Volume 2: Transceivers

Subscribe

Send Feedback

AV-5V3
2020.05.29

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

now part of Intel

Contents

Transceiver Custom Configurations in Arria V Devices....5-1

Transceiver Configurations in Arria V GZ Devices.... 6-1

Transceiver Architecture in Arria V Devices 1

2020.05.29

AV53001 Subscribe Send Feedback

Describes the Arria® V transceiver architecture, channels, and transmitter and receiver channel datapaths.

Altera® 28-nm Arria V FPGAs provide integrated transceivers with the lowest power requirement at 12.5-, 10-, and 6-Gigabits per second (Gbps). These transceivers comply with a wide range of protocols and data rate standards.

Table 1-1: Arria V Variants

Variants	Hard Processor System (HPS)	Up to 6.5536 Gbps	Beyond 6.5536 Gbps
GX	N/A	Backplane	N/A
GT	N/A	Backplane	N/A
GZ	N/A	Backplane	Up to 12.5 Gbps with backplane support
SX	Yes	Backplane	N/A
ST	Yes	Backplane	N/A

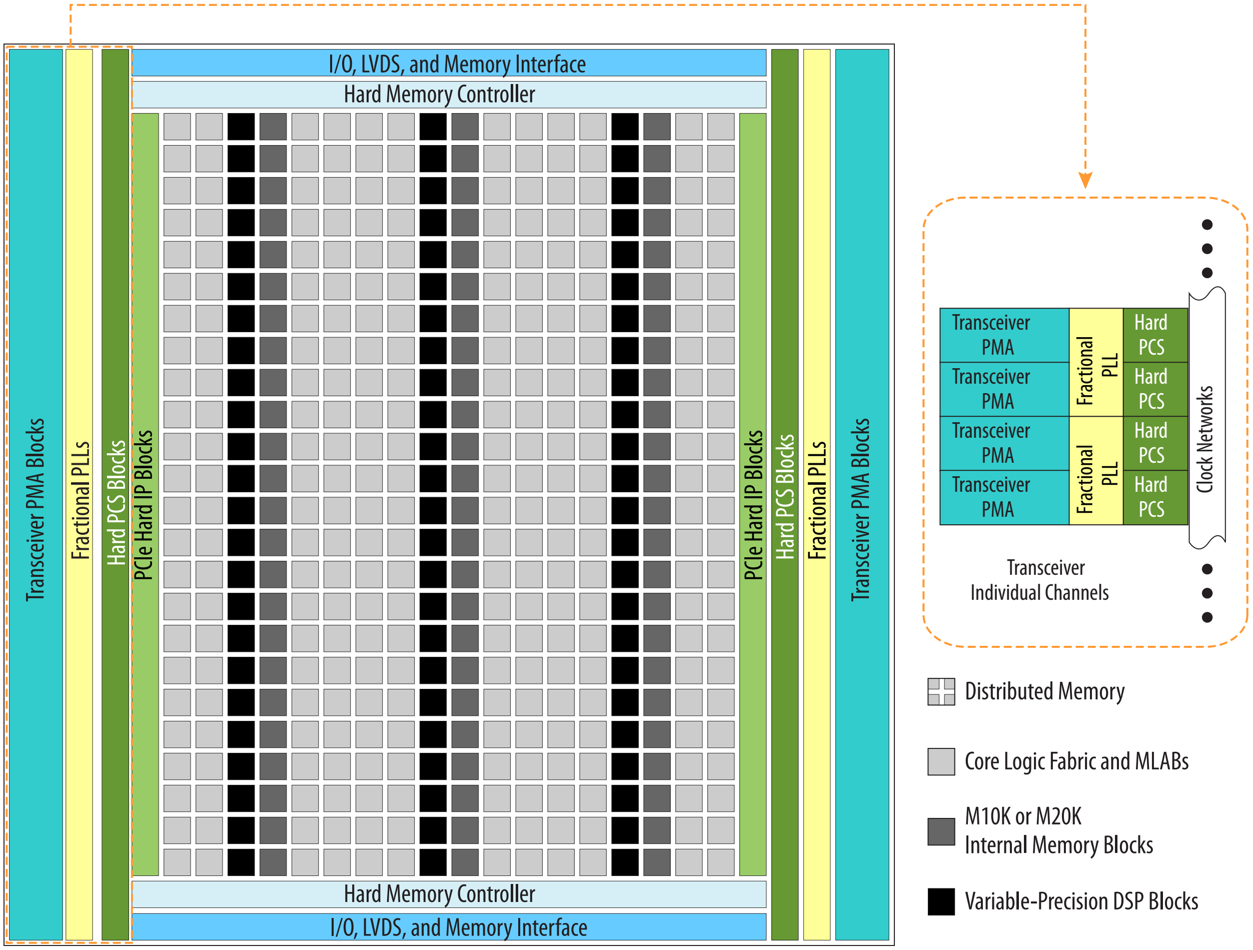
Related Information

Arria V Device Handbook: Known Issues

Lists the planned updates to the *Arria V Device Handbook* chapters.

Architecture Overview

Figure 1-1: Basic Layout of Transceivers in Arria V Devices



Notes:

1. This figure represents one variant of an Arria V device. Other variants may have transceivers and PCI Express (PCIe) hard IP only on the left side of the device.
2. This figure is a graphical representation of a top view of the silicon die, which corresponds to a reverse view for flip chip packages.

The Arria V hard IP for PCIe implements the PCIe protocol stack including the following layers:

- Physical interface/media access control (PHY/MAC) layer
- Data link layer
- Transaction layer

The embedded hard IP saves significant FPGA resources, reduces design risk, and reduces the time required to achieve timing closure. The hard IP complies with the PCI Express Base Specification 1.1, 2.0, and 3.0 for Gen1, Gen2, and Gen3 signaling data rates, respectively. In addition, the Arria V GZ variant supports PCI Express Base Specification 3.0 for Gen3 signaling datarates.

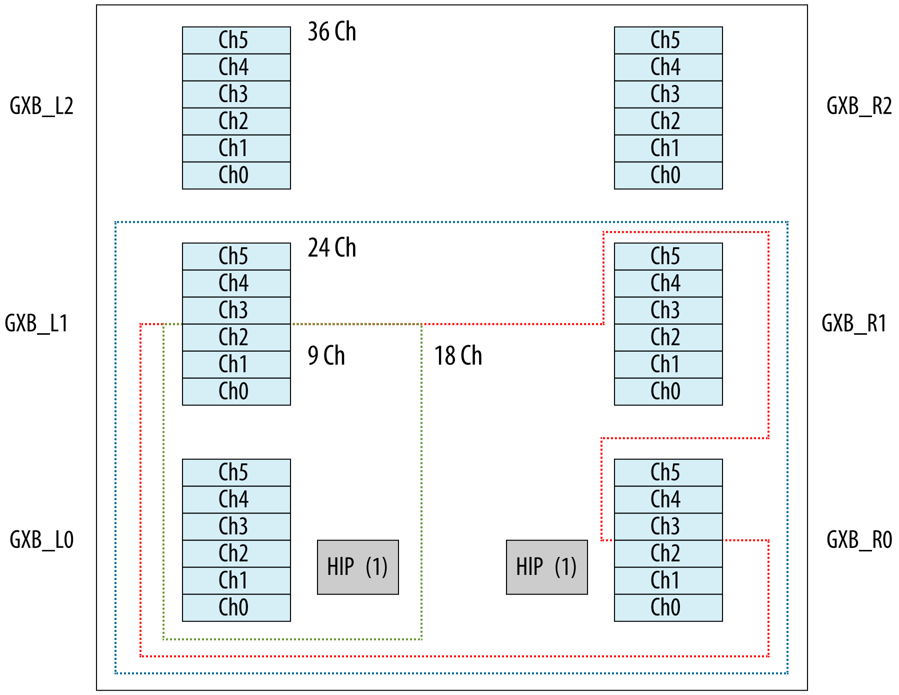
Related Information

For more information about the PCIe hard IP block architecture, see the Arria V Hard IP for PCI Express User Guide.

Transceiver Banks

The columns of Arria V transceivers are categorized in banks of six channels. The transceiver bank boundaries are important for clocking resources, bonding channels, and fitting. In some package variations, some transceiver banks are reduced to three channels. In the Arria V GX/GT/SX/ST, there are fundamentally two types of transceiver channels; 6-Gbps and 10-Gbps. By contrast, every Arria V GZ transceiver channel supports operation up to 12.5 Gbps data rates.

Figure 1-2: Transceiver Bank and PCIe Hard IP Location for GX Devices (1), (2)

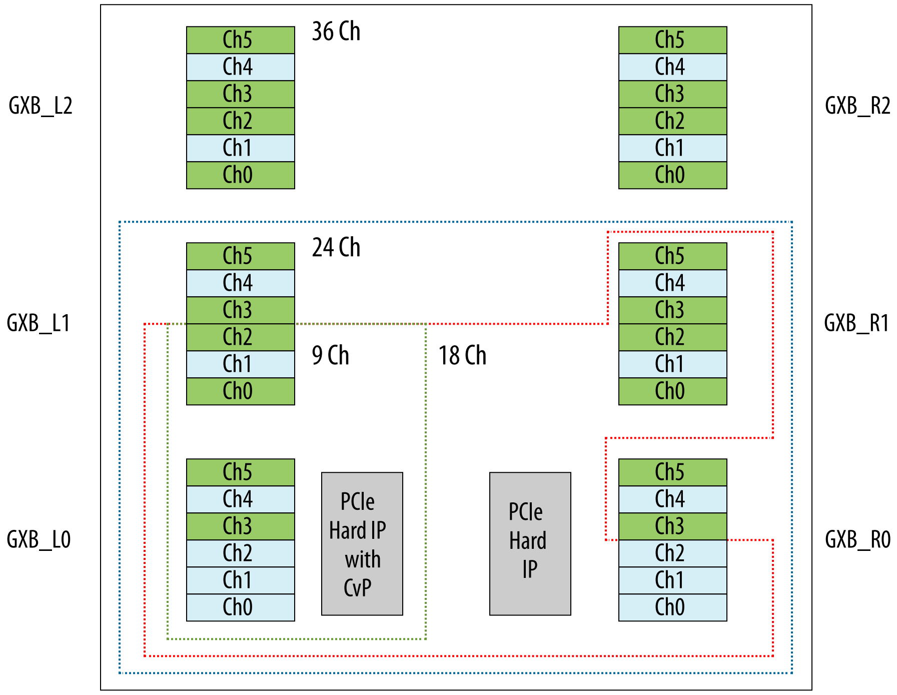


Notes:
1. PCIe HIP availability varies with device variants.
2. Blue blocks are 6 Gbps channels.

Table 1-2: Hard IP and Channel Resources in GX Variants

GX Variants	Left Hard IP	Right Hard IP	Total Channels
Base	None	None	9
Mainstream	1	None	9, 18
Extended Feature	1	1	24, 36

Figure 1-3: Transceiver Bank and PCIe Hard IP Location for GT Devices

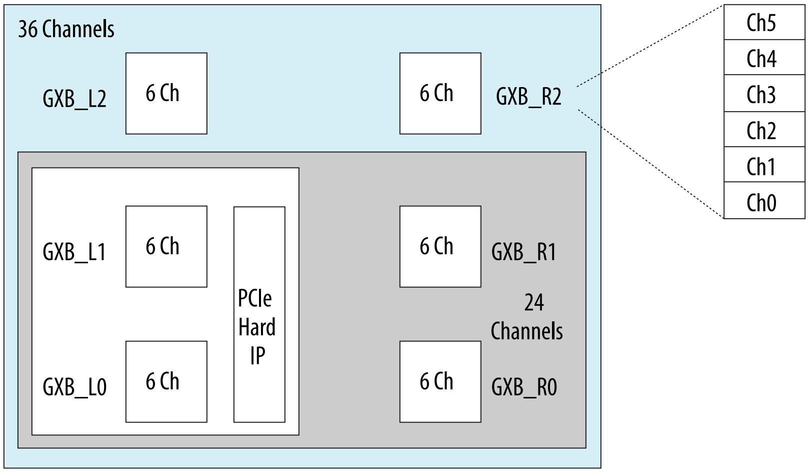


Notes:
1. Green blocks are 10-Gbps channels.
2. Blue blocks are 6-Gbps channels.

