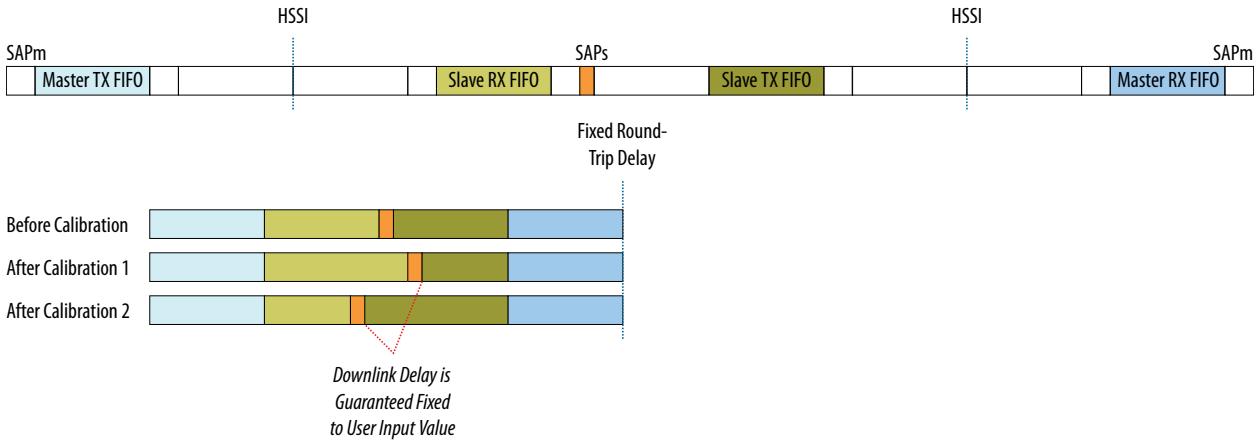
If you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor, the CPRI v6.0 IP core and IOPLL and dynamic phase control unit (DPCU) blocks work together to adjust the delay through the RE slave Rx path in response to information about the delay through the REC or RE master Tx path.

Figure 3-45: Downlink Slave Delay Adjustment for Single-Trip Delay Calibration



The feature introduces the new delay to maintain a single-trip delay measurement as close as possible to the desired single-trip delay you provide to the CPRI v6.0 IP core. The application can also provide Tx path delay information that is appropriate for other use scenarios. For example, you might want to adjust the latency to the synchronization SAP to compensate for the IQ mapper requirements in your system. Refer to the Altera wiki Latency Formula and Calculation Example page.

Figure 3-46: Single-Trip Delay Does Not Affect Link Delay



The `cal_cycle_delay` and `cal_step_delay` fields of the `DELAY_CAL_STD_CTRL2` register in the CPRI link slave hold the anticipated delay that you program. The `cal_current_delay` field of the `DELAY_CAL_STD_STATUS` register in the CPRI link slave holds the total actual variable delay measurement in the single trip from the synchronization SAP in the CPRI link master to the synchronization SAP in the current CPRI link slave. You can manually specify a consistency check whenever you want, or you can specify that the IP core should run a consistency check once every hyperframe.

If you turn on single-trip delay calibration by setting the `cal_en` bit in the `DELAY_CAL_STD_CTRL1` register, the single-trip calibration feature is active. The user programs the `cal_cycle_delay` and `cal_step_delay` fields with the number of whole and fractional `cpri_clkout` cycles of single-trip variable delay that the system requires. After each consistency check, the CPRI v6.0 slave IP core adjusts the Rx delay to compensate for mismatches between the programmed, required single-trip delay and the actual single-trip variable delay recorded in the `cal_current_delay` register field. The delay adjustment mechanism is dynamic phase shifting of the `cpri_coreclk`.

The slave IP core requires the master IP core measured Tx delay information to calculate the `cal_current_delay` value. The master IP core sends this information to its downlink slave by one of two possible mechanisms. You program the master IP core `DELAY_CAL_STD4` register to specify whether the master sends this information in the incoming hyperframe (and at which location in the hyperframe) or the system writes it in the dedicated slave IP core `DELAY_CAL_STD_CTRL5` register. You program the slave IP core `DELAY_CAL_STD3` register to specify whether the slave receives this information in the incoming hyperframe (and at which location in the hyperframe) or the system writes it in the dedicated slave `DELAY_CAL_STD_CTRL5` register. If both CPRI master and slave are CPRI v6.0 IP cores, and the register values in the CPRI link master and slave do not match, the single-trip delay calibration does not function correctly. If the CPRI master is not a CPRI v6.0 IP core, the CPRI slave must receive the correct information in the programmed register or hyperframe location.

Related Information

Altera wiki CPRI IP v6 Latency Formula and Calculation Example
Provides information about how to use the single-trip delay calibration feature.

Single-Trip Latency Measurement and Calibration Interface Signals

Table 3-20: Single-Trip Latency Measurement and Calibration Interface Signals

If you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor, the single-trip latency measurement and calibration interface is available. This interface is designed to connect to the DPCU block that helps implement single-trip delay calibration.

All interface signals are clocked by the `reconfig_clk` clock.

Signal Name	Direction	Description
`cal_status[1:0]`	Input	Status information from DPCU to CPRI v6.0 IP core.
`cal_ctrl[15:0]`	Output	Control information from CPRI v6.0 IP core to DPCU.

Round-Trip Delay Calibration

If you turn on **Enable round-trip delay calibration** in the CPRI v6.0 parameter editor, the CPRI v6.0 IP core provides an additional, optional mechanism to help minimize the variation in the round-trip delay through a CPRI REC or RE master.

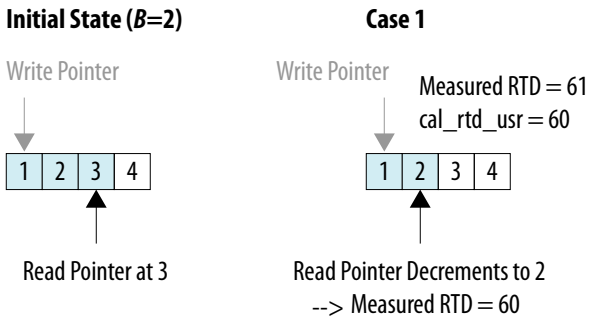
If you turn on **Enable round-trip delay calibration** in the CPRI v6.0 parameter editor, the dynamic pipelining feature for round-trip delay calibration introduces a delay in the Rx path in an REC or RE master. This delay is introduced to the Rx path immediately following the Rx elastic buffer. The feature introduces the new delay to maintain a round-trip delay measurement as close as possible to the anticipated round-trip delay you provide to the CPRI v6.0 IP core. The `DELAY_CAL_RTD` register holds the anticipated delay that you program, an enable bit you turn on to activate the feature, and a status field in which the CPRI v6.0 IP core reports its relative success in maintaining the round-trip delay you requested.

The CPRI v6.0 IP core is configured with a set of $N=2^B$ pipelined registers in the Rx path, where B is the value you specified for **Round-trip delay calibration FIFO depth** in the parameter editor. If the two lowest order bits of the `cal_rtd_ctrl` field (bits [25:24] of the `DELAY_CAL_RTD` register) have the value of 2'b01, the auto-calibration feature is active. The user programs the `cal_rtd_usr` field with the expected number of `cpri_clkout` cycles of round-trip delay. The CPRI v6.0 IP core adjusts the number of pipeline registers the data passes through (in contrast to the number of registers it bypasses) to compensate for mismatches between the desired round-trip delay programmed in the `cal_rtd_usr` field and the actual round-trip delay recorded in the `ROUND_TRIP_DELAY` register.

The `cal_rtd_status` field reports whether the CPRI IP core is successful in keeping the round-trip delay at the value you prescribed in the `cal_rtd_usr` field. If the two lowest order bits of the `cal_rtd_ctrl` field (bits [25:24] of the `DELAY_CAL_RTD` register) have the value of 2'b01, the value of the `cal_rtd_status` field should remain at 2'b10. If the value reaches 2'b11, indicating that the IP core is unable to meet the calibration requirement, you should reset the link or restart the calibration. You might also need to re-instantiate the IP core with a larger number of pipeline registers to support additional adjustment.

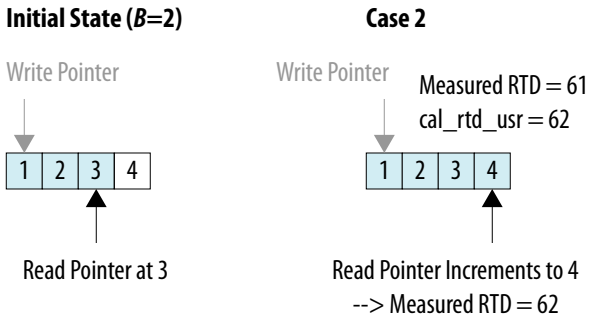
Initially, the number of pipeline registers the CPRI v6.0 IP core uses is one half the total number N of available register stages, plus one. This initial setting allows the IP core to adjust the number up or down as required, and adds $(N/2)+1$ latency cycles to the RX path delay and the round-trip delay. For buffer depth N, the pipeline read pointer can move $(N/2)-1$ entries in either direction from its initial state.

Figure 3-47: Pipeline Read Pointer Decrements to Decrease Measured Round-Trip Delay



In Case 1, the application writes the value of 60 in the `cal_rtd_usr` field of the `DELAY_CAL_RTD` register. When the CPRI v6.0 IP core measures the actual round-trip delay and sets the `round_trip_delay` field of the `ROUND_TRIP_DELAY` register to the value of 61, the CPRI v6.0 IP core responds by moving the read pointer to decrease the pipeline length, and therefore the measured round-trip delay value, by one `cpri_clkout` cycle. The adjustment achieves the desired effect: the measured round-trip delay value changes to 60.

Figure 3-48: Pipeline Read Pointer Increments to Increase Measured Round-Trip Delay



In Case 2, the application writes the value of 62 in the `cal_rtd_usr` field of the `DELAY_CAL_RTD` register. When the CPRI v6.0 IP core measures the actual round-trip delay and sets the `round_trip_delay` field of the `ROUND_TRIP_DELAY` register to the value of 61, the CPRI v6.0 IP core responds by moving the read pointer to increase the pipeline length, and therefore the measured round-trip delay value, by one `cpri_clkout` cycle. The adjustment achieves the desired effect: the measured round-trip delay value changes to 60.

Tx Bitslip Delay

To increase the consistency of the round-trip delay, the CPRI v6.0 RE slave introduces a variable bitslip on the Tx path to complement the variability in the word aligner in the transceiver on the Rx path.

The CPRI v6.0 IP core reports the Rx bitslip through the word aligner in the `rx_bitslip_out` field of the `XCVR_BITSLIP` register, and compensates for the variable delay by adding a bitslip in the Tx path. The current size of this bitslip in bits is available in the `tx_bitslip_in` field of the `XCVR_BITSLIP` register. When you leave the `tx_bitslip_en` field at its default value of 0, this feature is active.

The Tx bitslip feature ensures stability in the round-trip delay through a CPRI v6.0 RE core, but introduces a variable component in each of the Tx and Rx paths when considered independently. In CPRI v6.0 IP cores in master clocking mode, the `rx_bitslip_out` field has the constant value of 0.

If you set the value of the `tx_bitslip_en` field to 1, you can override the current `rx_bitslip_out` value to control the Tx bitslip delay manually. Intel does not recommend implementing the manual override.

In CPRI v6.0 IP core variations that target an Arria V, Cyclone V, or Stratix V device, the Tx bitslip functionality is included in the Transceiver PHY IP core that is generated with the CPRI v6.0 IP core. These variations include the `XCVR_BITSLIP` register to support manual override of the Tx bitslip delay.

Note: Intel does not recommend implementing the manual override for the Tx bitslip.

The total of the Tx bitslip delay and the word aligner bitslip delay in the transceiver receiver is added to the detailed round-trip delay calculation as part of the Tx and Rx transceiver delays. However, the total of these two bit values does not reach the duration of a single `cpri_clkout` cycle, nor does it reach the threshold of the CPRI V6.0 specification R-20 and R-21 requirements. The bitslip delay is noticeable only with an oscilloscope.

Related Information

XCVR_BITSLIP Register on page 5-19

CPRI v6.0 IP Core Transceiver and Transceiver Management Interfaces

The CPRI v6.0 IP core configures the interface to the CPRI serial link in an Intel FPGA device transceiver channel. The IP core provides multiple interfaces for managing the transceiver. The transceiver is configured with a Native PHY IP core and exposes many of its optional interfaces for ease of IP core integration in your design.

CPRI Link

The CPRI v6.0 IP core configures the interface to the CPRI serial link in an Intel FPGA device transceiver channel.