Table 1-3: Hard IP and Channel Resources in GT Variants

GT Variants	Left Hard IP	Right Hard IP	Total Channels
Mainstream	1	None	9, 18
Extended Feature	1	1	24, 36

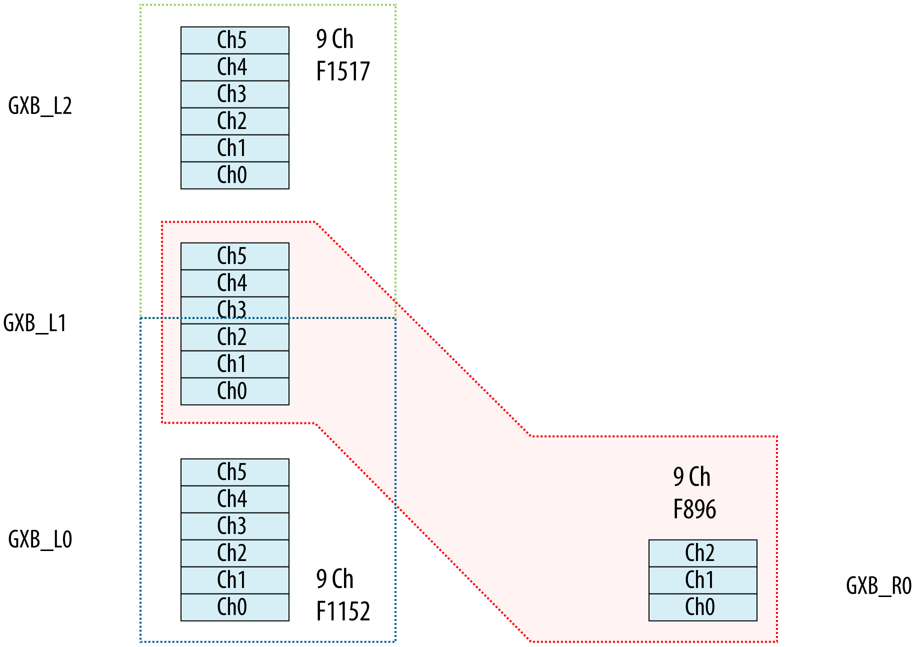
Figure 1-4: Transceiver Bank and PCIe Hard IP Location for GZ Devices



Notes:
1. 12-channel devices use banks L0 and L1.
2. All channels capable of backplane support up to 12.5 Gbps.

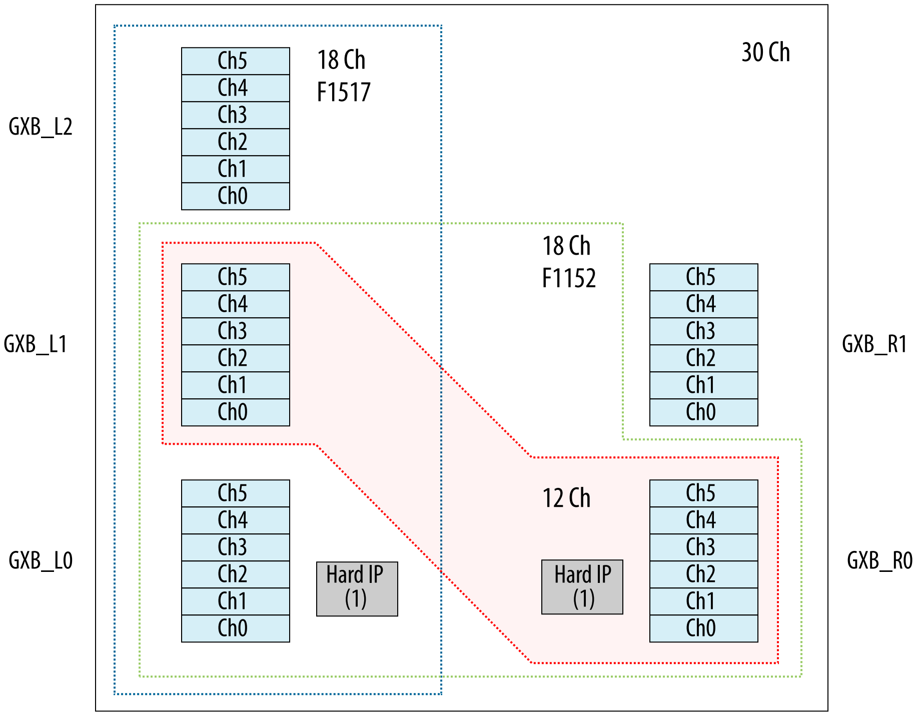
Figure 1-5: Transceiver Bank Location for SX Devices (9 channels)

SX devices with 9 channels do not have PCIe Hard IP blocks.



Note: Blue blocks are 6 Gbps channels.

Figure 1-6: Transceiver Bank and PCIe Hard IP Location for SX Devices (12,18, 30 channels) (1)

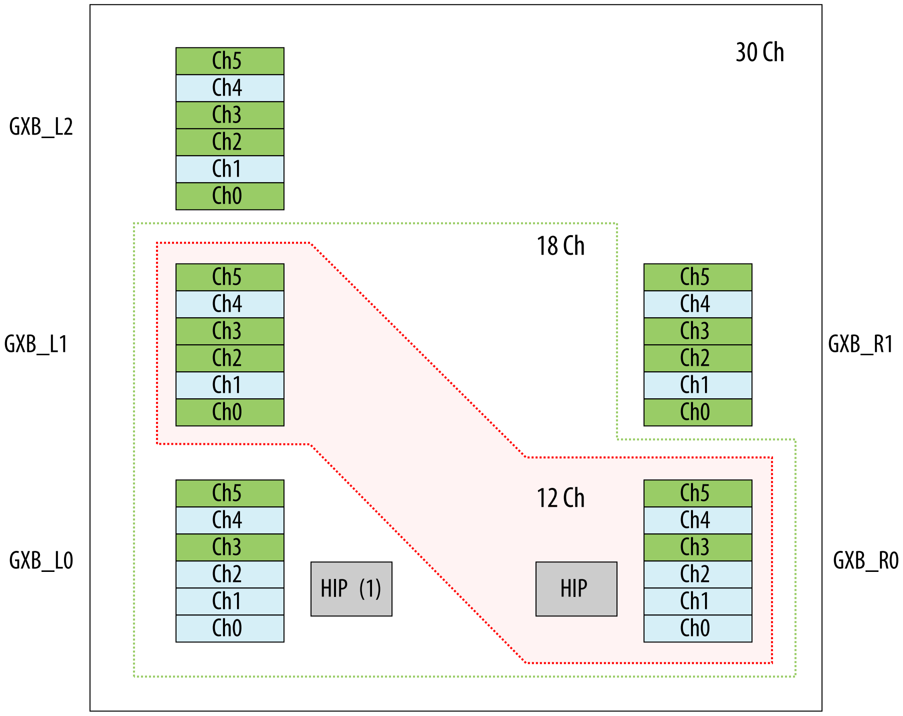


Notes:
1. PCIe Hard IP availability varies with device variants.
2. Blue blocks are 6 Gbps channels.

Table 1-4: Hard IP and Channel Resources in SX Variants

SX Variants	Left Hard IP	Right Hard IP	Total Channels
Mainstream	None	1	12
	1	None	18 (F1517 package)
Extended Feature	1	1	18 (F1152 package)
	1	1	30

Figure 1-7: Transceiver Bank and PCIe Hard IP Location for ST Devices[1], [2], [3]



Notes:
1. PCIe HIP availability varies with device variants.
2. Green blocks are 10-Gbps channels.
3. Blue blocks are 6-Gbps channels. With the exception of Ch0 to Ch2 in GXB_L0 and GXB_R0, the 6-Gbps channels can be used for TX-only or RX-only 10-Gbps channels.

Table 1-5: Hard IP and Channel Resources in ST Variants

ST Variants	Left Hard IP	Right Hard IP	Total Channels
Mainstream	None	1	12
Extended Feature	1	1	18
	1	1	30

Table 1-6: Usage Restrictions on Specific Channels Across Device Variants

Device Variants	Channel Location	Usage Restriction
GX, SX	Ch1, Ch2 of GXB_L0[1] Ch1, Ch2 of GXB_R0[1]	No support for PCS with phase compensation FIFO in registered mode (for example, CPRI or deterministic latency)

[1] The PMA clock of Channel 1 and Channel 2 of GBX_L0 and GXB_R0 cannot be routed out of the FPGA fabric for Arria V GX, GT, ST, and SX devices.

Device Variants	Channel Location	Usage Restriction
ST, GT	Ch1, Ch2 of GXB_L0 and GXB_R0[(1)]	No support for PCS with phase compensation FIFO in registered mode (for example, CPRI or deterministic latency)
ST	Ch0, Ch1, Ch2 of GXB_L0 and GXB_R0	No PMA Direct Support
GT	Ch0, Ch1, Ch2 of GXB_L0 and GXB_R0[(1)]	No PMA Direct Support

10-Gbps Support Capability in GT and ST Devices

Arria V GT/ST devices support up to four full duplex 10-Gbps channels in each transceiver bank. The bottom transceiver banks, and transceiver banks with only three transceiver channels, support up to two full duplex 10-Gbps channels.

Enhanced Small Form-Factor Pluggable (SFP+) Interface

Arria V GT devices are compliant to SFF 8431 with considerations to the number of channel requirements and board designs.Contact your local Altera sales representative for details.

10GBASE-KR Support

For 10GBASE-KR support, please contact Altera.

9.8 Gbps CPRI Application

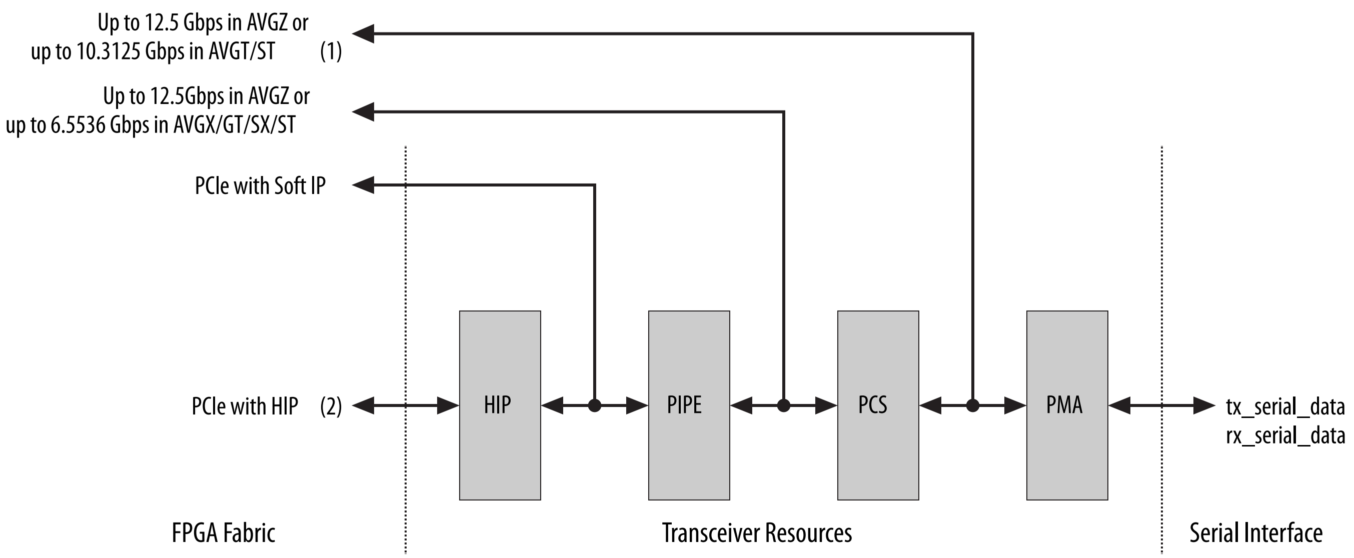
For 9.8 Gbps CPRI support, please contact Altera.

Transceiver Channel Architecture

The Arria V transceivers are comprised of a transmitter and receiver that can operate individually or simultaneously—providing a full-duplex physical layer implementation for high-speed serial interfacing. Each transmitter and receiver are divided into two blocks: PMA and PCS. The PMA block connects the FPGA to the channel, generates the required clocks, and converts the data from parallel to serial or serial to parallel. The PCS block performs digital processing logic between the PMA and the FPGA core.

Multiply the interface speed together with the serialization factor to determine the maximum supported data rate for any given transceiver configuration. For example, the Arria V GT supports a maximum interface speed of 161 MHz in PMA direct mode. To calculate the maximum supported data rate for a serialization factor of 20 in PMA direct mode, multiply 161 x 20 = 3220 Mbps.

Figure 1-8: Full Duplex Channel Interface Architecture



Notes:
1. 10-Gbps channel is available in GT and ST variants.
2. See the Related Information for specific channels that support interfaces with the HIP.
3. GX and GT can support up to 6.5536 Gbps.
4. GZ can support up to 12.5 Gbps.