Table 3-21: CPRI Link Interface Signals

Signal Name	Direction	Description
xcvr_rxdatain	Input	High-speed serial data receiver port.
xcvr_txdataout	Output	High-speed serial data transmitter port.
xcvr_los	Input	Asynchronous signal that forces link to LOS state for quick resynchronization. If you implement the CPRI link with a fiber optic channel, you could connect this input signal to the SFP module LOS signal so that it is asserted when the SFP module loses signal. Otherwise, you should tie this signal to the value of 0.

Main Transceiver Clock and Reset Signals

Table 3-22: Main Transceiver Clock and Reset-Done Signals

The clocks for individual interfaces are listed with the relevant interface signals.

Signal Name	Direction	Description
xcvr_cdr_refclk	Input	Receiver CDR reference clock. You must drive this clock at the frequency you specified for the **Receiver CDR reference clock frequency (MHz)** parameter in the CPRI v6.0 parameter editor.
xcvr_recovered_clk	Output	Direct recovered clock from the receiver CDR. Use this output clock to drive the external clean-up PLL when your IP core is in slave mode. This clock is present only in CPRI v6.0 IP cores in slave clocking mode with **Operation mode** set to the value of **RX/TX Duplex** or **RX Simplex**.
xcvr_reset_tx_done	Output	Indicates the transmitter and IP core Tx path have completed the internal reset sequence. This signal is clocked by the cpri_clkout clock.
xcvr_reset_rx_done	Output	Indicates the receiver and IP core Rx path have completed the internal reset sequence. This signal is clocked by the cpri_clkout clock.

Signal Name	Direction	Description
tx_analogreset_ack	Output	This signal rises after the TX analog reset process completes. This signal falls after you deassert the tx_analogreset signal.This signal is asynchronous. Refer to Resetting Transceiver Channels in the *Intel Arria 10 Transceiver PHY User Guide*or the appropriate Intel Stratix 10 Transceiver PHY user guide. This signal is available in CPRI v6.0 IP cores that target anIntel Arria 10 device or an Intel Stratix 10 device. Your custom auto-rate negotiation logic can monitor this signal to determine when it can safely begin reconfiguring the device transceiver to a new CPRI line bit rate.
rx_analogreset_ack	Output	This signal rises after the RX analog reset process completes. This signal falls after you deassert the rx_analogreset signal. This signal is asynchronous. Refer to Resetting Transceiver Channels in the *Intel Arria 10 Transceiver PHY User Guide* or the appropriate Intel Stratix 10 Transceiver PHY user guide. This signal is available in CPRI v6.0 IP cores that target an Intel Arria 10 device or an Intel Stratix 10 device. Your custom auto-rate negotiation logic can monitor this signal to determine when it can safely begin reconfiguring the device transceiver to a new CPRI line bit rate.

Related Information

- **Resetting Transceiver Channels section of the Intel Arria 10 Transceiver PHY User Guide**
  Information about the tx_analogreset_ack and rx_analogreset_ack signals on Intel Arria 10 devices.
- **Resetting Transceiver Channels section of the Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide**
  Information about the tx_analogreset_ack and rx_analogreset_ack signals on Intel Stratix 10 ES1 devices.
- **Resetting Transceiver Channels section of the Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide**
  Information about the tx_analogreset_ack and rx_analogreset_ack signals on Intel Stratix 10 production devices.

Arria V, Arria V GZ, Cyclone V, and Stratix V Transceiver Reconfiguration Interface

This interface is available only if you turn on at least one of these parameters in the CPRI v6.0 parameter editor:

- **Enable line bit rate auto-negotiation**
- **Enable start-up sequence state machine**

Table 3-23: Arria V, Arria V GZ, Cyclone V, and Stratix V Transceiver Reconfiguration Interface Signals

All interface signals are clocked by the reconfig_clk clock.

Signal Name	Direction	Description
reconfig_clk	Input	Clock for CPRI v6.0 IP core transceiver start-up and reconfiguration. The frequency range for this clock is 100–150 MHz.
reconfig_reset	Input	Asynchronous active-high reset signal for transceiver start-up and reconfiguration. Used for rate switching and auto-rate negotiation.
reconfig_to_xcvr[69:0]	Input	Parallel transceiver reconfiguration bus from the Altera Transceiver Reconfiguration Controller to the transceiver in the CPRI v6.0 IP core.
reconfig_from_xcvr[45:0]	Output	Parallel transceiver reconfiguration bus to the Altera Transceiver Reconfiguration Controller from the transceiver in the CPRI v6.0 IP core.

Related Information

- **Adding the Transceiver Reconfiguration Controller** on page 2-22
- **Altera Transceiver PHY IP Core User Guide**
  Information about the Transceiver Reconfiguration Controller IP core.

Intel Arria 10 and Intel Stratix 10 Transceiver Reconfiguration Interface

If you turn on certain parameters in the CPRI v6.0 parameter editor, Intel provides a dedicated Avalon-MM interface, called the transceiver reconfiguration interface, to access the Intel Arria 10 or Intel Stratix 10 transceiver registers. You access the Native PHY IP core registers through this dedicated interface and not through the IP core general purpose control and status interface. This interface provides access to the hard PCS registers on the device.

The Avalon-MM interface implements a standard memory-mapped protocol. You can connect an Avalon master to this bus to access the registers of the embedded Native PHY IP core.

This interface is available only in variations that target an Intel Arria 10 or Intel Stratix 10 device, and only if you turn on at least one of these parameters in the CPRI v6.0 parameter editor:

- **Enable line bit rate auto-negotiation**
- **Enable start-up sequence state machine**
- **Enable ADME, transceiver capability, control and status registers access**
- **Enable single-trip delay calibration**

Table 3-24: CPRI v6.0 IP Core Transceiver Reconfiguration Interface Signals

The reconfig_clk clocks the signals on the CPRI v6.0 IP core transceiver reconfiguration interface.

Signal Name	Direction	Description
reconfig_clk	Input	Clocks the signals on the CPRI v6.0 transceiver reconfiguration interface. Supports frequency range 100–150 MHz.
reconfig_reset	Input	Asynchronous active-high reset signal. Resets the CPRI v6.0 transceiver reconfiguration interface and all of the registers to which it provides access.

Signal Name	Direction	Description
reconfig_write	Input	You must assert this signal to request a write transfer.
reconfig_read	Input	You must assert this signal to request a read transfer.
reconfig_address[9:0]	Input	Address for reads and writes.
reconfig_writedata[31:0]	Input	Write data.
reconfig_readdata[31:0]	Output	Read data.
reconfig_waitrequest	Output	The interface is busy. Do not issue Avalon-MM commands to this interface while this signal is high.

Figure 3-49: Read Transaction on the Transceiver Reconfiguration Interface

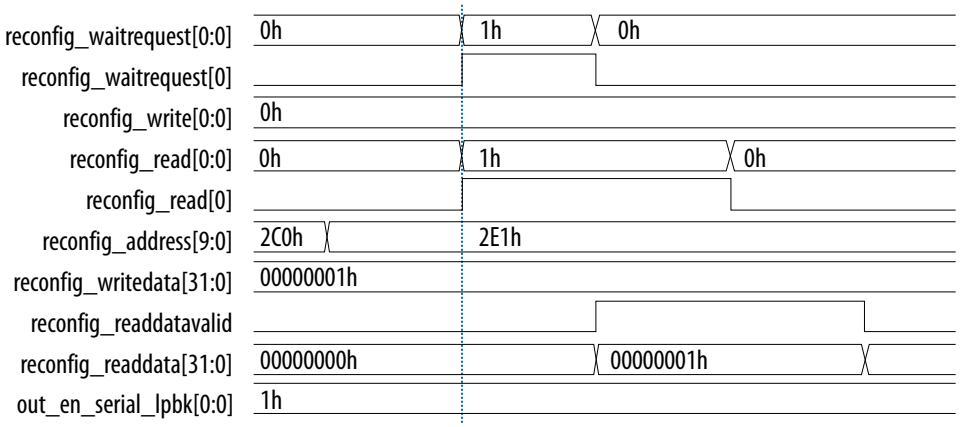
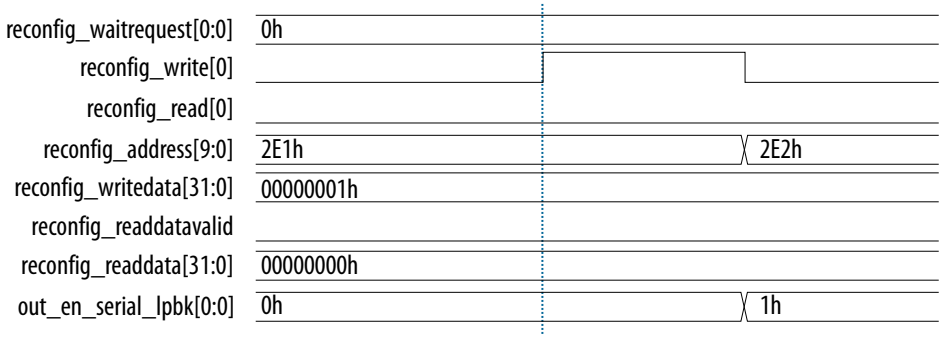


Figure 3-50: Write Transaction on the Transceiver Reconfiguration Interface



Related Information

- **Avalon Interface Specifications**
  For more information about the Avalon-MM protocol, refer to the *Avalon Memory-Mapped Interfaces* chapter.
- **Intel Arria 10 Transceiver PHY User Guide**
  Information about the Intel Arria 10 Native PHY IP core hard PCS registers that you can program through the transceiver reconfiguration interface.
- **Arria 10 Transceiver Registers**
  Detailed information about the Intel Arria 10 transceiver registers.
- **Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide**
  Information about the Intel Stratix 10 Native PHY IP core hard PCS registers that you can program through the transceiver reconfiguration interface in your Intel Stratix 10 design that targets an Intel Stratix 10 ES1 device.
- **Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide**
  Information about the Intel Stratix 10 Native PHY IP core hard PCS registers that you can program through the transceiver reconfiguration interface in your Intel Stratix 10 design that targets an Intel Stratix 10 production device.

Interface to the External Reset Controller

Table 3-25: CPRI v6.0 IP Core External Reset Controller Interface Signals

The CPRI v6.0 IP core requires that you generate and connect at least one external transceiver reset controller.

Signal Name	Direction	Description
`xcvr_tx_analogreset`	Input	Analog reset to transmitter from external reset controller.
`xcvr_tx_digitalreset`	Input	Digital reset to transmitter from external reset controller.
`xcvr_tx_cal_busy`	Output	Indicates to external reset controller that the transmitter is still busy with the calibration process.
`xcvr_rx_analogreset`	Input	Analog reset to receiver from external reset controller.
`xcvr_rx_digitalreset`	Input	Digital reset to receiver from external reset controller.
`xcvr_rx_cal_busy`	Output	Indicates to external reset controller that the receiver is still busy with the calibration process.
`xcvr_reset_tx_ready`	Input	Indicates the Tx reset controller reset sequence is completed. When this signal is asserted, the IP core begins a reset of the IP core Tx path.
`xcvr_reset_rx_ready`	Input	Indicates the Rx reset controller reset sequence is completed. When this signal is asserted, the IP core begins a reset of the IP core Rx path.

Related Information

- **Adding the Reset Controller** on page 2-20
- **Altera Transceiver PHY IP Core User Guide**
  Information about the Altera Transceiver Reset Controller IP core for 28-nm devices.
- **Intel Arria 10 Transceiver PHY User Guide**
  Information about the Transceiver Reset Controller IP core for Intel Arria 10 devices.
- **Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide**
  Information about the Intel Stratix 10 Transceiver Reset Controller IP core for Intel Stratix 10 ES1 devices.
- **Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide**
  Information about the Intel Stratix 10 Transceiver Reset Controller IP core for Intel Stratix 10 production devices.

Interface to the External PLL

Table 3-26: CPRI v6.0 IP Core External PLL Interface Signals

The CPRI v6.0 IP core requires that you generate and connect an external transceiver PLL IP core.

Signal Name	Direction	Description
`xcvr_ext_pll_clk`	Input	Clocks the transmitter PMA. You should drive this input clock with the output of the external transceiver TX PLL. In Arria 10 devices, you have a choice of different TX PLL IP cores to configure. You must ensure that you configure a PLL IP core that is capable of driving the frequency that the CPRI v6.0 IP core requires to run at the specified CPRI line bit rate.

Related Information

- **Adding the Transceiver TX PLL IP Core** on page 2-18
- **Altera Transceiver PHY IP Core User Guide**
  Information about the 28-nm device transceiver PLL IP cores.
- **Arria 10 Transceiver PHY User Guide**
  Information about the Arria 10 transceiver PLL IP cores.

Transceiver Debug Interface

Table 3-27: Transceiver Debug Interface Signals

The CPRI v6.0 IP core provides a `xcvr_rx_is_lockedtodata` status signal. If you turn on **Enable L1 debug interfaces** in the CPRI v6.0 parameter editor, the IP core provides some additional status signals from the transceiver.

All of the transceiver debug signals are asynchronous.

Signal Name	Direction	Description
xcvr_rx_is_lockedtodata	Output	Indicates that the receiver CDR is locked to the incoming serial data. This signal is available whether or not you turn on **Enable L1 debug interfaces** in the parameter editor.
xcvr_rx_is_lockedtoref	Output	Indicates that the receiver CDR is locked to the xcvr_cdr_refclk reference clock.
xcvr_rx_errdetect[3:0]	Output	Each bit [n] indicates the receiver has detected an 8B/10B code group violation in byte [n] of the 32-bit data word.
xcvr_rx_disperr[3:0]	Output	Each bit [n] indicates that the receiver has detected an 8B/10B parity error in byte [n] of the 32-bit data word.
xcvr_rx_blk_sh_err	Output	Indicates that the receiver has detected a 64B/66B SYNC_HEADER violation.

Testing Features

The CPRI v6.0 IP core supports multiple testing features.

CPRI v6.0 IP Core Loopback Modes

Figure 3-51: CPRI v6.0 IP Core Loopback Modes

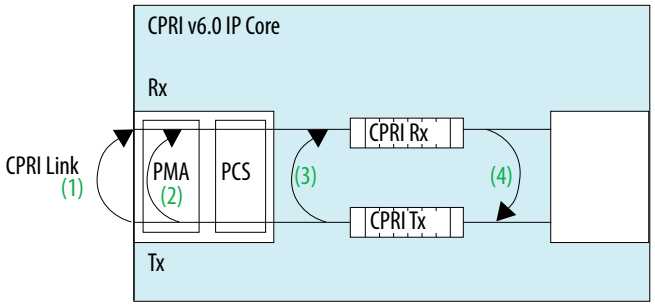


Table 3-28: Loopback Modes

Tag in Figure	Description	How to Configure
1	External loopback: Use this configuration to test the full Tx and Rx paths from an application through the CPRI link and back to the application.	Connect a CPRI REC master's CPRI Tx interface to its CPRI Rx interface by physically connecting the CPRI v6.0 IP core's high-speed transceiver output pins to its high-speed transceiver input pins. The connection medium must support the data rate requirements of the CPRI v6.0 IP core.

Tag in Figure	Description	How to Configure
2	Transceiver PMA serial forward loopback path is active.	Turn on **Enable transceiver PMA serial forward loopback path** in the parameter editor and set the `loop_forward` field of the `LOOPBACK` register to the value 2'b01.
3	Active parallel loopback path does not exercise the transceiver.	Turn on **Enable parallel forward loopback paths** in the parameter editor and set the `loop_forward` field of the `LOOPBACK` register to the value 2'b10 (to include the stitching logic to the transceiver) or 2'b11 (to exclude the stitching logic to the transceiver).
4	Reverse loopback path is active.	Turn on **Enable parallel reversed loopback paths** in the parameter editor and set the `loop_reversed` field of the `LOOPBACK` register to a non-zero value. The register value specifies the parts of the CPRI frame that participate in the loopback path. Other parts of the CPRI frame are filled in from the local IP core.

Related Information

LOOPBACK Register on page 5-15

CPRI v6.0 IP Core Self-Synchronization Feature

Intel provides a self-synchronization testing feature that supports an RE slave in a CPRI link external loopback configuration. This feature is intended to work correctly only for Layer 1 testing.

By default, only an REC master can function correctly in a CPRI link external loopback configuration. An RE slave in external loopback configuration cannot achieve frame synchronization, because the CPRI RX interface must lock on to the K28.5 character before the CPRI TX interface can begin sending K28.5 characters. Therefore, no K28.5 character is ever transmitted on the RE slave loopback CPRI link.

However, in an RE slave CPRI v6.0 IP core you can specify that the CPRI TX interface begin sending K28.5 characters before the CPRI Rx interface locks on to the K28.5 character from the CPRI link. This feature supports a CPRI RE slave in achieving frame synchronization without being connected to a CPRI master, and allows you to test your CPRI RE slave without the need for an additional CPRI v6.0 IP core.

To turn on this feature, connect your CPRI RE slave in a CPRI link external loopback configuration, and set the `tx_enable_force` field of the `L1_CONFIG` register to the value of 1.

Related Information

L1_CONFIG Register on page 5-5

CPRI v6.0 IP Core Signals

4

2019.01.02

UG-20008

Subscribe Send Feedback

The CPRI v6.0 IP core communicates with the surrounding design through multiple external signals. Many of the signal interfaces are optional; their presence or absence depends on whether or not you enable the corresponding interface in the CPRI v6.0 parameter editor.

In the case of the interfaces that provide direct access to all or part of the CPRI frame, write transmit delay relative to the AUX TX interface synchronization signals depends on the **Auxiliary and direct interfaces write latency cycle(s)** parameter setting.

CPRI v6.0 IP Core L2 Interface

The CPRI v6.0 IP core optionally communicates with an user-provided Ethernet MAC through the following signals.

Table 4-1: MII and GMII Signals

Each IP core has either MII signals, GMII signals, or neither.

Signal Name	Direction	Interface
mii_rxclk	Input	RX MII signals These signals are available only if you set the value of **Ethernet PCS interface** to **MII** in the CPRI v6.0 parameter editor.
mii_rxreset_n	Input	
mii_rxdv	Output	
mii_rxer	Output	
mii_rxd[3:0]	Output	
mii_txclk	Input	TX MII signals These signals are available only if you set the value of **Ethernet PCS interface** to **MII** in the CPRI v6.0 parameter editor.
mii_txreset_n	Input	
mii_txen	Input	
mii_txer	Input	
mii_txd[3:0]	Input	

Signal Name	Direction	Interface
mii_tx_fifo_status[3:0]	Output	MII status signals These signals are available only if you set the value of **Ethernet PCS interface** to **MII** in the CPRI v6.0 parameter editor.
mii_rx_fifo_status[3:0]	Output	
gmii_rxclk	Input	RX GMII signals These signals are available only if you set the value of **Ethernet PCS interface** to **GMII** in the CPRI v6.0 parameter editor.
gmii_rxreset_n	Input	
gmii_rxdv	Output	
gmii_rxer	Output	
gmii_rxd[7:0]	Output	
gmii_txclk	Input	TX GMII signals These signals are available only if you set the value of **Ethernet PCS interface** to **GMII** in the CPRI v6.0 parameter editor.
gmii_txreset_n	Input	
gmii_txen	Input	
gmii_txer	Input	
gmii_txd[7:0]	Input	
gmii_txfifo_status[3:0]	Output	GMII status signals These signals are available only if you set the value of **Ethernet PCS interface** to **GMII** in the CPRI v6.0 parameter editor.
gmii_rxfifo_status[3:0]	Output	

Related Information

- **Media Independent Interface (MII) to External Ethernet Block** on page 3-44
  Describes the MII signals and provides MII timing diagrams.
- **Gigabit Media Independent Interface (GMII) to External Ethernet Block** on page 3-47

CPRI v6.0 IP Core L1 Direct Access Interfaces

The CPRI v6.0 IP core can communicate with the surrounding design through multiple optional interfaces that provide direct access to all or part of the CPRI frame.

Table 4-2: L1 Direct Access Interface Signals

Signal Name	Direction	Description
aux_rx_rfp	Output	AUX RX interface status signals These signals are available only if you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor.
aux_rx_hfp	Output	
aux_rx_bfn[11:0]	Output	
aux_rx_z[7:0]	Output	
aux_rx_x[7:0]	Output	
aux_rx_seq[6:0]	Output	
aux_rx_data[31:0]	Output	AUX RX interface data signals These signals are available only if you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor.
aux_rx_ctrl[3:0]	Output	
aux_tx_sync_rfp	Input	AUX TX interface control and status signals These signals are available only if you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor.
aux_tx_err[3:0]	Output	
aux_tx_rfp	Output	
aux_tx_hfp	Output	
aux_tx_bfn[11:0]	Output	
aux_tx_z[7:0]	Output	
aux_tx_x[7:0]	Output	
aux_tx_seq[6:0]	Output	
aux_tx_data[31:0]	Input	AUX TX interface data signals These signals are available only if you turn on **Enable auxiliary interface** in the CPRI v6.0 parameter editor.
aux_tx_mask[31:0]	Input	
aux_tx_ctrl[3:0]	Output	
iq_rx_valid[3:0]	Output	Direct IQ RX interface These signals are available only if you turn on **Enable direct IQ mapping interface** in the CPRI v6.0 parameter editor.
iq_rx_data[31:0]	Output	
iq_tx_ready[3:0]	Output	Direct IQ TX interface These signals are available only if you turn on **Enable direct IQ mapping interface** in the CPRI v6.0 parameter editor.
iq_tx_valid[3:0]	Input	
iq_tx_data[31:0]	Input	
ctrl_axc_rx_valid[3:0]	Output	Direct Ctrl_AxC RX interface These signals are available only if you turn on **Enable direct ctrl_axc access interface** in the CPRI v6.0 parameter editor.
ctrl_axc_rx_data[31:0]	Output	

Signal Name	Direction	Description
ctrl_axc_tx_ready[3:0]	Output	Direct Ctrl_AxC TX interface These signals are available only if you turn on **Enable direct ctrl_axc access interface** in the CPRI v6.0 parameter editor.
ctrl_axc_tx_valid[3:0]	Input	
ctrl_axc_tx_data[31:0]	Input	
vs_rx_valid[3:0]	Output	Direct VS RX interface These signals are available only if you turn on **Enable direct vendor specific access interface** in the CPRI v6.0 parameter editor.
vs_rx_data[31:0]	Output	
vs_tx_ready[3:0]	Output	Direct VS TX interface These signals are available only if you turn on **Enable direct vendor specific access interface** in the CPRI v6.0 parameter editor.
vs_tx_valid[3:0]	Input	
vs_tx_data[31:0]	Input	
rtvs_rx_valid	Output	Direct RTVS RX interface These signals are available only if you turn on **Enable direct real-time vendor specific interface** in the CPRI v6.0 parameter editor.
rtvs_rx_data[31:0]	Output	
rtvs_tx_ready	Output	Direct RTVS TX interface These signals are available only if you turn on **Enable direct real-time vendor specific interface** in the CPRI v6.0 parameter editor.
rtvs_tx_valid	Input	
rtvs_tx_data[31:0]	Input	
hdlc_rx_valid	Output	Direct HDLC serial RX interface These signals are available only if you turn on **Enable HDLC serial interface** in the CPRI v6.0 parameter editor.
hdlc_rx_data	Output	
hdlc_tx_ready	Output	Direct HDLC serial TX interface These signals are available only if you turn on **Enable HDLC serial interface** in the CPRI v6.0 parameter editor.
hdlc_tx_valid	Input	
hdlc_tx_data	Input	

Signal Name	Direction	Description
z130_local_lof	Output	Direct L1 control and status interface These signals are available only if you turn on **Enable direct Z.130.0 alarm bits access interface** in the CPRI v6.0 parameter editor.
z130_local_los	Output	
z130_sdi_assert	Input	
z130_local_rai	Output	
z130_reset_assert	Input	
z130_remote_lof	Output	
z130_remote_los	Output	
z130_sdi_req	Output	
z130_remote_rai	Output	
z130_reset_req	Output	

Related Information

- **AUX Interface** on page 3-18
- **AUX Interface Signals** on page 3-18
- **Direct IQ Interface** on page 3-30
- **Ctrl_AxC Interface** on page 3-32
- **Direct Vendor Specific Access Interface** on page 3-34
- **Real-Time Vendor Specific Interface** on page 3-36
- **Direct HDLC Serial Interface** on page 3-38
- **Direct L1 Control and Status Interface** on page 3-40

CPRI v6.0 IP Core Management Interfaces

The CPRI v6.0 IP core provides multiple interfaces for managing the IP core and the properties of the CPRI link.