Table 1-7: Architecture Differences Between 6- and 10-Gbps Arria V GT/ST Channels and Arria V GZ Channels

Architecture Differences	6-Gbps Channel	10-Gbps Channel[2]	Arria V GZ Channel
Transmitter PCS Capability	Up to 6.5536 Gbps	Up to 6.5536 Gbps	Up to 12.5 Gbps
Receiver PCS Capability	Up to 6.5536 Gbps	Up to 6.5536 Gbps [3]	Up to 12.5 Gbps
Transmitter/Receiver PMA Capability	Up to 6.5536 Gbps	Up to 10.3125 Gbps	Up to 12.5 Gbps
PMA Direct (PMA-Fabric Interface)	Not supported	Supported	Supported
Serialization Factor	8, 10, 16, 20	8, 10, 16, 20, 64, 80	8, 10, 16, 20, 32, 40, 64, 80

Related Information

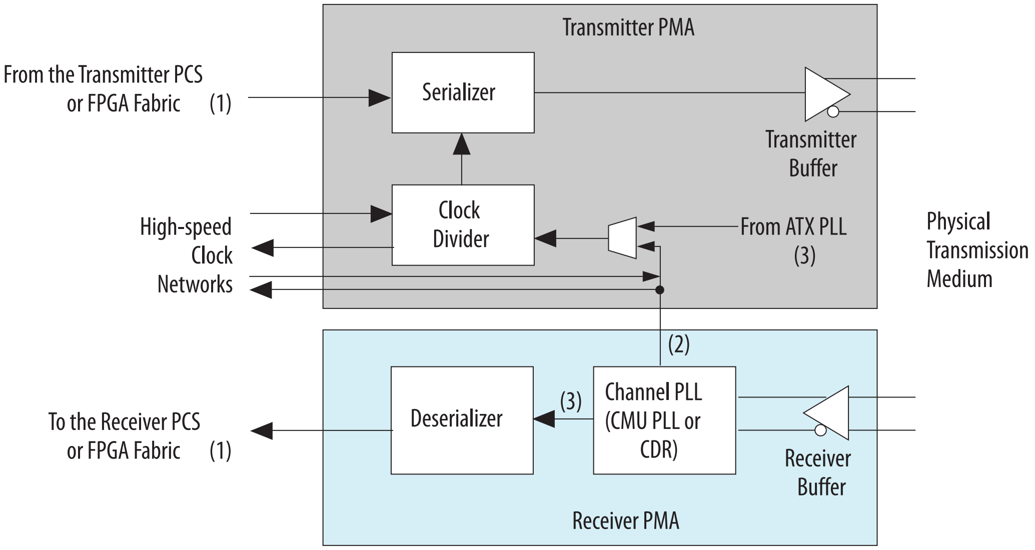
- **Arria V Hard IP for PCI Express User Guide**
- **For Transceiver-FPGA interface speed specifications, refer to the Arria V Device Datasheet.**

[2] 10-Gbps channel is only available in GT and ST variants.
[3] Arria V GT/ST devices cannot use the PCS when running at 10 Gbps.

PMA Architecture

The PMA includes the transmitter and receiver datapaths, CMU PLL (configured from the channel PLL), the ATX PLL, and the clock divider. The analog circuitry and differential on-chip termination (OCT) in the PMA requires the calibration block to compensate for process, voltage, and temperature variations (PVT).

Figure 1-9: Transceiver Channel PMA for Arria V Devices


Transmitter PMA
From the Transmitter PCS or FPGA Fabric (1)
Serializer
Transmitter Buffer
High-speed Clock Networks
Clock Divider
From ATX PLL (3)
Physical Transmission Medium
(2)
To the Receiver PCS or FPGA Fabric (1)
Deserializer
(3)
Channel PLL (CMU PLL or CDR)
Receiver Buffer
Receiver PMA

Notes:
1. The channel PLL provides the serial clock when configured as a CMU PLL.
2. The channel PLL recovers the clock and serial data stream when configured as a CDR.
3. ATX PLL available only in GZ devices.

Transmitter PMA Datapath

Table 1-8: Functional Blocks in the Transmitter PMA Datapath

Block	Functionality
Serializer	• Converts the incoming low-speed parallel data from the transmitter PCS to high-speed serial data and sends the data LSB first to the transmitter buffer. • Supports the optional polarity inversion and bit reversal features. • Supports 8, 10, 16, and 20-bit serialization factors in Arria V GX, SX, GT, ST, and GZ devices. • Additionally supports 64 and 80-bit serialization factors for 10-Gbps transceiver channels in Arria V ST and GT devices. • Additionally supports 32, 40, 64, and 80-bit serialization factors in Arria V GZ devices.

Block	Functionality
Transmitter Buffer	• The 1.4-V (Arria V GZ Only) and 1.5-V pseudo current mode logic (PCML) output buffer conditions the high-speed serial data for transmission into the physical medium. Arria GZ supports 1.4 and 1.5V PCML. • Supports the following features: • Programmable differential output voltage (VOD) • Programmable pre-emphasis • Programmable V_{CM} current strength • Programmable slew rate • On-chip biasing for common-mode voltage (TX V_{CM}) • Differential OCT (85, 100, 120 and 150 Ω) • Transmitter output tristate • Receiver detect (for the PCIe receiver detection function)

Serializer

The serializer provides parallel-to-serial data conversion and sends the data LSB first from the transmitter physical coding sublayer (PCS) to the transmitter buffer. Additionally, the serializer provides the polarity inversion and bit reversal features.

Polarity Inversion

The positive and negative signals of a serial differential link might accidentally be swapped during board layout.

The polarity inversion feature of the transmitter corrects this error without requiring a board respin or major updates to the logic in the FPGA fabric. The polarity inversion feature inverts the polarity of every bit at the input to the serializer, which has the same effect as swapping the positive and negative signals of the serial differential link.

Polarity inversion is controlled dynamically with the `tx_invpolarity` register. When you enable the polarity inversion feature, it may cause initial disparity errors at the receiver with 8B/10B-coded data. The downstream system at the receiver must be able to tolerate these disparity errors.

Caution: Enabling polarity inversion midway through a serialized word corrupts the word.

Bit Reversal

You can reverse the transmission bit order to achieve MSB-to-LSB ordering using the bit reversal feature at the transmitter.

Table 1-9: Bit Reversal Feature

	Transmission Bit Order	
Bit Reversal Option	8- or 10-bit Serialization Factor	16- or 20-bit Serialization Factor
Disabled (default)	LSB to MSB	LSB to MSB

	Transmission Bit Order	
Bit Reversal Option	8- or 10-bit Serialization Factor	16- or 20-bit Serialization Factor
Enabled	MSB to LSB For example: 8-bit—`D[7:0]` rewired to `D[0:7]` 10-bit—`D[9:0]` rewired to `D[0:9]`	MSB to LSB For example: 16-bit—`D[15:0]` rewired to `D[0:15]` 20-bit—`D[19:0]` rewired to `D[0:19]`

Transmitter Buffer

The transmitter buffer includes additional circuitry to improve signal integrity, such as the programmable differential output voltage (V_{OD}), programmable three-tap pre-emphasis circuitry, internal termination circuitry, and PCIe receiver detect capability to support a PCIe configuration.

Modifying programmable values within transmitter output buffers can be performed by a single reconfiguration controller for the entire FPGA, or multiple reconfiguration controllers if desired. Within each transceiver bank a maximum of two reconfiguration controllers is allowed; one for the three-transceiver triplet in the upper-half of the bank, and one for the lower-half. This is due to a single slave interface to all PLLs and PMAs within each triplet. Therefore, many triplets can be connected to a single reconfiguration controller, but only one reconfiguration controller can be connected to the three transceivers within any triplet.

Note: A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

Note: The Arria V GT transmitter buffer has only one tap for the pre-emphasis.

Figure 1-10: Transmitter Buffer in Arria V Devices

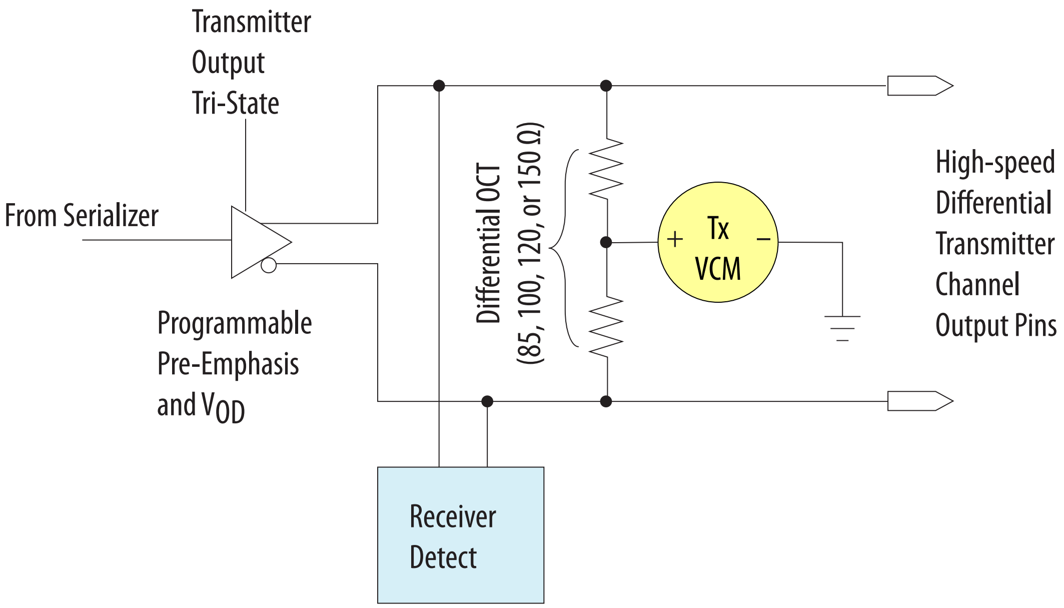


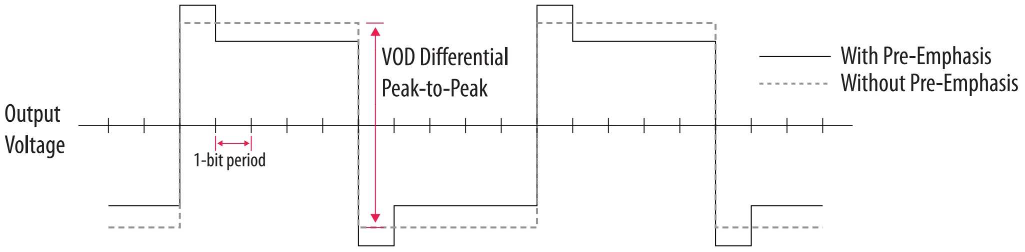
Table 1-10: Description of the Transmitter Buffer Features

Category	Features	Description
Improve Signal Integrity	Programmable Differential Output Voltage (V_{OD})	Controls the current mode drivers for signal amplitude to handle different trace lengths, various backplanes, and receiver requirements. The actual V_{OD} level is a function of the current setting and the transmitter termination value.
	Programmable Pre-Emphasis	Boosts the high-frequency components of the transmitted signal, which may be attenuated when propagating through the transmission medium. The physical transmission medium can be represented as a low-pass filter in the frequency domain. Variation in the signal frequency response that is caused by attenuation significantly increases the data-dependent jitter and other intersymbol interference (ISI) effects at the receiver end. Using the pre-emphasis feature maximizes the data opening at the far-end receiver. Arria V GZ channels provide three pre-emphasis taps: pre-tap (16 settings), first post-tap (32 settings), and second post-tap (16 settings). Arria V GX, SX, GT and ST provides only one pre-emphasis tap which is first post-tap (32 settings).The pre-tap sets the pre-emphasis on the data bit before the transition. The first post-tap and second post-tap set the pre-emphasis on the transition bit and the following bit, respectively. The pre-tap and second post-tap also provide inversion control, shown by negative values.
	Programmable Slew Rate	Controls the rate of change for the signal transition.
Save Board Space and Cost	On-Chip Biasing	Establishes the required transmitter common-mode voltage (TX V_{CM}) level at the transmitter output. The circuitry is available only if you enable on-chip termination (OCT). When you disable OCT, you must implement off-chip biasing circuitry to establish the required TX V_{CM} level.
	Differential OCT	The termination resistance is adjusted by the calibration circuitry, which compensates for the process, voltage, and temperature variations (PVT). You can disable OCT and use external termination. However, you must implement off-chip biasing circuitry to establish the required TX V_{CM} level. TX V_{CM} is tri-stated when using external termination.
Reduce Power	Programmable V_{CM} Current Strength	Controls the impedance of V_{CM}. A higher impedance setting reduces current consumption from the on-chip biasing circuitry.