Table 4-3: CPRI v6.0 IP Core Management Signals

Signal Name	Direction	Description
cpri_clkout	Output	Main clock signals
cpri_coreclk	Input	
tx_clkout	Output	
reset_n	Input	Main reset signals
reset_rx_n	Input	
reset_tx_n	Input	

Signal Name	Direction	Description
cpu_clk	Input	CPU interface
cpu_reset_n	Input	
cpu_address[15:0]	Input	
cpu_byteenable[3:0]	Input	
cpu_read	Input	
cpu_write	Input	
cpu_writedata[31:0]	Input	
cpu_readdata[31:0]	Output	
cpu_waitrequest	Output	
cpu_irq	Output	
state_startup_seq[2:0]	Output	Start-up sequence interface With the exception of the state_l1_synch signal, these signals are available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor.
state_l1_synch[2:0]	Output	
nego_bitrate_complete	Input	
nego_protocol_complete	Input	
nego_cm_complete	Input	
nego_vss_complete	Input	
nego_l1_timer_expired	Input	
nego_bitrate_in[4:0]	Input	Auto-rate negotiation control and status interface These signals are available only if you turn on **Enable line bit rate auto-negotiation** in the CPRI v6.0 parameter editor.
nego_bitrate_out[4:0]	Output	
ex_delay_clk	Input	Extended delay measurement interface
ex_delay_reset_n	Input	
latency_sclk	Input	Extended delay measurement interface signals that are available only in IP core variations that target an Intel Stratix 10 device.
latency_sreset_n	Input	

Signal Name	Direction	Description
cal_status[1:0]	Input	Single-trip delay calibration interface This signals are available only if you turn on **Enable single-trip delay calibration** in the CPRI v6.0 parameter editor.
cal_ctrl[15:0]	Output	
rx_lcv	Output	L1 debug interface These signals are available only if you turn on **Enable L1 debug interfaces** in the CPRI v6.0 parameter editor.
rx_freq_alarm	Output	

Related Information

- **CPRI v6.0 IP Core Clocking Structure** on page 3-3
- **CPRI v6.0 IP Core Reset Requirements** on page 3-10
- **CPU Interface to CPRI v6.0 IP Core Registers** on page 3-49
- **CPU Interface Signals** on page 3-50
- **Start-Up Sequence Interface Signals** on page 3-15
- **Auto-Rate Negotiation** on page 3-55
- **Extended Delay Measurement** on page 3-56
- **Extended Delay Measurement Interface** on page 3-59
- **L1 Debug Interface** on page 3-44

CPRI v6.0 IP Core Transceiver and Transceiver Management Signals

The CPRI v6.0 IP core configures the interface to the CPRI serial link in an Intel FPGA device transceiver channel. The IP core provides multiple interfaces for managing the transceiver. The transceiver is configured with a Native PHY IP core and exposes many of its optional interfaces for ease of IP core integration in your design.

Table 4-4: Transceiver and Transceiver Management Signals

Signal Name	Direction	Description
xcvr_cdr_refclk	Input	Main transceiver clock and reset-done signals The tx_analogreset_ack and rx_analogreset_ack signals are available only in Intel Arria 10 and Intel Stratix 10 variations.
xcvr_recovered_clk	Output	
xcvr_reset_tx_done	Output	
xcvr_reset_rx_done	Output	
tx_analogreset_ack	Output	
rx_analogreset_ack	Output	

Signal Name	Direction	Description
xcvr_rxdatain	Input	CPRI link interface
xcvr_txdataout	Output	
xcvr_los	Input	
reconfig_clk	Input	28-nm device transceiver reconfiguration interface These signals are present only in IP core variations that target an Arria V, Arria V GZ, Cyclone V, or Stratix V device.
reconfig_to_xcvr[69:0]	Input	
reconfig_from_xcvr[45:0]	Output	
reconfig_clk	Input	Intel Arria 10 or Intel Stratix 10 transceiver reconfiguration interface These signals are present only in IP core variations that target an Intel Arria 10 or Intel Stratix 10 device.
reconfig_reset	Input	
reconfig_write	Input	
reconfig_read	Input	
reconfig_address[9:0]	Input	
reconfig_writedata[31:0]	Input	
reconfig_readdata[31:0]	Output	
reconfig_waitrequest	Output	

Signal Name	Direction	Description
xcvr_tx_analogreset	Input	Interface to external reset controller
xcvr_tx_digitalreset	Input	
xcvr_tx_cal_busy	Output	
xcvr_rx_analogreset	Input	
xcvr_rx_digitalreset	Input	
xcvr_rx_cal_busy	Output	
xcvr_reset_tx_ready	Input	
xcvr_reset_rx_ready	Input	
xcvr_rx_analogreset_stat	Input	
xcvr_rx_digitalreset_stat(2)	Input	
xcvr_rx_analogreset_stat(2)	Input	
xcvr_tx_digitalreset_stat(2)	Input	
xcvr_ext_pll_clk	Input	Interface to external TX PLL
xcvr_rx_is_lockedtodata	Output	Transceiver debug interface
xcvr_rx_is_lockedtoref	Output	Transceiver debug interface These signals are present only if you turn on **Enable L1 debug interfaces** in the CPRI v6.0 parameter editor.
xcvr_rx_errdetect[3:0]	Output	
xcvr_rx_disperr[3:0]	Output	
xcvr_rx_blk_sh_err	Output	

Related Information

- **Main Transceiver Clock and Reset Signals** on page 3-69
- **CPRI Link** on page 3-68
- **Arria V, Arria V GZ, Cyclone V, and Stratix V Transceiver Reconfiguration Interface** on page 3-70
- **Intel Arria 10 and Intel Stratix 10 Transceiver Reconfiguration Interface** on page 3-71
- **Interface to the External Reset Controller** on page 3-73
- **Interface to the External PLL** on page 3-74
- **Transceiver Debug Interface** on page 3-74

(2) These signals are present only in IP core variation that target an Intel Stratix 10 device.

CPRI v6.0 IP Core Registers 5

2019.01.02

UG-20008

The CPRI v6.0 IP core internal registers are accessible using the CPU interface, an Avalon-MM interface which conforms to the *Avalon Interface Specifications*.

All of these registers are 32 bits wide and the addresses are shown as hexadecimal byte address values. The registers can be accessed on a 32-bit (4-byte) basis. The addressing for the registers therefore increments by units of 4.

Write access to a Reserved or undefined location has no effect. Read accesses to a Reserved or undefined location return an undefined result.

Table 5-1: Register Access Codes

Lists the access codes used to describe the type of register bits.

Code	Description
RW	Read / write
RO	Read only
RC	Read to clear
UR0	Reserved —undefined result on read, no effect on write

Table 5-2: Control and Status Register Map

Offset	Register Name	Function	Location of Additional Information
0x00	`INTR`	Interrupt Control and Status	**INTR Register** on page 5-3
0x04	`L1_STATUS`	Layer 1 Status	**L1_STATUS Register** on page 5-4
0x08	`L1_CONFIG`	Layer 1 Configuration	**L1_CONFIG Register** on page 5-5
0x0C	`BIT_RATE_CONFIG`	Bit Rate Configuration	**BIT_RATE_CONFIG Register** on page 5-5
0x10	`PROT_VER`	Protocol Version Control and Status	**PROT_VER Register** on page 5-6

Offset	Register Name	Function	Location of Additional Information
0x14	TX_SCR	Transmitter Scrambler Control	**TX_SCR Register** on page 5-7
0x18	RX_SCR	Receiver Scrambler Status	**RX_SCR Register** on page 5-7
0x1C	CM_CONFIG	Layer 2 Control and Management Configuration	**CM_CONFIG Register** on page 5-8
0x20	CM_STATUS	Layer 2 Control and Management Status	**CM_STATUS Register** on page 5-9
0x24	START_UP_SEQ	Start-Up Sequence Control and Status	**START_UP_SEQ Register** on page 5-9
0x28	START_UP_TIMER	Start-Up Sequence Timer Control	**START_UP_TIMER Register** on page 5-11
0x2C	FLSAR	L1 Inband Z.130.0 Control and Status	**FLSAR Register** on page 5-11
0x30	CTRL_INDEX	Control Word Index	**CTRL_INDEX Register** on page 5-12
0x34	TX_CTRL	Transmit Control Word	**TX_CTRL Register** on page 5-13
0x38	RX_CTRL	Receive Control Word	**RX_CTRL Register** on page 5-14
0x3C	RX_ERR	Receiver Error Status	**RX_ERR Register** on page 5-14
0x40	RX_BFN	Recovered Radio Frame Counter	**RX_BFN Register** on page 5-14
0x44	LOOPBACK	Loopback Control	**LOOPBACK Register** on page 5-15
0x48	TX_DELAY	Transmit Buffer Delay Control and Status	**TX_DELAY Register** on page 5-16
0x4C	RX_DELAY	Receiver Buffer Delay Control and Status	**RX_DELAY Register** on page 5-17
0x50	TX_EX_DELAY	Transmit Buffer Extended Delay Measurement Control and Status	**TX_EX_DELAY Register** on page 5-18
0x54	RX_EX_DELAY	Receiver Buffer Extended Delay Measurement Status	**RX_EX_DELAY Register** on page 5-18
0x58	ROUND_TRIP_DELAY	Round Trip Delay	**ROUND_TRIP_DELAY Register** on page 5-19
0x5C	XCVR_BITSLIP	Transceiver Bit Slip Control and Status	**XCVR_BITSLIP Register** on page 5-19
0x60	DELAY_CAL_STD_CTRL1	Single-Trip Delay Calibration Control 1	**DELAY_CAL_STD_CTRL1 Register** on page 5-20

Offset	Register Name	Function	Location of Additional Information
0x64	DELAY_CAL_STD_CTRL2	Single-Trip Delay Calibration Control 2	**DELAY_CAL_STD_CTRL2 Register** on page 5-21
0x68	DELAY_CAL_STD_CTRL3	Single-Trip Delay Calibration Control 3	**DELAY_CAL_STD_CTRL3 Register** on page 5-22
0x6C	DELAY_CAL_STD_CTRL4	Single-Trip Delay Calibration Control 4	**DELAY_CAL_STD_CTRL4 Register** on page 5-22
0x70	DELAY_CAL_STD_CTRL5	Single-Trip Delay Calibration Control 5	**DELAY_CAL_STD_CTRL5 Register** on page 5-23
0x74	DELAY_CAL_STD_STATUS	Single-Trip Delay Calibration Status	**DELAY_CAL_STD_STATUS Register** on page 5-24
0x80	DELAY_CAL_RTD	Round Trip Delay Calibration Control and Status	**DELAY_CAL_RTD Register** on page 5-24
0x84	XCVR_TX_FIFO_DELAY	Intel Stratix 10Transmitter FIFO Delay	**XCVR_TX_FIFO_DELAY Register** on page 5-25
0x88	XCVR_RX_FIFO_DELAY	Intel Stratix 10Receiver FIFO Delay	**XCVR_RX_FIFO_DELAY Register** on page 5-26

INTR Register

Table 5-3: CPRI v6.0 IP Core INTR Register at Offset 0x00

Bits	Field Name	Type	Value on Reset	Description
31:19	Reserved	UR0	13'b0	
18	intr_sdi_pending	RW	1'b0	Indicates a remote SDI detected interrupt is not yet serviced
17	intr_rai_pending	RW	1'b0	Indicates a remote RAI detected interrupt is not yet serviced
16	intr_reset_pending	RW	1'b0	Indicates a remote reset request or acknowledge interrupt is not yet serviced
15:3	Reserved	UR0	13'b0	
2	intr_sdi_en	RW	1'b0	Z.130.0 remote SDI detected interrupt enable
1	intr_rai_en	RW	1'b0	Z.130.0 remote RAI detected interrupt enable

Bits	Field Name	Type	Value on Reset	Description
0	intr_reset_en	RW	1'b0	Z.130.0 remote reset request or acknowledge received interrupt enable

L1_STATUS Register

Table 5-4: CPRI v6.0 IP Core L1_STATUS Register at Offset 0x04

Bits	Field Name	Type	Value on Reset	Description
31:13	Reserved	UR0	19'b0	
12	rx_rfp_hold	RC	1'b0	Radio frame pulse received. This bit is asserted every 10 ms and remains asserted until cleared by user logic.(3)
11	rx_freq_alarm_hold	RC	1'b0	CPRI receive clock is not synchronous with main IP core clock (cpri_clkout). This alarm is asserted each time mismatches are found between the recovered CPRI receive clock and cpri_clkout, and remains asserted until cleared by user logic. (3) If you turn on **Enable L1 debug interfaces** in the CPRI v6.0 parameter editor, the original asynchronous pulse that sets this register field is visible on the rx_freq_alarm output signal. However, that signal is not available if you turn off **Enable L1 debug interfaces**.
10	rx_los_hold	RC	1'b0	Hold rx_los. (3)
9	rx_err_hold	RC	1'b0	Hold rx_err. (3)
8	rx_hfnsync_hold	RC	1'b0	Hold rx_hfnsync. (3)
7:3	Reserved	UR0	5'b0	
2	rx_los	RC	1'b0	Indicates receiver is in LOS state.
1	rx_err	RC	1'b0	Indicates 8B10B LCV or 64B/66aB sync header violations detected.
0	rx_hfnsync	RC	1'b0	Indicates receiver has achieved hyperframe synchronization state (HFNSYNC).