Category	Features	Description
Protocol-Specific Function	Transmitter Output Tri-State	Enables the transmitter differential pair voltages to be held constant at the same value determined by the TX V_{CM} level with the transmitter in the high impedance state. The transmitter output tri-state feature is compliant with differential and common-mode voltage levels and operation time requirements for transmitter electrical idle, as specified in the PCI Express Base Specification 2.0 for Gen1 and Gen2 signaling rates, and PCI Express Base Specification 3.0 for Gen3 signaling rates (Arria V GZ only).
	Receiver Detect	Provides link partner detection capability at the transmitter end using an analog mechanism for the receiver detection sequence during link initialization in the Detect state of the PCI Express® (PCIe) Link Training and Status State Machine (LTSSM) states. The circuit detects if there is a receiver downstream by changing the transmitter V_{CM} to create a step voltage and measuring the resulting voltage rise time. For proper functionality, the series capacitor (AC-coupled link) and receiver termination values must comply with the PCI Express Base Specification 2.0 for Gen1 and Gen2 signaling rates and PCI Express Base Specification 3.0 for Gen3 signaling rates (Arria V GZ only). The circuit is clocked using `fixedclk` and requires that the transmitter OCT be enabled with the output tri-stated.

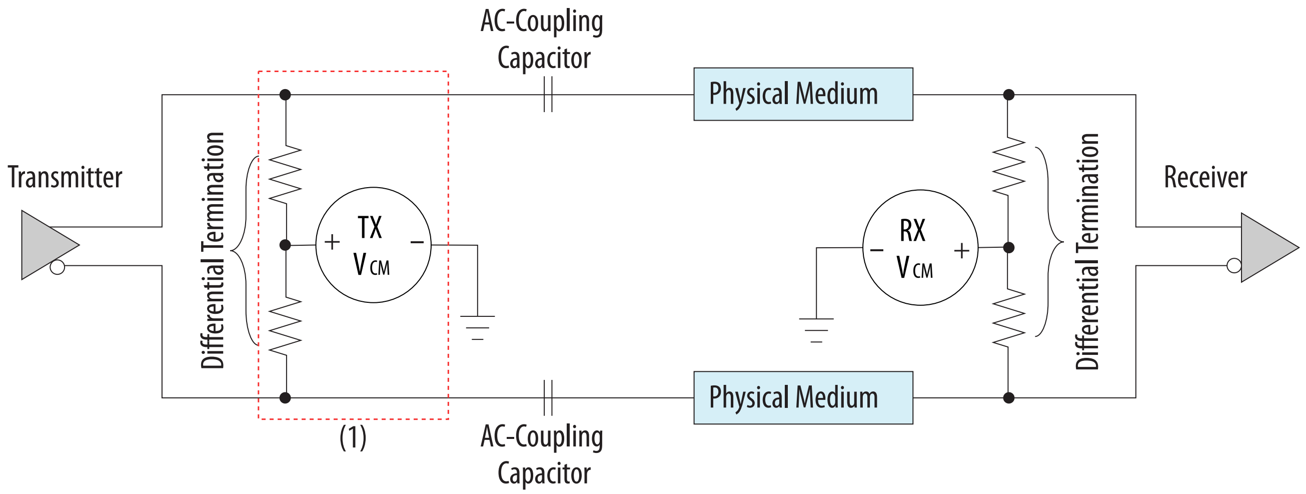
Figure 1-11: Example of Pre-Emphasis Effect on Signal Transmission at Transmitter Output

Shows the signal transmission at the transmitter output with and without applying pre-emphasis post-tap for a 5 Gigabit per second (Gbps) signal with an alternating data pattern of five 1s and five 0s.



The receiver can be AC- or DC-coupled to a transmitter. In an AC-coupled link, the AC-coupling capacitor blocks the transmitter V_{CM}. At the receiver end, the termination and biasing circuitry restores the V_{CM} level that is required by the receiver.

Figure 1-12: AC-Coupled Link with Arria V Transmitter



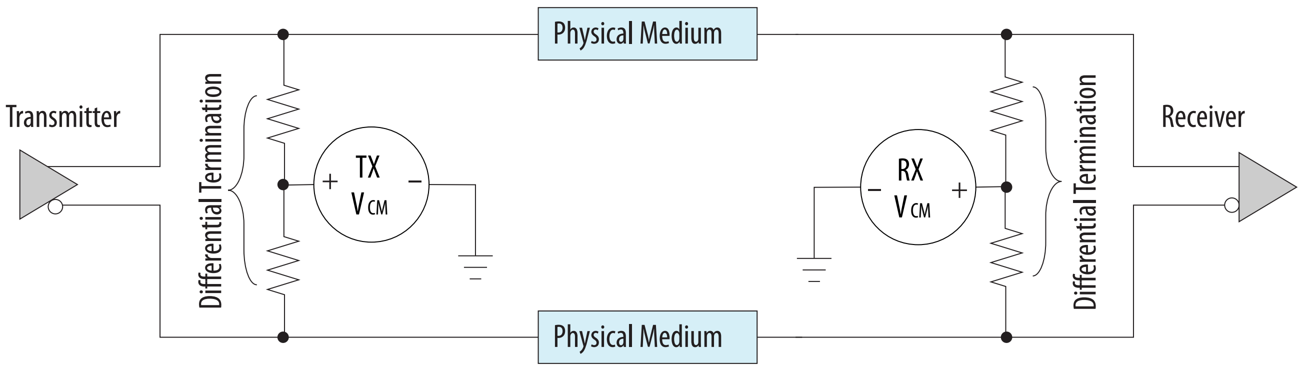
Notes:

1. When you disable OCT, you must implement external termination and off-chip biasing circuitry to establish the required TX V_{CM} level.

When used in a DC-coupled link, the transmitter V_{cm} is fixed to 0.7 V. The receiver V_{cm} is required to be at 0.7 V. DC coupling is supported for serial data rates up to 3.2 Gbps.

You can DC-couple the Arria V GZ channel transmitter only to another Arria V GZ channel receiver for the entire datarate range from 600 Mbps to 12.5 Gbps so long as the same V_{CM} value is observed.

Figure 1-13: DC-Coupled Link with Arria V Transmitter



Related Information

- Arria V Device Datasheet
- Altera Transceiver PHY IP Core User Guide

Receiver PMA Datapath

Describes the receiver buffer, channel phase-locked loop (PLL) configured for clock data recovery (CDR) operation, and deserializer blocks in the receiver PMA datapath.

Table 1-11: Functional Blocks in the Receiver PMA Datapath

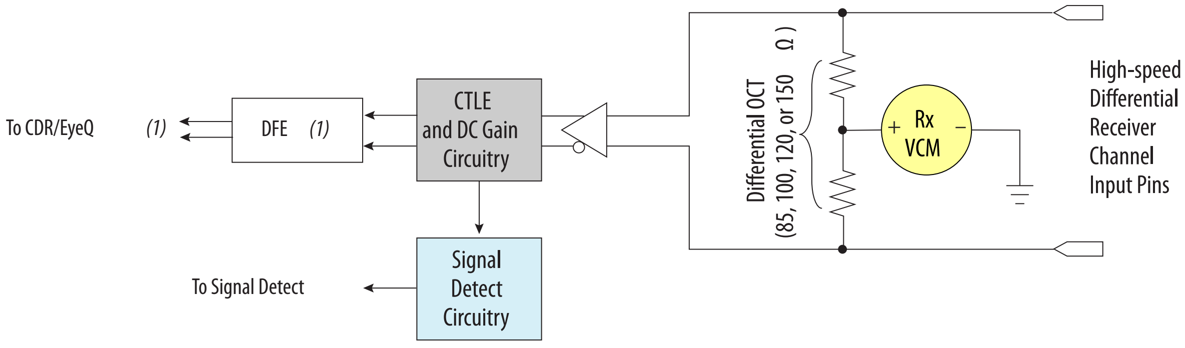
Block	Functionality
Receiver Buffer	• Receives the serial data stream and feeds the stream to the channel PLL if you configure the channel PLL as a CDR. • Supports the following features: • Programmable CTLE (Continuous Time Linear Equalization) • Programmable DC gain • Programmable V_{CM} current strength • On-chip biasing for common-mode voltage (RX V_{CM}) • I/O standard (1.4 V (Arria V GZ), **PCML**, 1.5 V **PCML**, 2.5 V **PCML**, **LVDS**, **LVPECL**) • Differential OCT (85, 100, 120 and 150 Ω) • Signal detect
Channel PLL	• Recovers the clock and serial data stream if you configure the channel PLL as a CDR. • Requires offset cancellation to correct the analog offset voltages. • If you do not use the channel PLL as a CDR, you can configure the channel PLL as a CMU PLL for clocking the transceivers. For more information about the channel PLL configured as a CMU PLL, refer to **CMU PLL** on page 1-27.
Deserializer	• Converts the incoming high-speed serial data from the receiver buffer to low-speed parallel data for the receiver PCS. • Receives serial data in LSB-to-MSB order. • Supports the optional clock-slip feature for applications with stringent latency uncertainty requirement. • Supports 8, 10, 16, and 20-bit deserialization factors in Arria V GX, SX, GT, ST, and GZ devices. • Additionally supports the 64 and 80-bit serialization factor for 10-Gbps transceiver channels Arria V ST and GT devices. • Additionally supports the 32, 40, 64, and 80-bit serialization factor in Arria V GZ devices.

Receiver Buffer

The receiver input buffer receives serial data from the `rx_serial_data` port and feeds the serial data to the channel PLL configured as a CDR PLL.

Figure 1-14: Receiver Buffer

Channel PLL configured as a CDR.



Note:

1. Available only in Arria V GZ devices. Arria V GX,SX,GT and ST devices do not have the decision feedback equalizer (DFE) feature.

Modifying programmable values within receiver input buffers can be performed by a single reconfiguration controller for the entire FPGA, or multiple reconfiguration controllers if desired. Within each transceiver bank a maximum of two reconfiguration controllers is allowed; one for the three-transceiver triplet in the upper-half of the bank, and one for the lower-half. This is due to a single slave interface to all PLLs and PMAs within each triplet. Therefore, many triplets can be connected to a single reconfiguration controller, but only one reconfiguration controller can be connected to the three transceivers within any triplet.

Note: A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

Receiver Analog Settings

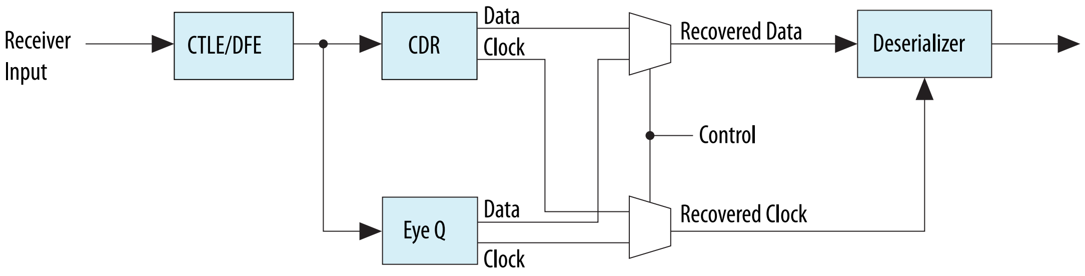
Arria V GZ channels have two receiver analog modes: half-bandwidth and full-bandwidth. The half-bandwidth data rate is up to 6.25 Gbps; the full-bandwidth data rate is from 6.25 Gbps to 12.5 Gbps. You can select which mode to use in the Assignment Editor of the Quartus II software (Receiver Equalizer Gain Bandwidth Select).

Table 1-12: Arria V Receiver Buffer Features

Category	Features	Description
Improve Signal Integrity	Programmable CTLE (Continuous Time Linear Equalization)	Boosts the high-frequency components of the received signal, which may be attenuated when propagating through the transmission medium. The physical transmission medium can be represented as a low-pass filter in the frequency domain. Variation in the signal frequency response that is caused by attenuation leads to data-dependent jitter and other ISI effects, causing incorrect sampling on the input data at the receiver. The amount of the high-frequency boost required at the receiver to overcome signal attenuation depends on the loss characteristics of the physical medium.
	Programmable DC Gain	Provides equal boost to the received signal across the frequency spectrum.
	Decision Feedback Equalization (DFE)	The decision feedback equalization feature consists of a 5-tap equalizer, which boosts the high frequency components of a signal without noise amplification by compensating for inter-symbol interference (ISI). There are two decision feedback equalization modes: manual and auto-adaptation. The DFE is supported only in Arria V GZ devices.
	EyeQ	The EyeQ feature is a debug and diagnosis tool that helps you analyze the received data by measuring the horizontal and vertical eye opening. The EyeQ is supported only in Arria V GZ devices, and not supported in Arria V GX, SX, ST and GT devices. There are two multiplexers for the data and clock which select one path to feed to the deserializer.
Save Board Space and Cost	On-Chip Biasing	Establishes the required receiver common-mode voltage (RX V_{CM}) level at the receiver input. The circuitry is available only if you enable OCT. When you disable OCT, you must implement off-chip biasing circuitry to establish the required RX V_{CM} level.
	Differential OCT	The termination resistance is adjusted by the calibration circuitry, which compensates for the PVT. You can disable OCT and use external termination. However, you must implement off-chip biasing circuitry to establish the required RX V_{CM} level. RX V_{CM} is tri-stated when using external termination.
Reduce Power	Programmable V_{CM} Current Strength	Controls the impedance of V_{CM}. A higher impedance setting reduces current consumption from the on-chip biasing circuitry. **Note:** There is no programmable option for Arria V GX, SX, GT and ST devices because only one V_{CM} value is offered for AC coupled link in non-PCIe mode.