(3) This register field is a read-to-clear field. You must read the register twice to read the true value of the field after frame synchronization is achieved. If you observe this bit asserted during link initialization, read the register again after link initialization to confirm any errors.

L1_CONFIG Register

Table 5-5: CPRI v6.0 IP Core L1_CONFIG Register at Offset 0x08

Bits	Field Name	Type	Value on Reset	Description
31:4	Reserved	UR0	28'b0	
3	tx_ctrl_insert_en	RW	1'b0	Master enable for insertion of control transmit table entries in CPRI hyperframe. This signal enables control bytes for which the CTRL_INDEX register tx_control_insert bit is high to be written to the CPRI frame.
2	tx_enable_force	RW	1'b0	Specifies whether the RE slave self-synchronization testing feature is activated. If the feature is activated, the CPRI RE slave attempts to achieve link synchronization without a CPRI link connection to a CPRI master. Set this field to the value of 1 to enable the feature. The value of 1 is only valid if the CPRI IP core is configured as a CPRI slave.
1	synchronization_mode	RW	(4)	Specifies whether the CPRI v6.0 IP core is configured as a CPRI slave or a CPRI master, according to the following values: • 1'b0: The IP core is configured as a CPRI master. • 1'b1: The IP core is configured as a CPRI slave.
0	tx_enable	RW	1'b0	Enable transmission on CPRI link.

Related Information

- **Writing the Hyperframe Control Words** on page 3-54
- **CPRI v6.0 IP Core Self-Synchronization Feature** on page 3-76

BIT_RATE_CONFIG Register

Table 5-6: CPRI v6.0 IP Core BIT_RATE_CONFIG Register at Offset 0x0C

Bits	Field Name	Type	Value on Reset	Description
31:5	Reserved	UR0	27'b0	

(4) Reset value is the value you specify for **Synchronization mode** in the CPRI v6.0 parameter editor.

Bits	Field Name	Type	Value on Reset	Description
4:0	bit_rate	(5)	(6)	CPRI line bit rate to be used in next attempt to achieve frame synchronization, encoded according to the following valid values: • 5'b00001: 0.6144 Gbps • 5'b00010: 1.2288 Gbps • 5'b00100: 2.4576 Gbps • 5'b00101: 3.0720 Gbps • 5'b01000: 4.9150 Gbps • 5'b01010: 6.1440 Gbps • 5'b01100: 8.11008 Gbps • 5'b10000: 9.8304 Gbps • 5'b10100: 10.1376 Gbps If the input signal nego_bitrate_in has a non-zero value, the CPRI v6.0 IP core uses the encoded value driven on nego_bitrate_in in the next attempt to achieve frame synchronization, and ignores the value in the bit_rate register field. The value driven on the nego_bitrate_in signal, if it is non-zero, always overrides the value in this register field.

Related Information

Auto-Rate Negotiation on page 3-55

PROT_VER Register

Table 5-7: CPRI v6.0 IP Core PROT_VER Register at Offset 0x10

Bits	Field Name	Type	Value on Reset	Description
31:25	Reserved	UR0	7'b0	
24	rx_prot_ver_valid	RO	1'b0	Value received in incoming Z.2.0 control byte is a valid CPRI v6.0 protocol version encoding.
23:16	rx_prot_ver	RO	8'b0	Encoded protocol version received in incoming Z.2.0 control byte.
15:10	Reserved	UR0	6'b0	

(5) If you turn on **Enable line bit rate auto-negotiation**, this register field is a RW register field. If you turn off **Enable line bit rate auto-negotiation**, this register field is a RO register field.

(6) Reset value is the value you specify for **Line bit rate** in the CPRI v6.0 parameter editor.

Bits	Field Name	Type	Value on Reset	Description
9	prot_ver_auto	RW	1'b1	Enables auto negotiation of protocol version. If you turn on **Enable protocol version and C&M channel setting auto-negotiation**, this field is a RW register field with the default value of 1. Otherwise, this field is a RO register field with the default value of 0.
		RO	1'b0	
8	rx_prot_ver_filter	RW	1'b1	Enable filtering or protection of the Z.2.0 value across five consecutive hyperframes. If you turn on **Enable protocol version and C&M channel setting auto-negotiation**, this field is a RW register field with the default value of 1. Otherwise, this field is a RO register field with the default value of 0.
		RO	1'b0	
7:0	tx_prot_ver	RW	8'b01	Transmit protocol version to be mapped to Z.2.0 to indicate whether or not the current hyperframe transmission is scrambled. The value 1 indicates it is not scrambled and the value 2 indicates it is scrambled. If the prot_ver_auto field has the value of 1, the IP core automatically updates the tx_prot_ver field.

TX_SCR Register

Table 5-8: CPRI v6.0 IP Core TX_SCR Register at Offset 0x14

Bits	Field Name	Type	Value on Reset	Description
31	Reserved	UR0	1'b0	
30:0	tx_scr_seed	RW	31'b0	Transmitter scrambler seed. If the seed has value 0, the transmission is not scrambled.

RX_SCR Register

Table 5-9: CPRI v6.0 IP Core RX_SCR Register at Offset 0x18

Bits	Field Name	Type	Value on Reset	Description
31	rx_scr_active	RO	1'b0	Indicates that the incoming hyperframe is scrambled. The value 1 indicates that the incoming communication is scrambled, and the value 0 indicates that it is not scrambled. The IP core determines whether or not the incoming communication is scrambled based on the protocol version.
30:0	rx_scr_seed	RO	31'b0	Received scrambler seed. The receiver descrambles the incoming CPRI communication based on this seed.

CM_CONFIG Register

Table 5-10: CPRI v6.0 IP Core CM_CONFIG Register at Offset 0x1C

Bits	Field Name	Type	Value on Reset	Description
31:13	Reserved	UR0	19'b0	
12	slow_cm_rate_auto	RW	1'b1	Enable auto-negotiation of HDLC rate. If you turn on **Enable protocol version and C&M channel setting auto-negotiation**, this field is a RW register field with the default value of 1. Otherwise, this field is a RO register field with the default value of 0.
		RO	1'b0	
11	slow_cm_rate_filter	RW	1'b1	Enable filtering of HDLC rate.
10:8	tx_slow_cm_rate	RW	3'b110	Rate configuration for slow Control and Management (HDLC). To be inserted in CPRI control byte Z.66.0.
7	fast_cm_ptr_auto	RW	1'b1	Enable auto-negotiation of Ethernet rate. If you turn on **Enable protocol version and C&M channel setting auto-negotiation**, this field is a RW register field with the default value of 1. Otherwise, this field is a RO register field with the default value of 0.
		RO	1'b0	
6	fast_cm_ptr_filter	RW	1'b1	Enable filtering of Ethernet rate.
5:0	tx_fast_cm_ptr	RW	6'd20	Pointer to first CPRI control word used for fast Control and Management (Ethernet). To be inserted in CPRI control byte Z.194.0.

CM_STATUS Register

Table 5-11: CPRI v6.0 IP Core CM_STATUS Register at Offset 0x20

Bits	Field Name	Type	Value on Reset	Description
31:12	Reserved	UR0	20'b0	
11	rx_slow_cm_rate_valid	RO	1'b0	Indicates that a valid HDLC rate has been accepted.
10:8	rx_slow_cm_rate	RO	3'b0	Accepted received HDLC rate. The IP core receives this rate in the incoming Z.66.0 control byte.
7	Reserved	UR0	1'b0	
6	rx_fast_cm_ptr_valid	RO	1'b0	Indicates that a valid Ethernet rate has been accepted. The IP core receives this rate in the incoming Z.194.0 control byte.
5:0	rx_fast_cm_ptr	RO	6'b0	Accepted received Ethernet rate. Valid values are between 0x14 (decimal 20) and 0x3F (decimal 63), inclusive.

START_UP_SEQ Register

Table 5-12: CPRI v6.0 IP Core START_UP_SEQ Register at Offset 0x24

This register is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor.

Bits	Field Name	Type	Value on Reset	Description
31:17	Reserved	UR0	15'b0	
16	startup_timer_expired	RO	1'b0	Indicates that the internal L1 start-up timer is expired, based on the value of the startup_timer_period field of the START_UP_TIMER register.
15:11	Reserved	UR0	5'b0	

Bits	Field Name	Type	Value on Reset	Description
10:8	state_startup_seq	RO	3'b0	Indicates the current state of the start-up sequence. This field has the following valid values: • 3'b000: State A: Standby • 3'b001: State B: L1 Synchronization • 3'b011: State C: Protocol Setup • 3'b010: State D: Control and Management Setup • 3'b110: State E: Interface and VSS Negotiation • 3'b111: State F: Operation • 3'b101: State G: Passive Link
7:4	Reserved	UR0	4'b0	
3	nego_vss_complete	RW	1'b0	Indicates the Vendor Specific negotiation is complete. You must set this bit to move the start-up sequence state machine from state E to state F. If you turn on **Enable start-up sequence state machine**, the nego_vss_complete input signal writes directly to this register bit.
2	nego_cm_complete	RW	1'b0	Indicates the Control and Management negotiation is complete and the start-up sequence state machine can move from state D to state E. If the slow_cm_rate_auto field or the fast_cm_ptr_auto field, or both, in the CM_CONFIG register has the value of 1, the IP core updates this bit if the user does not update it. If you turn on **Enable start-up sequence state machine**, the nego_cm_complete input signal writes directly to this register bit. **Note:** If the Control and Management negotiation cannot complete successfully, per Figure 30 in the CPRI specification V6.0 (2013-08-30), the start-up sequence state machine moves from state D to state G. In that case the IP core does not set the nego_cm_complete bit, and the state_startup_seq field reflects the state change.
1	nego_protocol_complete	RW	1'b0	Indicates the protocol version negotiation is complete and the start-up sequence state machine can move from state C to state D. If the prot_ver_auto field of the PROT_VER register has the value of 1, the IP core updates this bit if the user does not update it. If you turn on **Enable start-up sequence state machine**, the nego_protocol_complete input signal writes directly to this register bit.

Bits	Field Name	Type	Value on Reset	Description
0	nego_bitrate_complete	RW	1'b0	Indicates the CPRI line bit rate negotiation is complete. If you turn on **Enable start-up sequence state machine**, the nego_bitrate_complete input signal writes directly to this register bit.

START_UP_TIMER Register

Table 5-13: CPRI v6.0 IP Core START_UP_TIMER Register at Offset 0x28

This register is available only if you turn on **Enable start-up sequence state machine** in the CPRI v6.0 parameter editor.

Bits	Field Name	Type	Value on Reset	Description
31:20	Reserved	UR0	12'b0	
19:0	startup_timer_period	RW	20'b0	Threshold value for L1 start-up timer to expire. The unit is cpri_coreclk clock cycles.

FLSAR Register

Table 5-14: CPRI v6.0 IP Core FLSAR Register at Offset 0x2C

The FLSAR register is the L1 inband control word Z.130.0 control and status register.

Bits	Field Name	Type	Value on Reset	Description
31	Reserved	UR0	1'b0	
30	local_lof	RO	1'b0	Local loss of frame (LOF) detected
29	local_los	RO	1'b0	Local loss of signal (LOS) detected
28	lof_detected	RO	1'b0	Remote LOF detected
27	los_detected	RO	1'b0	Remote LOS detected
26	sdi_detected	RO	1'b0	Remote service access point (SAP) defect indication (SDI) detected.

Bits	Field Name	Type	Value on Reset	Description
25	rai_detected	RO	1'b0	Remote alarm indication (RAI) detected.
24	reset_detected	RO	1'b0	Reset request or acknowledgement detected.
23:17	Reserved	UR0	7'b0	
16	sdi_gen	RW	1'b0	Enable Z.130.0 SDI generation.
15:9	Reserved	UR0	7'b0	
8	rai_gen	RW	1'b1	Enable Z.130.0 RAI generation as a result of local loss of signal (LOS) or loss of frame (LOF).
7:1	Reserved	UR0	7'b0	
0	reset_gen	RW	1'b0	Enable Z.130.0 reset generation.

Related Information

Direct L1 Control and Status Interface on page 3-40

CTRL_INDEX Register

Table 5-15: CPRI v6.0 IP Core CTRL_INDEX Register at Offset 0x30

Frequency differences between the control and status interface clock cpu_clk and the main CPRI v6.0 IP core clock cpri_clkout might cause non-zero read latency and more than one clock cycle of write latency when accessing this register.