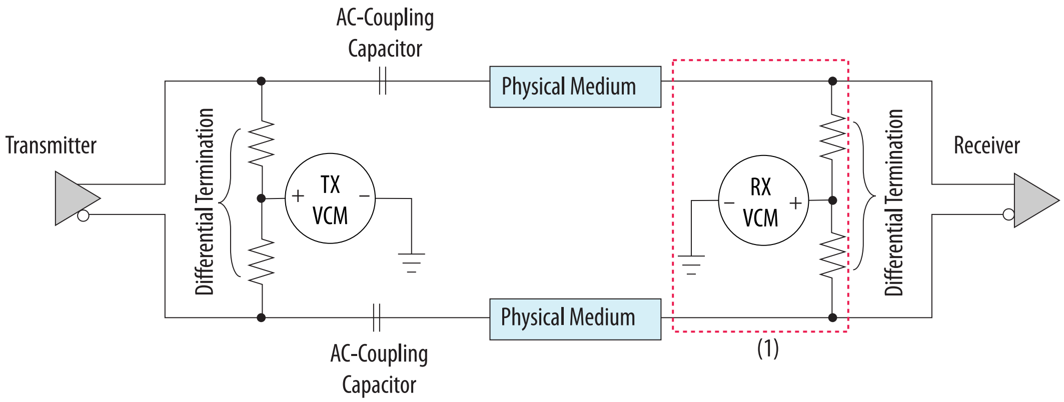
Category	Features	Description
Protocol-Specific Function	Signal Detect	Senses if the signal level present at the receiver input is above or below the threshold voltage that you specified. The detection circuitry has a hysteresis response that asserts the status signal only when a number of data pulses exceeding the threshold voltage are detected and deasserts the status signal when the signal level below the threshold voltage is detected for a number of recovered parallel clock cycles. The circuitry requires the input data stream to be 8B/10B-coded. Signal detect is compliant to the threshold voltage and detection time requirements for electrical idle detection conditions as specified in the PCI Express Base Specification 2.0 for Gen1 and Gen2 signaling rates and PCI Express Base Specification 3.0 for Gen3 signaling rates (Arria V GZ only).

Figure 1-15: Receiver and EyeQ Architecture



The receiver can be AC- or DC-coupled to a transmitter. In an AC-coupled link, the AC-coupling capacitor blocks the transmitter V_{CM}. At the receiver end, the termination and biasing circuitry restores the V_{CM} level that is required by the receiver.

Figure 1-16: AC-Coupled Link with Arria V Receiver

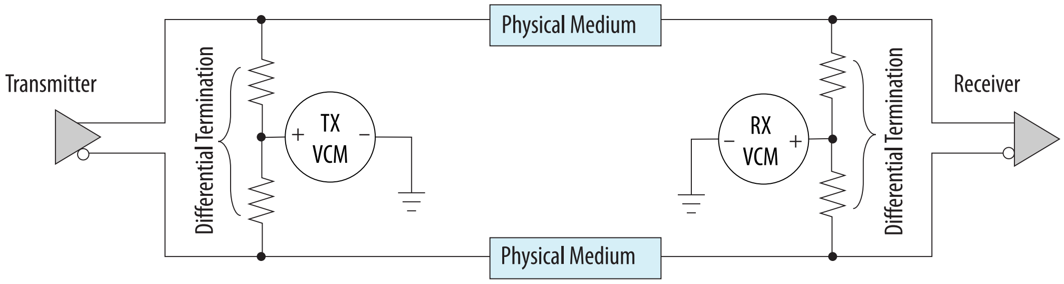


Note:
1. When you disable OCT, you must implement external termination and off-chip biasing circuitry to establish the required RX V_{CM} level.

When used in a DC-coupled link, the transmitter V_{cm} is fixed to 0.7V. The receiver V_{cm} is required to be at 0.7V. DC coupling is supported for serial data rates up to 3.2 Gbps.

You can DC-couple the Arria V GZ channel transmitter only to another Arria V GZ channel receiver for the entire datarate range from 600 Mbps to 12.5 Gbps so long as the same V_{CM} value is observed.

Figure 1-17: DC-Coupled Link with Arria V Receiver



Related Information

- For more information about the EyeQ feature, refer to the Altera Transceiver PHY IP Core User Guide.
- For more information about the Receiver Buffer and electrical specifications, refer to the Arria V Device Datasheet.

Continuous Time Linear Equalization (CTLE)

Each receiver buffer has five independently programmable equalization circuits that boost the high-frequency gain of the incoming signal, thereby compensating for the low-pass characteristics of the physical medium. The CTLE operates in two modes: manual mode and adaptive equalization (AEQ) mode

Manual Mode

Manual mode allows you to manually adjust the continuous time linear equalization to improve signal integrity. You can statically set the equalizer settings in the IP or you can dynamically change the equalizer settings with the reconfiguration controller IP.

Adaptive Equalization Mode

AEQ mode eliminates the need for manual tuning by enabling the Arria V device to automatically tune the receiver equalization settings based on the frequency content of the incoming signal and comparing that with internally generated reference signals. The AEQ block resides within the PMA of the receiver channel and is available on all GX channels.

Note: AEQ is supported by Arria V GZ devices, but not Arria V GX, GT, SX, and ST devices.

There are two AEQ modes: one-time and powerdown:

- One-time mode—The AEQ finds a stable setting of the receiver equalizer and locks to that value. After the stable setting is locked, the equalizer values do not.
- Powerdown mode—The AEQ of the specific channel is placed in standby mode and the CTLE uses the manually set value. Note that the CTLE cannot be bypassed.

You can dynamically switch between these modes.

Related Information

For more information about enabling different options and using them to control the AEQ hardware, see the Altera Transceiver PHY IP Core User Guide.

Channel PLL

If you configure the channel PLL as a CDR PLL, the channel PLL recovers the clock and data from the serial data stream. If you do not use the channel PLL as a CDR PLL, you can configure it as a clock multiplier unit (CMU) PLL for clocking the transceivers.

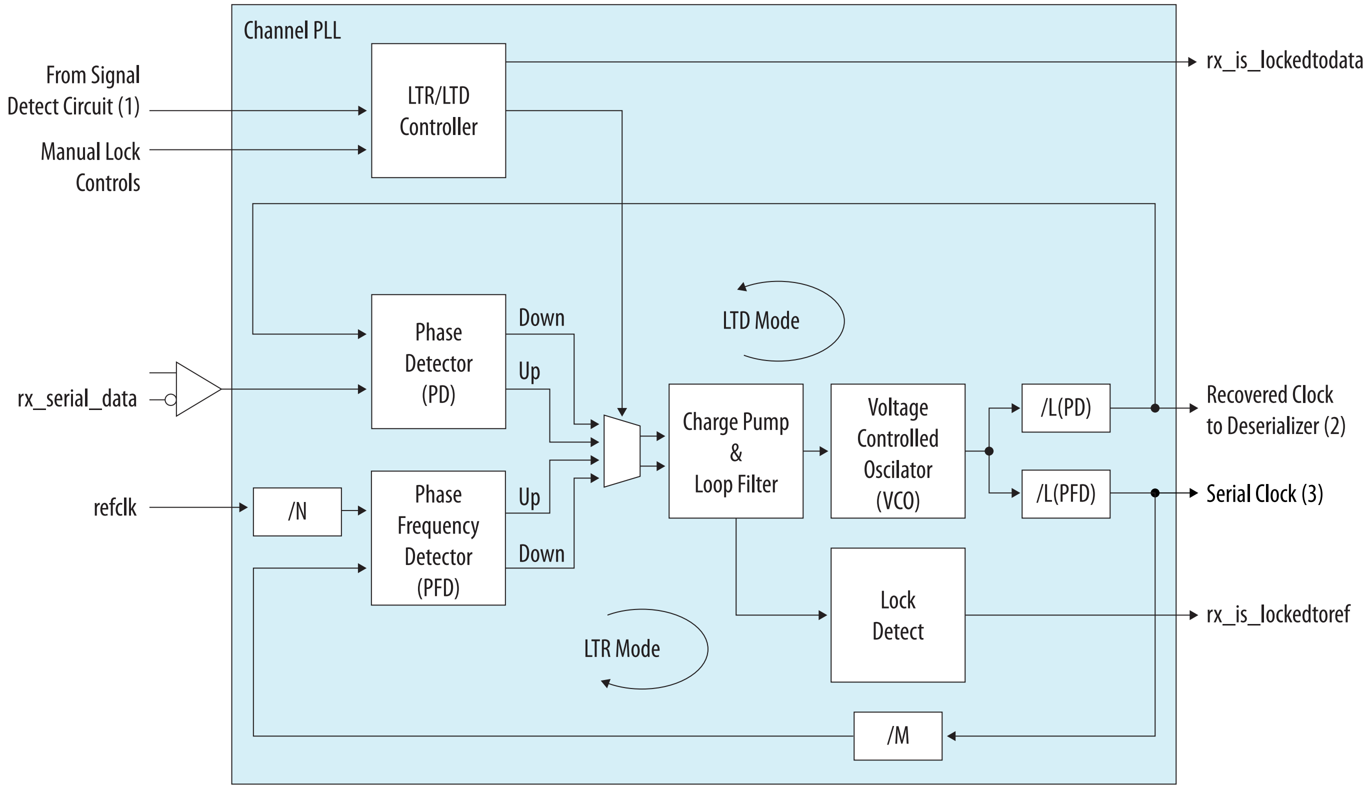
Related Information

CMU PLL on page 1-27
Refer to this section for more information about the channel PLL operation when configured as a CMU PLL.

Channel PLL Architecture

The channel PLL supports operation in either lock-to-reference (LTR) or lock-to-data (LTD) mode.

Figure 1-18: Channel PLL Block Diagram



Notes:

1. Applicable in a PCIe configuration only.
2. Applicable when configured as a CDR PLL.
3. Applicable when configured as a CMU PLL.
4. The PCIe rateswitch control allows dynamic switching between Gen3, Gen2, and Gen1 line rates in a PCIe Gen2 and Gen3 design. In addition, the Arria V GZ PCIe rateswitch control allows dynamic switching between Gen3, Gen2, and Gen1 line rates in a PCIe Gen3 design.

In LTR mode, the channel PLL tracks the input reference clock. The phase-frequency detector (PFD) compares the phase and frequency of the voltage controlled oscillator (VCO) output and the input reference clock. The resulting PFD output controls the VCO output frequency to half the data rate with the appropriate counter (M or L) value given an input reference clock frequency. The lock detect determines whether the PLL has achieved lock to the phase and frequency of the input reference clock.

During normal operation, the CDR must be in LTD mode to recover the clock from the incoming serial data. In LTD mode, the phase detector (PD) in the CDR tracks the incoming serial data at the receiver input. Depending on the phase difference between the incoming data and the CDR output clock, the PD controls the CDR charge pump that tunes the voltage controlled oscillator (VCO).

Note: The phase frequency detector (PFD) is inactive in LTD mode. `rx_is_lockedtoref` toggles randomly and is not significant in LTD mode.

Use the LTR/LTD controller only when you configure the channel PLL as a CDR PLL.

Channel PLL Counters

Table 1-13: Channel PLL Counters

The Quartus® II software automatically selects the appropriate counter values for each transceiver configuration.

Counter	Description	Values
N	Pre-scale counter to divide the input reference clock frequency to the PFD by the N factor	1, 2, 4, 8
M	Feedback loop counter to multiply the VCO frequency above the input reference frequency to the PFD by the M factor	1, 4, 5, 8, 10, 12, 16, 20, 25
L (PFD)	VCO post-scale counter to divide the VCO output frequency by the L factor in the LTR loop	1, 2, 4, 8
L (PD)	VCO post-scale counter to divide the VCO output frequency by the L factor in the LTD loop	1, 2, 4, 8

CDR PLL Operation

Description of Arria V CDR PLL operation modes.

The CDR PLL independently recovers the clock and data from the incoming serial data and sends the clock and data to the deserializer. The CDR PLL supports the full range of data rates.

The CDR PLL requires offset cancellation to correct the analog offset voltages that may exist from process variations between the positive and negative differential signals in the CDR circuitry.

The CDR PLL operates either in LTR mode or LTD mode. After power-up or reset of the receiver PMA, the CDR PLL must first operate in LTR mode to keep the VCO output frequency close to the optimum recovered clock rate.

In LTR mode, the phase detector is not active. When the CDR PLL locks to the input reference clock, you can switch the CDR PLL to LTD mode to recover the clock and data from the incoming serial data.

In LTD mode, the PFD output is not valid and may cause the lock detect status indicator to toggle randomly. When there is no transition on the incoming serial data for an extended duration, you must switch the CDR PLL to LTR mode to wait for the real serial data.

The time needed for the CDR PLL to lock to data depends on the transition density and jitter of the incoming serial data and the parts per million (ppm) difference between the receiver input reference clock and the upstream transmitter reference clock. The receiver PCS must be held in reset until the CDR PLL locks to data and produces a stable recovered clock.

The LTR/LTD controller directs the CDR PLL transition between the LTR and LTD modes. The controller supports operation in both automatic lock mode and manual lock mode.

Related Information

For a detailed description of the offset cancellation process, see Dynamic Reconfiguration in Arria V Devices.

CDR PLL in Automatic Lock Mode

In automatic lock mode, the LTR/LTD controller directs the transition between the LTR and LTD modes when a set of conditions are met to ensure proper CDR PLL operation. The mode transitions are indicated by the `rx_is_lockedtodata` signal. In Arria V GZ devices, the mode transitions are indicated by the `pma_rx_is_lockedtodata` signal.

After power-up or reset of the receiver PMA, the CDR PLL is directed into LTR mode. The controller transitions the CDR PLL from LTR to LTD mode when all the following conditions are met:

- The frequency of the CDR PLL output clock and input reference clock is within the configured ppm frequency threshold setting.
- The phase of the CDR PLL output clock and input reference clock is within approximately 0.08 unit interval (UI) of difference.
- In PCIe configurations only—the signal detect circuitry must also detect the presence of the signal level at the receiver input above the threshold voltage specified in the PCI Express Base Specification 2.0 and PCI Express Base Specification 3.0 (Arria V GZ only).

The controller transitions the CDR PLL from LTD to LTR mode when either of the following conditions is met:

- The difference in between frequency of the CDR PLL output clock and input reference clock exceeds the configured ppm frequency threshold setting.
- In PCIe configurations only—the signal detect circuitry detects the signal level at the receiver input below the threshold voltage specified in the PCI Express Base Specification 2.0 and PCI Express Base Specification 3.0 (Arria V GZ only).
- In Arria V GZ, after switching to LTD mode, the `rx_is_lockedtodata` status signal is asserted. Lock to data takes a minimum of 4 µs, however the actual lock time depends on the transition density of the incoming data and the parts per million (ppm) difference between the receiver input reference clock and the upstream transmitter reference clock. The receiver PCS logic must be held in reset until the CDR produces a stable recovered clock.

If there is no transition on the incoming serial data for an extended duration, the CDR output clock may drift to a frequency exceeding the configured ppm threshold when compared with the input reference clock. In such a case, the LTR/LTD controller transitions the CDR PLL from LTD to LTR mode.

CDR PLL in Manual Lock Mode

In manual lock mode, the LTR/LTD controller directs the transition between the LTR and LTD modes based on user-controlled settings in the `pma_rx_set_locktodata` and `pma_rx_set_locktoref` registers. Manual lock mode provides the flexibility to manually control the CDR PLL mode transitions bypassing the ppm detection as required by certain applications that include, but not limited to, the following:

- Link with frequency differences between the upstream transmitter and the local receiver clocks exceeding the CDR PLL ppm threshold detection capability. For example, a system with asynchronous spread-spectrum clocking (SSC) downspread of –0.5% where the SSC modulation results in a ppm difference of up to 5000.
- Link that requires a faster CDR PLL transition to LTD mode, avoiding the duration incurred by the ppm detection in automatic lock mode.

In manual lock mode, your design must include a mechanism—similar to a ppm detector—that ensures the CDR PLL output clock is kept close to the optimum recovered clock rate before recovering the clock and data. Otherwise, the CDR PLL might not achieve locking to data. If the CDR PLL output clock frequency is detected as not close to the optimum recovered clock rate in LTD mode, direct the CDR PLL to LTR mode.

Related Information

For information about the proper sequence after power-up reset, see Transceiver Reset Control and Power-Down in Arria V Devices.

Deserializer

The deserializer provides serial-to-parallel data conversion and assumes the data is received LSB first from the receiver buffer. Additionally, the deserializer provides the clock-slip feature.

Clock-Slip

Word alignment in the PCS may contribute up to one parallel clock cycle of latency uncertainty. The clock-slip feature allows word alignment operation with a reduced latency uncertainty by performing the word alignment function in the deserializer. Use the clock slip feature for applications that require deterministic latency.

The deterministic latency state machine in the word aligner from the PCS automatically controls the clock-slip operation. After completing the clock-slip process, the deserialized data is word-aligned into the receiver PCS.

Transmitter PLL

In Arria V GX/GT/SX/ST devices, there are two transmitter PLL sources: CMU PLL and fPLL. In Arria V GZ devices, there are three transmitter PLL sources: ATX PLL, CMU PLL, and fPLL.

Table 1-14: Transmitter PLL Capability and Availability

Transmitter PLL	Serial Data Range	Availability
ATX PLL [4]	0.600 Gbps to 12.5 Gbps	Two transceivers per bank.
CMU PLL	0.611 Gbps to 10.3125 Gbps	Every channel when not used as receiver CDR (only two per transceiver bank capable to drive other channels)
fPLL	0.611 Gbps to 3.125 Gbps	Two per transceiver bank

Auxiliary Transmit (ATX) PLL Architecture

Arria V GZ devices contain two ATX PLLs per transceiver bank that can generate the high-speed clocks for the transmitter channels.

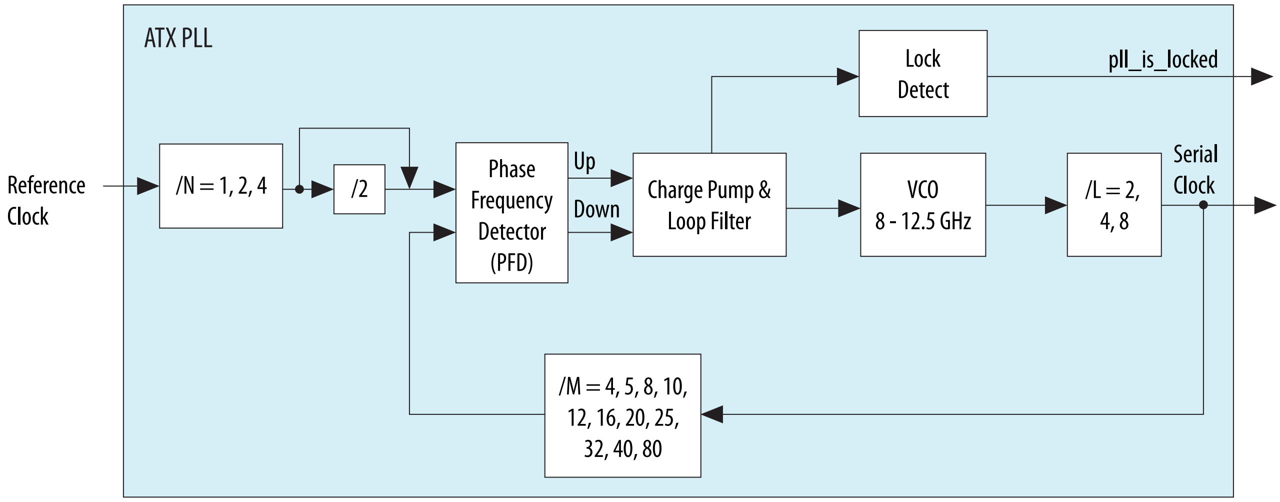
Compared with CMU PLLs, ATX PLLs have lower jitter and do not consume a transceiver channel; however an ATX PLL's frequency range is more limited.

The serial clock from the ATX PLL is routed to the transmitter clock dividers and can be further divided down to half the data rate of the individual channels. For best performance you should use the reference clock input that resides in the same transceiver block as your channel. However, you can use any dedicated reference clocks along the same side of the device to clock the ATX PLL.

[4] ATX PLL only available in Arria V GZ devices.

Note: Altera recommends that all Arria V GZ devices use the ATX PLL for channels operating between 8 to 12.5 Gbps data rates and to use the dedicated reference clock input residing in the same transceiver bank for the selected ATX PLL for best performance

Figure 1-19: ATX PLL Architecture



Related Information

For ATX PLL specifications such as input clock frequency or supported output data ranges, refer to the Arria V Device Datasheet.

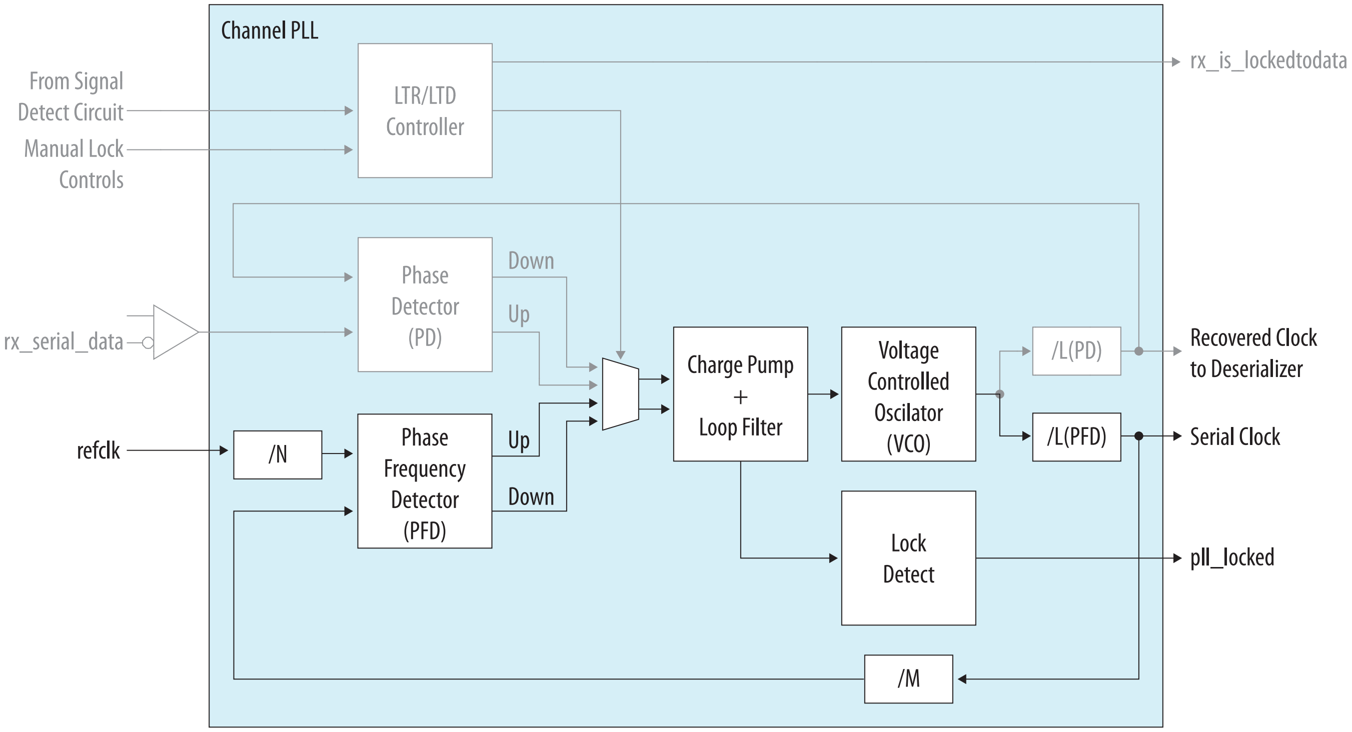
CMU PLL

In Arria V devices, if you do not use the channel PLL as a CDR, you can independently configure every channel PLL as a CMU PLL for clocking the transceivers.

Note: CDR functionality for the receiver is not available when you configure the channel PLL as a CMU PLL—you can use the transceiver channel only as a transmitter.