Bits	Field Name	Type	Value on Reset	Description
31:27	Reserved	UR0	5'b0	
26:24	rx_ctrl_seq	RW	3'b0	Sequence number for CPRI control word 32-bit section monitoring. The value in this field determines the 32-bit section of the control receive table entry that appears in the RX_CTRL register. The value indicates whether this is the first, second, third, fourth, or fifth (for RTVS) 32-bit section in the control word.
23:16	rx_ctrl_x	RW	8'b0	Index for CPRI control word monitoring (X value in frame location #Z.X.Y). The value in this field determines the control receive table entry of which a 32-bit section appears in the RX_CTRL register.
15:12	Reserved	UR0	4'b0	

Bits	Field Name	Type	Value on Reset	Description
11	tx_ctrl_insert	RW	1'b0	Control word 32-bit section transmit enable. This value is stored in the control transmit table with its associated entry. When you change the value of the tx_ctrl_seq field or the tx_ctrl_x field, the stored tx_ctrl_insert bit associated with the indexed entry appears in the tx_ctrl_insert field. At the time the CPRI v6.0 IP core can insert a control transmit table entry in the associated position in the outgoing hyperframe on the CPRI link, if the tx_ctrl_insert bit associated with that entry has the value of 1, and the tx_ctrl_insert_en bit of the L1_CONFIG register is asserted, the IP core inserts the table entry in the hyperframe.
10:8	tx_ctrl_seq	RW	3'b0	Sequence number for CPRI control word 32-bit section insertion. The value in this field determines the 32-bit section of the control transmit table entry that appears in the TX_CTRL register. The value indicates whether this is the first, second, third, fourth, or fifth (for RTVS) 32-bit section in the control word.
7:0	tx_ctrl_x	RW	8'b0	Index for CPRI control word insertion (X value in frame location #Z.X.Y). The value in this field determines the control transmit table entry of which a 32-bit section appears in the TX_CTRL register.

TX_CTRL Register

Table 5-16: CPRI v6.0 IP Core TX_CTRL Register at Offset 0x34

Frequency differences between the control and status interface clock cpu_clk and the main CPRI v6.0 IP core clock cpri_clkout might cause non-zero read latency and more than one clock cycle of write latency when accessing this register.

Bits	Field Name	Type	Value on Reset	Description
31:0	tx_ctrl_data	RW	32'b0	CPRI control word 32-bit section to be transmitted in CPRI hyperframe position Z.x, where x is the index in the tx_ctrl_x field of the CTRL_INDEX register. The tx_ctrl_seq field of the CTRL_INDEX register indicates whether this is the first, second, third, fourth, or fifth such 32-bit section.

RX_CTRL Register

Table 5-17: CPRI v6.0 IP Core RX_CTRL Register at Offset 0x38

Frequency differences between the control and status interface clock `cpu_clk` and the main CPRI v6.0 IP core clock `cpri_clkout` might cause non-zero read latency and more than one clock cycle of write latency when accessing this register.

Bits	Field Name	Type	Value on Reset	Description
31:0	`rx_ctrl_data`	RO	32'b0	Most recent received CPRI control word 32-bit section from CPRI hyperframe position Z.x, where x is the index in the `rx_ctrl_x` field of the `CTRL_INDEX` register. The `rx_ctrl_seq` field of the `CTRL_INDEX` register indicates whether this is the first, second, third, fourth, or fifth such 32-bit section.

RX_ERR Register

Table 5-18: CPRI v6.0 IP Core RX_ERR Register at Offset 0x3C

Bits	Field Name	Type	Value on Reset	Description
31:16	Reserved	UR0	16'b0	
15:8	`sh_err`	RC	8'b0	Number of 64B/66B sync header violations detected in the transceiver. Enables CPRI link debugging. This register turns over to the value of 0 when it increments from the value of 255.
7:0	`lcv`	RC	8'b0	Number of line code violations (LCVs) detected in the 8B/10B decoding block in the transceiver. Enables CPRI link debugging. This register turns over to the value of 0 when it increments from the value of 255. This counter includes LCVs that occur during initialization.

RX_BFN Register

Table 5-19: CPRI v6.0 IP Core RX_BFN Register at Offset 0x40

Bits	Field Name	Type	Value on Reset	Description
31:12	Reserved	UR0	20'b0	
11:0	`bfn`	RO	12'b0	Current BFN number.

LOOPBACK Register

Table 5-20: CPRI v6.0 IP Core LOOPBACK Register at Offset 0x44

Bits	Field Name	Type	Value on Reset	Description
31:11	Reserved	UR0	21'b0	
10:8	`loop_reversed`	RW	3'b0	Testing reverse loopback mode. If you turn on **Enable parallel reversed loopback paths** in the CPRI v6.0 parameter editor, this register field specifies the parts of the CPRI frame that are sent on the reverse loopback path. If you do not turn on this parameter, you should leave this register field at its default value of 3'b0. For standard testing, you would turn on these loopback modes in a CPRI RE slave only. This field has the following valid values: • 3'b000: No loopback. • 3'b001: Full CPRI frame loopback. Incoming CPR data and control words are sent back as-is in outgoing CPR communication. • 3'b010: I/Q data loopback. Incoming CPRI data are sent back in outgoing CPRI communication; control words are generated locally. • 3'b011: Fast control and management loopback. Incoming CPRI fast control and management (Ethernet) control and data words are sent back in outgoing CPRI communication; remaining data and control words are generated locally. • 3'b100: Fast control and management and VS loopback. Incoming CPRI fast control and management (Ethernet) and vendor-specific control words are sent back in outgoing CPRI communication; data and remaining control words are generated locally.
7:2	Reserved	UR0	6'b0	

Bits	Field Name	Type	Value on Reset	Description
1:0	loop_forward	RW	2'b0	Testing forward loopback mode. If you turn on **Enable transceiver PMA serial forward loopback path** or **Enable parallel forward loopback paths** in the CPRI v6.0 parameter editor, this register field specifies the loopback path that is currently active. If you do not turn on either of these parameters, you should leave this register field at its default value of 2'b0. This field has the following valid values: • 2'b00: No loopback. • 2'b01: Transceiver PMA loopback path is active. This path does not exercise the transceiver PCS. This option is available only if you turn on **Enable transceiver PMA serial forward loopback path** . • 2'b10: Active loopback path includes extended delay measurement logic but does not exercise the transceiver. This option is available only if you turn on **Enable parallel forward loopback paths**. • 2'b11: Active loopback path includes framing and deframing logic, but does not exercise the extended delay measurement logic and does not exercise the transceiver. This option is available only if you turn on **Enable parallel forward loopback paths**.

Related Information

CPRI v6.0 IP Core Loopback Modes on page 3-75

TX_DELAY Register

Table 5-21: CPRI v6.0 IP Core TX_DELAY Register at Offset 0x48

Bits	Field Name	Type	Value on Reset	Description
31:9	Reserved	UR0	23'b0	

Bits	Field Name	Type	Value on Reset	Description
8	tx_buf_resync	RW	1'b0	Force transmit buffer pointer resynchronization. You can use this register field to resynchronize if, for example, the buffer fill level becomes too high due to due to environmental impacts on the device, such as temperature. Resynchronizing might lead to data loss or corruption. Do not use this register field to resynchronize after a dynamic CPRI line bit rate change. After a dynamic CPRI line bit rate change the IP core forces resynchronization internally without referring to this register.
7:4	Reserved	UR0	4'b0	
3:0	tx_buf_delay	RO	4'b0	Current transmit buffer fill level.

Related Information

Tx Path Delay on page 3-62

RX_DELAY Register

Table 5-22: CPRI v6.0 IP Core RX_DELAY Register at Offset 0x4C

Bits	Field Name	Type	Value on Reset	Description
31:25	Reserved	UR0	7'b0	
24	rx_buf_resync	RW	1'b0	Force receive buffer pointer resynchronization. You can use this register field to resynchronize if, for example, the buffer fill level becomes too high due to due to environmental impacts on the device, such as temperature. Resynchronizing might lead to data loss or corruption. Do not use this register field to resynchronize after a dynamic CPRI line bit rate change. After a dynamic CPRI line bit rate change the IP core forces resynchronization internally without referring to this register.
23:17	Reserved	UR0	7'b0	
16	rx_byte_delay	RO	1'b0	Current byte-alignment delay. This field is relevant for the Rx path delay calculation.
15:RX_BUF_DEPTH	Reserved	UR0	0	

Bits	Field Name	Type	Value on Reset	Description
(RX_BUF_DEPTH - 1):0	rx_buf_delay	RO	0	Current receive buffer fill level. Unit is 32-bit words. Maximum value is $2^{RX\_BUF\_DEPTH}$-1.

Related Information

Rx Path Delay on page 3-61

TX_EX_DELAY Register

Table 5-23: CPRI v6.0 IP Core TX_EX_DELAY Register at Offset 0x50

Bits	Field Name	Type	Value on Reset	Description
31:24	tx_msrm_period	RW	8'b0	Integration period for Tx buffer extended delay measurement. Program this field with the user-defined value N, where M/N = ex_delay_clk period / cpri_clkout period.
23	tx_ex_delay_valid	RC	1'b0	Indicates that the tx_ex_delay field has been updated.
22:16	Reserved	UR0	7'b0	
15:0	tx_ex_delay	RO	16'b0	Tx buffer extended delay measurement result. Unit is cpri_clkout clock periods.

Related Information

- Extended Delay Measurement on page 3-56
- Extended Delay Measurement Interface on page 3-59

RX_EX_DELAY Register

Table 5-24: CPRI v6.0 IP Core IRX_EX_DELAY Register at Offset 0x54

Bits	Field Name	Type	Value on Reset	Description
31:24	rx_msrm_period	RW	8'b0	Integration period for Rx buffer extended delay measurement. Program this field with the user-defined value N_R, where M/N_R = ex_delay_clk period / cpri_clkout period.

Bits	Field Name	Type	Value on Reset	Description
23	rx_ex_delay_valid	RC	1'b0	Indicates that the rx_ex_delay field has been updated.
22:RX_BUF_DEPTH	Reserved	UR0	0	
(RX_BUF_DEPTH - 1):0	rx_ex_delay	RO	0	Rx buffer extended delay measurement result. Unit is cpri_clkout clock periods.

Related Information

- **Extended Delay Measurement** on page 3-56
- **Extended Delay Measurement Interface** on page 3-59

ROUND_TRIP_DELAY Register

Table 5-25: CPRI v6.0 IP Core ROUND_TRIP_DELAY Register at Offset 0x58

Bits	Field Name	Type	Value on Reset	Description
31:20	Reserved	UR0	12'b0	
19:0	round_trip_delay	RO	20'b0	Measured round-trip delay from aux_tx_rfp to aux_rx_rfp. Unit is cpri_clkout clock periods. The IP core updates this field with every radio frame.

Related Information

Round-Trip Delay on page 3-63

XCVR_BITSLIP Register

Table 5-26: CPRI v6.0 IP Core XCVR_BITSLIP Register at Offset 0x5C

Bits	Field Name	Type	Value on Reset	Description
31:21	Reserved	UR0	11'b0	

Bits	Field Name	Type	Value on Reset	Description
20:16	rx_bitslip_out	RO	5'b0	Number of bits of delay (bitslip) detected at the receiver word-aligner. Value can change at frame synchronization, when the transceiver is resetting. Any K28.5 symbol position change that occurs when word alignment is activated changes the bitslip value.
15:6	Reserved	UR0	10'b0	
5	tx_bitslip_en	RW	1'b0	Enable manual tx_bitslip_in updates.
4:0	tx_bitslip_in	RW	5'b0	Number of bits of delay (bitslip) the CPRI v6.0 IP core adds at the CPRI Tx link. The CPRI line bit rate determines the following maximum values for this field: • Maximum value for IP core variations with CPRI line bit rate 0.6144 Gbps: 9 bits. • Maximum value for IP core variations with CPRI line bit rate greater than 0.6144 Gbps: 19 bits.

Related Information

Tx Bitslip Delay on page 3-68

DELAY_CAL_STD_CTRL1 Register

Table 5-27: CPRI v6.0 IP Core DELAY_CAL_STD_CTRL1 Register at Offset 0x60

This register is available only in CPRI IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:17	Reserved	UR0	15'b0	
16	cal_reset	RW	1'b0	Reset single-trip delay calibration. Set this bit to the value of 1 to reset the IP core single-trip delay calibration module and the Intel-provided DPCU module that you have connected to the IP core. Reset this bit to the value of 0 to resume the normal functionality of these two modules. Setting this bit to the value of 1 resets all of the single-trip delay calibration logic. This reset does not affect the values in the DELAY_CAL_STD_CTRL*n* registers. However, the reset does override the value in the cal_int_check field.

Bits	Field Name	Type	Value on Reset	Description
15:9	Reserved	UR0	7'b0	
8	cal_int_check	RW	1'b0	Enable single-trip delay consistency checking. If you set this bit to the value of 1, the IP core checks once per hyperframe that the sum of the cal_step_delay and cal_cycle_delay field values in the DELAY_CAL_STD_CTRL2 register match the cal_current_delay field value in the DELAY_CAL_STD_STATUS register. If they do not match, the IP core triggers recalibration. If you set this bit to the value of 0, the IP core performs a consistency check only when you set the cal_en bit. If cal_int_check has the value of 0, the user can schedule consistency checks by resetting and setting the cal_en bit.
7:1	Reserved	UR0	7'b0	
0	cal_en	RW	1'b0	Enable single-trip delay calibration. Set this bit to the value of 1 to activate single-trip delay calibration. Reset this bit to the value of 0 to turn off single-trip delay calibration.

DELAY_CAL_STD_CTRL2 Register

Table 5-28: CPRI v6.0 IP Core DELAY_CAL_STD_CTRL2 Register at Offset 0x64

This register is available only in CPRI slave IP cores with the single-trip delay calibration feature.

The user provides this information to specify the anticipated or desired duration of the total variable component of the single-trip delay. This value reflects the system requirements.

Bits	Field Name	Type	Value on Reset	Description
31:25	Reserved	UR0	7'b0	
24:16	cal_step_delay	RW	9'b0	Additional fractional cpri_clkout clock cycles of delay in step units.
15:8	Reserved	UR0	8'b0	
7:0	cal_cycle_delay	RW	8'b0	Delay in full cpri_clkout clock cycles.

DELAY_CAL_STD_CTRL3 Register

Table 5-29: CPRI v6.0 IP Core DELAY_CAL_STD_CTRL3 Register at Offset 0x68

This register is available only in CPRI master IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:17	Reserved	UR0	15'b0	
16	cal_send_en	RW	1'b0	Enable a CPRI master IP core to include TX delay information in outgoing CPRI communication. Software must specify the location of this information in the transmitted radio frame by writing the location information in the cal_send_seq and cal_send_x fields.
15	Reserved	UR0	1'b0	
14:8	cal_send_seq	RW	7'b0	In a CPRI master IP core, specifies the sequence number in the outging basic frame that is the location of the TX delay information the IP core provides to the receiving CPRI slave. **Note:** If the CPRI slave that receives CPRI communication from this IP core on the CPRI link is an Intel FPGA CPRI v6.0 IP core, the value in this field must be identical to the value in the cal_rcv_seq field of the DELAY_CAL_STD_CTRL4 register in that CPRI slave.
7:0	cal_send_x	RW	8'b0	In a CPRI master IP core, specifies the basic frame number in the incoming hyperframe that is the location of the TX delay information the IP core provides to the receiving CPRI slave. **Note:** If the CPRI slave that receives CPRI communication from this IP core on the CPRI link is an Intel FPGA CPRI v6.0 IP core, the value in this field must be identical to the value in the cal_rcv_x field of the DELAY_CAL_STD_CTRL4 register in that CPRI slave.

DELAY_CAL_STD_CTRL4 Register

Table 5-30: CPRI v6.0 IP Core DELAY_CAL_STD_CTRL4 Register at Offset 0x6C

This register is available only in CPRI slave IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:17	Reserved	UR0	15'b0	
16	cal_rcv_en	RW	1'b0	Enable a CPRI slave IP core to receive TX delay information in incoming CPRI communication. Software must specify the location of this information in the incoming radio frame by writing the location information in the cal_rcv_seq and cal_rcv_x fields. The cal_tx_delay_usr_en field of the DELAY_CAL_STD_CTRL5 register overrides this register field.
15	Reserved	UR0	1'b0	
14:8	cal_rcv_seq	RW	7'b0	In a CPRI slave IP core, specifies the sequence number in the incoming basic frame that is the location of the TX delay information provided by the transmitting CPRI master. **Note:** If the CPRI master that transmits to this IP core on the CPRI link is an Intel FPGA CPRI v6.0 IP core, the value in this field must be identical to the value in the cal_send_seq field of the DELAY_CAL_STD_CTRL3 register in that CPRI master.
7:0	cal_rcv_x	RW	8'b0	In a CPRI slave IP core, specifies the basic frame number in the incoming hyperframe that is the location of the TX delay information provided by the transmitting CPRI master. **Note:** If the CPRI master that transmits to this IP core on the CPRI link is an Intel FPGA CPRI v6.0 IP core, the value in this field must be identical to the value in the cal_send_x field of the DELAY_CAL_STD_CTRL3 register in that CPRI master.

DELAY_CAL_STD_CTRL5 Register

Table 5-31: CPRI v6.0 IP Core DELAY_CAL_STD_CTRL5 Register at Offset 0x70

This register is available only in CPRI slave IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:17	Reserved	UR0	15'b0	

Bits	Field Name	Type	Value on Reset	Description
16	cal_tx_delay_usr_en	RW	1'b0	Enable a CPRI slave IP core to receive TX delay information in the cal_tx_delay_usr field. When the value of this field is 1, the IP core cannot receive TX delay information in incoming CPRI communication.
15:12	Reserved	UR0	4'b0	
11:0	cal_tx_delay_usr	RW	12'b0	TX delay value provided by software. This field is valid only when the cal_tx_delay_usr_en field has the value of 1. Unit is clk_ex_delay clock cycles.

DELAY_CAL_STD_STATUS Register

Table 5-32: CPRI v6.0 IP Core DELAY_CAL_STD_STATUS Register at Offset 0x74

This register is available only in CPRI slave IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:16	Reserved	UR0	16'b0	
15:0	cal_current_delay	RO	16'b0	Variable delay from the synchronization service access point (SAP) in the CPRI link master to the synchronization SAP in this CPRI v6.0 slave IP core. Unit is clk_ex_delay clock cycles. The IP core calculates this value. The Intel-provided single-trip delay calibration modules use this value to determine the values they write to the cal_step_delay and cal_cycle_delay fields of the IP core's DELAY_CAL_STD_CTRL2 register.

DELAY_CAL_RTD Register

Table 5-33: CPRI v6.0 IP Core DELAY_CAL_RTD Register at Offset 0x80

This register is available only in CPRI master IP cores with the single-trip delay calibration feature.

Bits	Field Name	Type	Value on Reset	Description
31:30	Reserved	UR0	2'b0	

Bits	Field Name	Type	Value on Reset	Description
29:28	cal_rtd_status	RW	2'b0	Round trip delay calibration status. Valid values are: • 00: Calibration is turned off. • 01: Calibration is running. • 11: Error: IP core is unable to meet the calibration requirement. • 10: Calibration has completed or is in the monitoring stage.
27	Reserved	UR0	1'b0	
26:24	cal_rtd_ctrl	RW	3'b0	Round trip delay calibration control. Valid values are one-hot: • Bit [26]: Active high reset bit that resets the calibration block. Use this bit to reset the calibration block after an error or to re-enable the calibration block to take it out of bypass mode. • Bit [25]: Enable or disable calibration block bypass mode. When the value of this bit is 1, round trip delay calibration is in bypass mode. When the value of this bit is 0, round trip delay calibration is not in bypass mode. • Bit [24]: Enable or disable round trip delay calibration. When the value of this bit is 1, round trip delay calibration is turned on. When the value of this bit is 0, round trip delay calibration is turned off.
23:20	Reserved	UR0	4'b0	
19:0	cal_rtd_usr	RW	20'b0	Desired round-trip delay value. Unit is cpri_clkout cycles.

XCVR_TX_FIFO_DELAY Register

Table 5-34: CPRI v6.0 IP Core XCVR_TX_FIFO_DELAY Register at Offset 0x84

This register is present only in IP core variations that target an Intel Stratix 10 device.

Bits	Field Name	Type	Value on Reset	Description
31	tx_pcs_fifo_delay_valid	RO	1'b0	Indicates that the tx_pcs_fifo_delay field has been updated.
30:25	Reserved	UR0	6'b0	

Bits	Field Name	Type	Value on Reset	Description
24:16	tx_pcs_fifo_delay	RO	9'b0	Delay count value for the transmitter PCS FIFO. Unit is multiples of 128/latency_sclk clock cycles (no units). In other words, the latency through the FIFO is <*latency_sclk period*>x <*this field value: delay count*>/128.
15	tx_core_fifo_delay_valid	RO	1'b0	Indicates that the tx_core_fifo_delay field has been updated.
14:9	Reserved	UR0	6'b0	
8:0	tx_core_fifo_delay	RO	9'b0	Delay count value for the transmitter core FIFO. Unit is multiples of 128/latency_sclk clock cycles (no units). In other words, the latency through the FIFO is <*latency_sclk period*>x <*this field value: delay count*>/128.

XCVR_RX_FIFO_DELAY Register

Table 5-35: CPRI v6.0 IP Core XCVR_RX_FIFO_DELAY Register at Offset 0x88

This register is present only in IP core variations that target an Intel Stratix 10 device.

Bits	Field Name	Type	Value on Reset	Description
31	rx_pcs_fifo_delay_valid	RO	1'b0	Indicates that the rx_pcs_fifo_delay field has been updated.
30:25	Reserved	UR0	6'b0	
24:16	rx_pcs_fifo_delay	RO	9'b0	Delay count value for the receiver PCS FIFO. Unit is multiples of 128/latency_sclk clock cycles (no units). In other words, the latency through the FIFO is <*latency_sclk period*>x <*this field value: delay count*>/128.
15	rx_core_fifo_delay_valid	RO	1'b0	Indicates that the rx_core_fifo_delay field has been updated.
14:9	Reserved	UR0	6'b0	

Bits	Field Name	Type	Value on Reset	Description
8:0	rx_core_fifo_delay	RO	9'b0	Delay count value for the receiver core FIFO. Unit is multiples of 128/latency_sclk clock cycles (no units). In other words, the latency through the FIFO is <*latency_sclk period*>x <*this field value: delay count*>/128.

Additional Information A

2019.01.02

UG-20008 Subscribe Send Feedback

CPRI v6.0 IP Core User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
16.0	CPRI v6.0 MegaCore Function User Guide 16.0
15.0	CPRI v6.0 MegaCore Function User Guide 15.0
14.0 and 14.0 Arria 10 Edition	CPRI v6.0 MegaCore Function User Guide 14.0 and 14.0 Arria 10 Edition

CPRI v6.0 IP Core User Guide Revision History

Table A-1: Document Revision History

Date	Compatible ACDS Version	Changes
2019.01.02	17.0 IR3, 17.0, 17.0 Update 1, and 17.0 Update 2	Clarified the value of `xcvr_recovered_clk` for the CPRI line bit rates in *Table: CPRI v6.0 IP Core Output Clocks*.