The CMU PLL operates only in lock-to-reference (LTR) mode and supports the full range of data rates.

Figure 1-20: CMU PLL in Arria V Devices



Using the input reference clock, the CMU PLL synthesizes the serial clock with a frequency that is half of the data rate. The CMU PLL output serial clock feeds the clock divider that resides in the transmitter of the same transceiver channel. Depending on the channel location in a transceiver bank, the CMU PLL of channels 1 and 4 feeds the output clock to the x1 clock lines.

Note: Transmitter PLLs within the upper-half or lower-half of a transceiver bank must be connected to the same Reconfiguration Controller.

Related Information

- **Receiver PMA Datapath** on page 1-16
- **For more information about the input reference clock and transmit PLL, see Transceiver Clocking in Arria V Devices.**

fPLL as Transmitter PLL

In addition to CMU PLL, the fPLL located adjacent to the transceiver banks are available for clocking the transmitters for serial data rates up to 3.125 Gbps.

Related Information

Clock Networks and PLLs in Arria V Devices

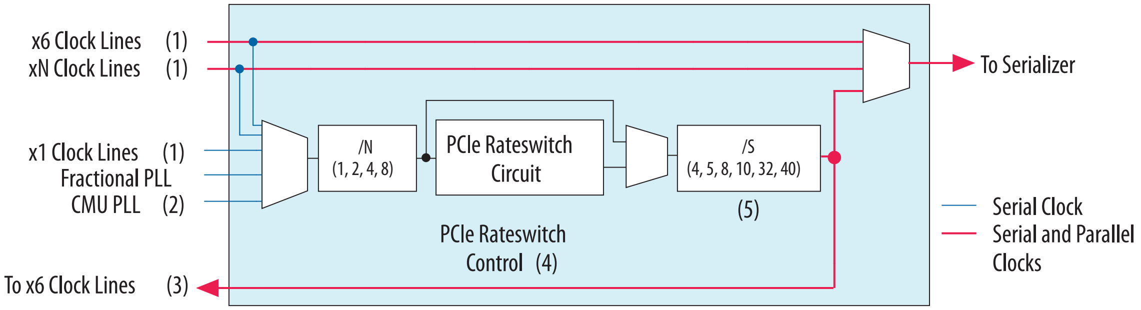
Clock Divider

Each Arria V transmitter channel has a clock divider.

There are two types of clock dividers, depending on the channel location in a transceiver bank:

- Local clock divider—channels 0, 2, 3, and 5 provide serial and parallel clocks to the PMA
- Central clock divider—channels 1 and 4 can drive the x6 and xN clock lines

Figure 1-21: Clock Divider for a Transceiver Channel in Arria V Devices



Notes:
1. For information about the x1, x6, and xN clock lines, see the Related Information.
2. Only from the channel PLL in the same transceiver channel configured as a CMU PLL.
3. Applicable for central clock dividers only (clock dividers in channels 1 and 4).
4. The PCIe rateswitch circuit allows dynamic switching between Gen2 and Gen1 line rates in a PCIe Gen2 design.
5. The divider settings are configured automatically depending on the serialization factor. The selected divider setting is half of the serialization factor. The 32 and 40 divider settings are only available for 10-Gbps channels.

Both types of clock dividers can divide the serial clock input to provide the parallel and serial clocks for the serializer in the channel if you use clocks from the clock lines or transmit PLLs. The central clock divider can additionally drive the x6 clock lines used to bond multiple channels.

In bonded channel configurations, both types of clock dividers can feed the serializer with the parallel and serial clocks directly, without dividing them from the x6 or xN clock lines.

Related Information

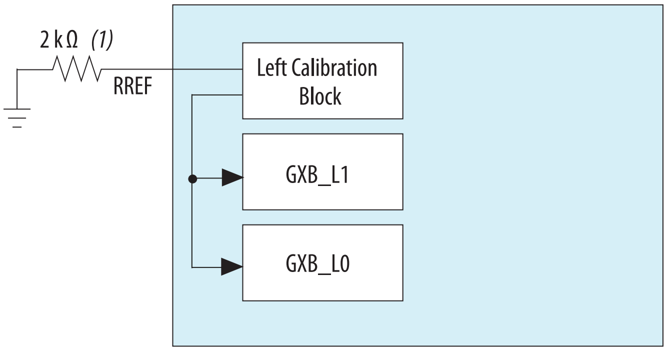
Transceiver Clocking in Arria V Devices

Calibration Block

The calibration block calibrates the differential OCT resistance and analog circuitry in the transceiver PMA to ensure the functionality is independent of PVT. It is also used for duty cycle calibration of the clock line at serial data rates above 4.9152 Gbps.

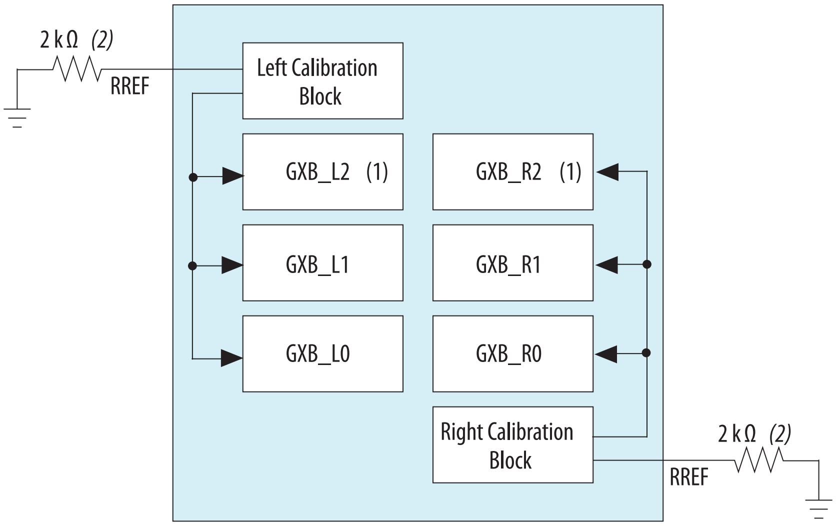
Up to two calibration blocks are available for the Arria V transceiver PMA.

Figure 1-22: Calibration Block Location and Connections in Arria V Devices with Transceivers on the Left Side of the Device Only



Note:
1. In Arria V GZ devices, you must use a 1.8 kΩ (maximum tolerance ± 1%) external resistor.

Figure 1-23: Calibration Block Location and Connections in Arria V Devices with Transceivers on Both Sides of the Device



Notes:
1. GXB_L2 and GXB_R2 banks are only available in some device variants.
2. In Arria V GZ devices, you must use a 1.8 kΩ (maximum tolerance ± 1%) external resistor.

The calibration block generates a constant internal reference voltage that is independent of PVT variations using the external reference resistor. The resulting reference currents are used to calibrate the transceiver banks.

Note: You must connect a separate 2 kΩ (tolerance maximum of ±1%) external resistor on each RREF pin to ground, except for Arria GZ devices. In Arria V GZ devices, you must use a 1.8 kΩ (maximum tolerance +/- 1%) external resistor. To ensure the calibration block operates properly, the RREF resistor connection in the board must be free from external noise.

Offset Cancellation in the Receiver Buffer and Receiver CDR

Process variation in smaller process silicon can lead to a V_{CM} offset between the p and n signals within the differential buffers. Arria V GZ devices have an automatic calibration in their receiver buffers to remove this V_{CM} offset.

You must use the reconfiguration controller IP for offset cancellation to take place. Calibration does not occur during transceiver reset, only during device configuration. Any signals that may appear on the receiver pin do not affect calibration because the receiver buffers are disconnected during calibration.

Note: A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

ATX PLL Calibration for Arria V GZ Devices

ATX PLL calibration optimizes the ATX PLL VCO settings for the desired output frequency. The reconfiguration controller IP must be instantiated for this calibration to run. The calibration occurs one time after device initialization.

Calibration Block Boundary

There is one calibration block in each quadrant of the device.

The calibration block also uses the reconfiguration controller clock (`mgmt_clk_clk`). This puts a restriction on the number of different reconfiguration clock sources that can be used in the design. All the transceiver channels controlled by a single calibration block must be connected to the same reconfiguration clock source.

Note: You can connect multiple reconfiguration controllers to the same clock source.

Table 1-15: Transceiver Calibration Block Boundary for Arria V GZ Devices

Arria V GZ Device	Package	Total Number of Transceiver channels in device	Total Number of Transceiver Channels per Side	Number of Contiguous Transceiver Channels Controlled by the Top Calibration Block (counting from top to bottom)	Number of Contiguous Transceiver Channels Controlled by the Bottom Calibration Block (counting from bottom to top)
5AGZC3	EH29	12	6 (Left) / 6 (Right)	3	3
	HF35	24	12 (Left) / 12 (Right)	6	6
	KF40	36	18 (Left) / 18 (Right)	9	9

Arria V GZ Device	Package	Total Number of Transceiver channels in device	Total Number of Transceiver Channels per Side	Number of Contiguous Transceiver Channels Controlled by the Top Calibration Block (counting from top to bottom)	Number of Contiguous Transceiver Channels Controlled by the Bottom Calibration Block (counting from bottom to top)
5AGZI3	EH29	12	6 (Left) / 6 (Right)	3	3
	HF35	24	12 (Left) / 12 (Right)	6	6
	KF40	36	18 (Left) / 18 (Right)	9	9
5AGZC4	EH29	12	6 (Left) / 6 (Right)	3	3
	HF35	24	12 (Left) / 12 (Right)	6	6
	KF40	36	18 (Left) / 18 (Right)	9	9
5AGZI4	EH29	12	6 (Left) / 6 (Right)	6	6
	HF35	24	12 (Left) / 12 (Right)	12	12
	KF40	36	18 (Left) / 18 (Right)	9	9

Related Information

Refer to the Transceiver Reconfiguration Controller PMA Analog Control Registers section of the Altera Transceiver PHY IP Core User Guide

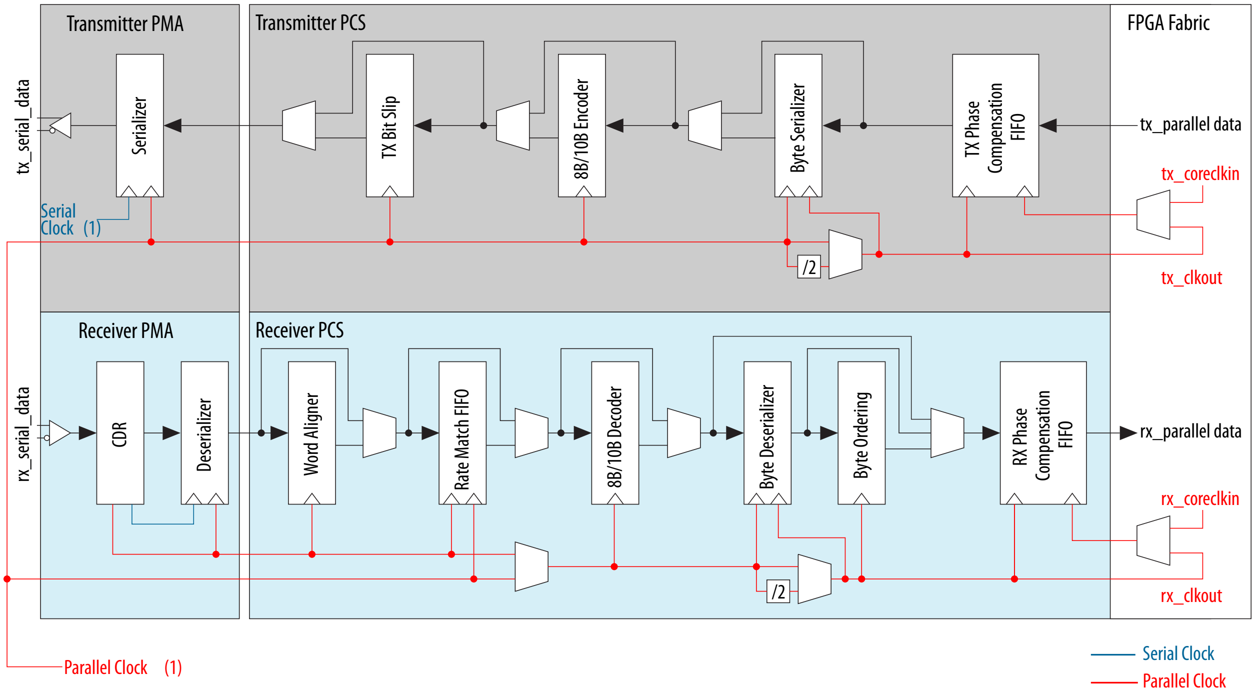
PCS Architecture

The PCS architecture of Arria V GX/GT/SX/ST devices is slightly different from the GZ devices.