Date	Compatible ACDS Version	Changes

2018.01.04	17.0 IR3, 17.0, 17.0 Update 1, and 17.0 Update 2	• Updated for the 17.0 software release, including Intel rebranding. Refer to **Installation and Licensing** on page 2-2, **Generating CPRI v6.0 IP Cores** on page 2-3, and **CPRI v6.0 IP Core File Structure** on page 2-3.
		• Added Stratix 10 device family support for CPRI line bit rates of 1.2288 Gbps through 10.1376 Gbps. Because Stratix 10 device support is not available in the Intel Quartus Prime Pro Edition releases v17.0, v17.0 Update 1, and v17.0 Update 2, this version of the IP core provides Stratix 10 device support only in the Intel Quartus Prime Pro Edition software v17.0 IR3. Auto-rate negotiation and simplex modes are not yet available for CPRI v6.0 IP core variations that target a Stratix 10 device.
		• Updated speed grade support table in **CPRI v6.0 IP Core Performance: Device and Transceiver Speed Grade Support** on page 1-5.
		  • Added Stratix 10 support.
		  • Added previously missing speed grade information for Arria V GZ devices at CPRI line rate 9.8304 Gbps.
		  • Removed Arria 10 support for the CPRI line rate of 0.6144 Gbps. In the 17.0.260 release, the IP core no longer supports the CPRI line rate of 0.6144 Gbps for Arria 10 devices.
		  • Corrected entries for Arria V GT and Cyclone V GT devices. The previous entries listed speed grades that do not exist for these devices.
		• Updated IP core parameters. Refer to **CPRI v6.0 IP Core Parameters** on page 2-7.
		  • Updated default value of **Line bit rate** parameter. Intel Arria 10 and Intel Stratix 10 devices do not support the CPRI line bit rate of 0.6144 Gbps. The new default value is the lowest CPRI line bit rate the target device family supports: 1228.8 Mbps for Intel Arria 10 and Intel Stratix 10 device families; 614.4 Mbps (as before) for all other supported device families.
		  • Specified the Intel Stratix 10 device family supports only **TX/RX Duplex** operation mode in the 17.0.260 release.
		  • Moved **Core clock source input** parameter before **Transmitter local clock division factor** parameter.
		  • Specified that IP core variations that target an Intel Arria 10 or Intel Stratix 10 device, with **Line bit rate** set to the value of 4915.2 Mbps or lower, support only the value of **1** for the **Transmitter local clock division factor** parameter. This change is new in the 17.0.260 release.
		  • Specified that IP core variations that target a device family other than the Stratix V device family, support only the value of **1** for the **Number of receiver CDR reference clock(s)** parameter.
		  • Added new **VCCR_GXB and VCCT_GXB supply voltage for the transceiver** parameter. This parameter affects the IP core only if it targets an Intel Stratix 10 device.
		  • Specified the **Recovered clock source** parameter is not available for Intel Arria 10 and Intel Stratix 10 devices. IP cores that target either of these device families, support only a PCS clock source for the `xcvr_recovered_clk`.
		  • Specified the **Enable single-trip delay calibration** parameter is only available for Intel Arria 10 devices.
		  • Specified the **Enable line bit rate auto-negotiation** parameter is not
		• Added information about extended delay measurement for new Stratix 10 hard FIFOs. Refer to **Extended Delay Measurement** on page 3-56 and the new section **Extended Delay Measurement for Intel Stratix 10 Hard FIFOs** on page 3-58. The new feature adds the following new parameters and registers:

Date	Compatible ACDS Version	Changes

2016.07.22	16.0	• Updated for 16.0 software release. Updated release information and resource utilization numbers for the 16.0 software release.
		• Updated speed grade support table in *CPRI v6.0 IP Core Performance: Device and Transceiver Speed Grade Support*.
		• Added Arria 10 and Stratix V device family support for CPRI line bit rate of 8.11008 Gbps. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258.
		• Removed Arria 10 device family support for CPRI line bit rate of 0.6144 Gbps. This IP core change occurred in the CPRI v6.0 IP core release 15.1.313.
		• Added support for single-trip delay calibration. This IP core change is new in the CPRI v6.0 IP core release 16.0.
		  • Added new parameters **Core clock source input** and **Enable single-trip delay calibration**. Refer to *CPRI v6.0 IP Core Parameters* and *Running the Testbench*.
		  • Added new clocking structure, new `IOPLL` and `DPCU` blocks, and new top-level signal `tx_clkout`. Refer to *CPRI v6.0 IP Core Clocking Structure*, *CPRI v6.0 IP Core Management Interfaces*, *Adding the Off-Chip Clean-Up PLL*, and new section *Example CPRI v6.0 Clock Connections in Different Clocking Modes*.
		  • Added new IP core signals `cal_status[1:0]` and `cal_ctrl[15:0]`. Refer to *Adding and Connecting the Single-Trip Delay Calibration Blocks* and *Single-Trip Latency Measurement and Calibration Interface Signals*.
		  • Added new registers `DELAY_CAL_STD_CTRL1`, `DELAY_CAL_STD_CTRL2`, `DELAY_CAL_STD_CTRL3`, `DELAY_CAL_STD_CTRL4`, `DELAY_CAL_STD_CTRL5`, and `DELAY_CAL_STD_STATUS`. Refer to *CPRI v6.0 IP Core Registers*.
		  • Added description of new features in new sections *Delay Calibration Features* and *Single-trip Delay Calibration*.
		• Added support for round-trip latency calibration. This IP core change is new in the CPRI v6.0 IP core release 16.0.
		  • Added new parameters **Enable round-trip delay calibration** and **Round-trip delay calibration FIFO depth**. Refer to *CPRI v6.0 IP Core Parameters* and *Running the Testbench*.
		  • Added new register `DELAY_CAL_RTD`. Refer to *DELAY_CAL_RTD Register*.
		  • Added description of new features in new section *Round-Trip Delay Calibration*.

Date	Compatible ACDS Version	Changes

		• Added parameters to control Avalon-MM CPU interface addressing mode and to enable ADME support. Refer to *CPRI v6.0 IP Core Parameters*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258.
		  • Previously the CPU interface addressing mode was always word (4-byte) addressing mode. For CPU interface addressing mode changes, refer to *CPU Interface to CPRI v6.0 IP Core Registers* and *CPU Interface Signals*.
		  • You can turn on ADME support to support debugging through the Altera System Console and to expose transceiver registers. This parameter is available only in CPRI v6.0 IP cores that target an Arria 10 device. For additional information about this parameter, refer to *Arria 10 Transceiver Reconfiguration Interface*.
		• The transceiver reconfiguration interface is no longer available in certain configurations of the CPRI v6.0 IP core. Refer to *CPRI v6.0 IP Core Parameters*, *Arria V, Arria V GZ, Cyclone V, and Stratix V Transceiver Reconfiguration Interface* and *Arria 10 Transceiver Reconfiguration Interface*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258.
		• The `xcvr_rx_is_lockedtodata` signal is now available whether or not you turn on **Enable debug interface** in the parameter editor. Refer to *Transceiver Debug Interface* and *CPRI v6.0 IP Core Transceiver and Transceiver Management Signals*. This IP core change is new in the CPRI v6.0 IP core release 16.0.
		• Expanded width of `round_trip_delay` field in `ROUND_TRIP_DELAY` register from 10 bits to 20 bits. Refer to *ROUND_TRIP_DELAY Register*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.313.
		• Added new top-level signals `tx_analogreset_ack` and `rx_analogreset_ack`. These signals are available in Arria 10 variations with **Enable line bit rate auto-negotiation** turned on. Refer to *Auto-Rate Negotiation* and *CPRI v6.0 IP Core Management Interfaces*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258.
		• Changed name of `cpri_10g_coreclk` input clock signal to `cpri_coreclk`. Refer to *CPRI v6.0 IP Core Clocking Structure* and *CPRI v6.0 IP Core Management Interfaces*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258.
		• Removed `xcvr_reset_tx` and `xcvr_reset_rx` signals. Refer to *Adding the Reset Controller* and *CPRI v6.0 IP Core Transceiver and Transceiver Management Signals*.
		• Specified the **Recovered clock source** parameter is no longer available for CPRI master IP cores. This IP core change is new in the CPRI v6.0 IP core release 16.0.

Date	Compatible ACDS Version	Changes
		• Specified the `xcvr_recovered_clk` output clock is available only in CPRI slave IP cores. Refer to *CPRI v6.0 IP Core Clocking Structure* and *Main Transceiver Clock and Reset Signals*. This IP core change occurred in the CPRI v6.0 IP core release 15.1.258. • Corrected direction of multiple Transceiver Reset Controller signals in Required Connections to and From Reset Controllers in CPRI v6.0 Design table in *Adding the Reset Controller*. • Fixed assorted typos and minor errors.
2015.09.29	15.0	• Corrected RX GMII Timing Diagram figure. Refer to *Gigabit Media Independent Interface (GMII) to External Ethernet Block*. • Clarified that the limitation of the testbench to CPRI line bit rates other than 614.4 Mbps only applies for Arria 10 devices. All other device families support a testbench for a DUT with the CPRI line bit rate of 614.4 Mbps. Refer to *Running the Testbench*.

2015.09.25	15.0	• Changed document part number from UG-01156 to UG-20008. The document title is not affected.
		• Added support for Arria V GT, Arria V GX, Cyclone V GT, and Cyclone V GX devices.
		• Updated for 15.0 software release. Updated release information and resource utilization numbers for the 15.0 software release.
		• Modified device speed grade recommendations:
		  • Added supported transceiver speed grade information.
		  • Added device and transceiver speed grade support information for the previously unsupported device families.
		• Added support for new GMII interface to external Ethernet block. Refer to *Gigabit Media Independent Interface (GMII) to External Ethernet Block*.
		• Expanded *Deterministic Latency and Delay Measurement and Calibration* section. Removed latency numbers and examples, which are now available on the Altera wiki.

Date	Compatible ACDS Version	Changes
		• Updated IP core parameters. Refer to *CPRI v6.0 IP Core Parameters*. • Added new **Operation mode** parameter. This parameter determines whether the IP core is in TX simplex, RX simplex, or duplex mode. Note the old **Operation mode** parameter from previous releases is renamed . • Renamed the old **Operation mode** parameter to **Synchronization mode**. This parameter determines the default clocking mode of the IP core (Master or Slave clocking mode). Unfortunately, the `operation_mode` field of the `LI_CONFIG` register is not renamed. This field supports dynamic reconfiguration of the IP core clocking mode. Refer to *L1_CONFIG Register*. • Added new **Transmitter local clock division factor** parameter. This parameter enables you to include multiple instances of the CPRI v6.0 IP core with different CPRI line bit rates using the same external transceiver TX PLL. • Added new **Number of receiver CDR reference clock(s)** parameter. This parameter supports Stratix V variations in auto-negotiation to or from the CPRI line bit rate of 10.1376 Gbps. • Added new **Recovered clock source** parameter. This parameter supports auto-negotiation in Stratix V variations to or from the CPRI line bit rate of 10.1376 Gbps. • Changed name of **Bit rate (Mbit/s)** parameter to **Line bit rate (MBit/s)**. Enhanced description to include new supported devices. • Changed name of **Supported receiver CDR frequency (MHz)** parameter to **Receiver CDR reference clock frequency (MHz)**. • Changed name of **Receiver FIFO depth** parameter to **Receiver soft buffer depth**. • Changed name of **Enable auto-rate negotiation** parameter to **Enable line bit rate auto-negotiation**. • Changed name of **Enable auto-rate negotiation down to 614.4 Mbps** parameter to **Enable line bit rate auto-negotiation down to 614.4 Mbps**. • Changed name of **Supported CPU interface standard** parameter to **Management (CSR) interface standard**. • Changed name of **Auxiliary latency cycle(s)** parameter to **Auxiliary and direct interfaces write latency cycle(s)**. • Changed name of **Enable all control word access** parameter to **Enable all control word access via management interface**. • Changed name of **Enable Z.130.0 access interface** parameter to **Enable direct Z.130.0 alarm bits access interface**. • Changed name of **Enable real-time vendor specific interface (R-16A)** parameter to **Enable direct real-time vendor specific interface** . • Changed name of **Enable L1 inband protocol negotiator** parameter to **Enable protocol version and C&M channel setting auto-negotiation**. • Changed order of some L1 Feature parameters to reflect their new order in the CPRI v6.0 parameter editor. • Changed name of **Enable direct HDLC serial interface** parameter to **Enable HDLC serial interface**. • Changed name of **Enable IEEE 802.3 100BASE-X 100Mbps MII** On/Off parameter to **Ethernet PCS interface** multi-value parameter, and added "None" and "MII", which are still available. • Added allowed value of 11 for **L2 Ethernet PCS Tx/Rx FIFO depth** parameter, increasing the maximum L2 Ethernet buffer depth to 2048. • Changed names of loopback-enable parameters to include "serial" or "parallel."

Date	Compatible ACDS Version	Changes
		• Appended "_n" to names of active low reset signals. • Modified recommended reset connections and added four new IP core reset signals: `reset_tx_n`, `reset_rx_n`, `xcvr_reset_tx`, and `xcvr_reset_rx`. • Fixed assorted typos and minor errors.
2015.02.16	14.0 and 14.0 Arria 10 Edition	• Corrected name of `ROUND_TRIP_DELAY` register at offset 0x058. The register name was previously listed incorrectly as `ROUND_DELAY`. • Corrected names of `rx_hfnsync` and `rx_hfnsync_hold` fields of `L1_STATUS` register at offset 0x04. The fields were previously listed incorrectly as `rx_state` and `rx_state_hold`. • Fixed assorted typos. **Note:** This version of the user guide documents the same IP core version that the 2014.08.18 user guide documents.
2014.08.18	14.0 and 14.0 Arria 10 Edition	Initial release for Arria 10 device support. Corrected multiple figures associated with the AUX interface synchronization signals and the **Auxiliary latency cycle(s)** parameter. Added discussion of external clean-up PLL in Getting Started chapter. Added multiple sections to Functional Description chapter, including section on latency. Added resource utilization numbers. Moved detailed signal descriptions into relevant sections in Functional Description chapter; the Signals chapter is now a port listing summary. Corrected assorted errors and typos.
2014.07.28	14.0	Preliminary restricted document release.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Altera:

IP-CPRI-V6