The GZ has three types of PCS blocks: standard PCS block, a 10G PCS block, and a PCIe Gen3 PCS block.

The GX, SX, GT, and ST devices have only one type of PCS block, which is similar to the GZ standard PCS block, except for the data rate support. The GX and GT PCS supports up to 6.5536 Gbps while the GZ PCS supports up to 9.8 Gbps. The 10G PCS of the GZ supports 12.5 Gbps, and the PCIe Gen3 PCS supports the PCIe Gen3 Base specification.

Figure 1-24: Transceiver Channel PCS in Arria V Devices



Note:
1. The serial and parallel clocks are sourced from the clock divider.

Note: For Arria V GT and ST devices, the PCS is not available when using the 10-Gbps channels; only the PMA is available. You must implement the PCS functions required for the interface using user logic in the FPGA fabric with an 80-bit FPGA fabric-transceiver width.

Arria V GZ transceiver PCS blocks fully support data rates up to 12.5 Gbps You have the option to bypass the PCS using the PMA direct mode.

Table 1-16: PCS Datapath Configurations

Parameter	Single-Width	Double-Width
PMA–PCS Interface Width	8 or 10 bit	16 or 20 bit
FPGA Fabric–Transceiver Interface Width	8 or 10 bit 16 or 20 bit [5]	16 or 20 bit 32 or 40 bit [5]

[5] The byte serializer and deserializer are enabled.

Parameter	Single-Width	Double-Width
Supported configurations	PCIe Gen1, Gen2, and Gen3 XAUI Custom	Custom
Data rate range in a custom configuration	0.6 to 3.75 Gbps	1.0 to 11 Gbps

Transmitter PCS Datapath for Arria V GX, SX, GT, and ST Devices and Arria V GZ Standard PCS

This section describes the transmitter phase compensation FIFO, byte serializer, 8B/10B encoder, and transmitter bit-slip blocks in the transmitter PCS datapaths.

Table 1-17: Functional Blocks in the Transmitter PCS Datapath

Block	Functionality
Transmitter Phase Compensation FIFO	• Compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock when interfacing the transmitter PCS with the FPGA fabric directly or with the PCIe hard IP block • Supports operation in phase compensation and registered modes
Byte Serializer	• Divides the FPGA fabric–transceiver interface frequency in half at the transmitter channel by doubling the transmitter input datapath width • Allows the transmitter channel to operate at higher data rates with the FPGA fabric–transceiver interface frequency that is within the maximum limit • Supports operation in double-width modes
8B/10B Encoder	• Generates 10-bit code groups from 8-bit data and 1-bit control identifier, in compliance with Clause 36 of the IEEE 802.3 specification • Supports operation in single- and double-width modes, and running disparity control
Transmitter Bit-Slip	• Enables user-controlled, bit-level delay in the data prior to serialization for serial transmission • Supports operation in single- and double-width modes

Transmitter Phase Compensation FIFO

The transmitter phase compensation FIFO is four words deep and interfaces the control and data signals between the transmitter PCS and FPGA fabric or PCIe hard IP block.

The FIFO supports the following operations:

- Phase compensation mode with various clocking modes on the read clock and write clock
- Registered mode with only one clock cycle of datapath latency

Phase Compensation Mode

The transmitter phase compensation FIFO compensates any phase difference between the read and write clocks for the transmitter control and data signals.

The low-speed parallel clock feeds the read clock; the FPGA fabric interface clock feeds the write clock. The clocks must have 0 ppm difference in frequency or a FIFO underrun or overflow condition may result.

The transmitter phase compensation FIFO supports various clocking modes on the read and write clocks depending on the transceiver configuration.

Related Information

For a detailed description of transmitter datapath interface clocking modes when using the transmitter phase compensation FIFO, see Transceiver Clocking in Arria V Devices.

Registered Mode

To eliminate the FIFO latency uncertainty for applications with stringent datapath latency uncertainty requirements, bypass the FIFO functionality in registered mode to incur only one clock cycle of datapath latency when interfacing the transmitter channel to the FPGA fabric. Configure the FIFO to registered mode when interfacing the transmitter channel to the PCIe hard IP block to reduce datapath latency. In registered mode, the low-speed parallel clock that is used in the transmitter PCS clocks the FIFO.

Byte Serializer

The byte serializer allows the transmitter channel to operate at higher data rates in a configuration that exceeds the FPGA fabric–transceiver interface frequency limit.

The byte serializer supports operation in single- and double-width modes. The datapath clock rate at the output of the byte serializer is twice the FPGA fabric–transmitter interface clock frequency. The byte serializer forwards the least significant word first followed by the most significant word.

Note: You must use the byte serializer in configurations that exceed the maximum frequency limit of the FPGA fabric–transceiver interface.

Table 1-18: Transmitter Input Datapath Conversion

Mode	Transmitter Input Datapath Width	Byte Serializer Output Datapath Width	Byte Serializer Output Ordering
Single Width	16	8	Least significant 8 bits of the 16-bit input first
	20	10	Least significant 10 bits of the 20-bit input first

Mode	Transmitter Input Datapath Width	Byte Serializer Output Datapath Width	Byte Serializer Output Ordering
Double Width	32	16	Least significant 16 bits of the 32-bit input first
	40	20	Least significant 20 bits of the 40-bit input first

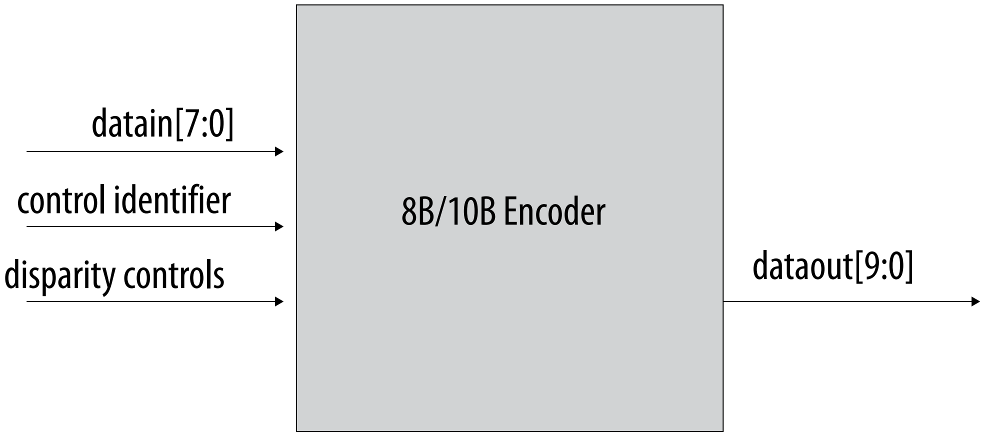
8B/10B Encoder

The 8B/10B encoder supports operation in single- and double-width modes with the running disparity control feature.

8B/10B Encoder in Single-Width Mode

In single-width mode, the 8B/10B encoder generates 10-bit code groups from 8-bit data and 1-bit control identifier with proper disparity according to the PCS reference diagram in Clause 36 of the IEEE 802.3 specification. The 10-bit code groups are generated as valid data code-groups (/Dx.y/) or special control code-groups (/Kx.y/), depending on the 1-bit control identifier.

Figure 1-25: 8B/10B Encoder Diagram in Single-Width Mode



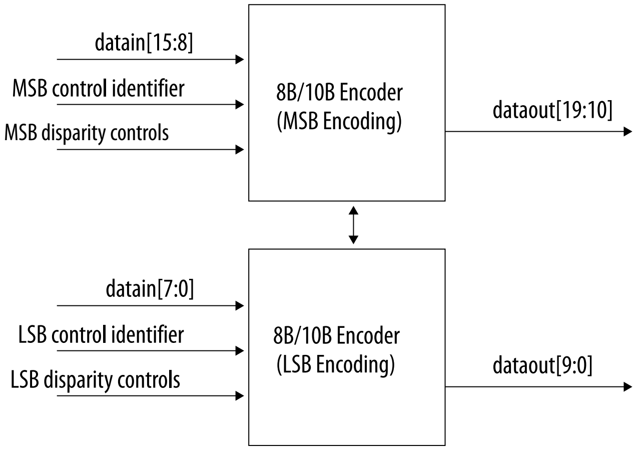
The IEEE 802.3 specification identifies only 12 sets of 8-bit characters as /Kx.y/. If other sets of 8-bit characters are set to encode as special control code-groups, the 8B/10B encoder may encode the output 10-bit code as an invalid code (it does not map to a valid /Dx.y/ or /Kx.y/ code), or unintended valid / Dx.y/ code, depending on the value entered.

8B/10B Encoder in Double-Width Mode

In double-width mode, two 8B/10B encoders are cascaded to generate two sets of 10-bit code groups from 16-bit data and two 1-bit control identifiers.

When receiving the 16-bit data, the 8-bit LSByte is encoded first, followed by the 8-bit MSByte.

Figure 1-26: 8B/10B Encoder Diagram in Double-Width Mode



Running Disparity Control

The 8B/10B encoder automatically performs calculations that meet the running disparity rules when generating the 10-bit code groups.

The running disparity control feature provides user-controlled signals (`tx_dispval` and `tx_forcedisp`) to manually force encoding into a positive or negative current running disparity code group. When enabled, the control overwrites the current running disparity value in the encoder based on user-controlled signals, regardless of the internally-computed current running disparity in that cycle.

Note: Using the running disparity control may temporarily cause a running disparity error at the receiver.

Encoder Output During Reset Sequence

Figure 1-27: 8B/10B Encoder Output During and After Reset Conditions

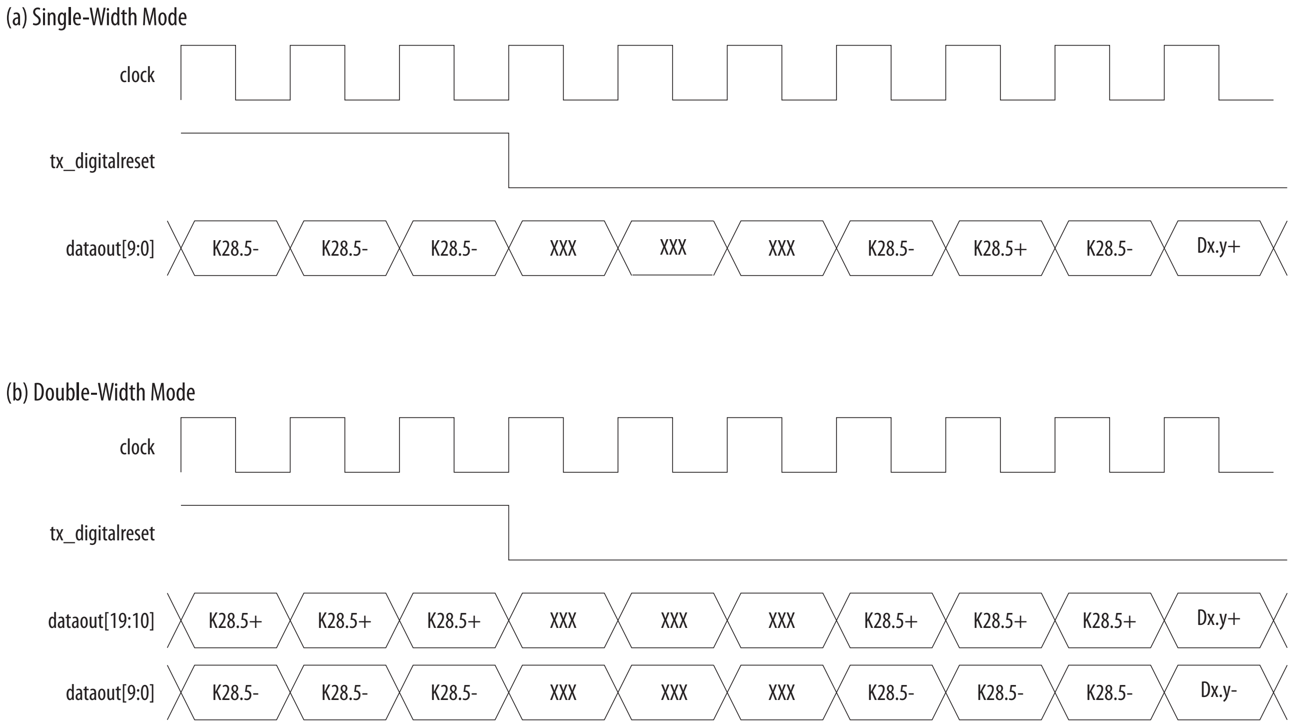


Table 1-19: 8B/10B Encoder Output During and After Reset Conditions

Operation Mode	During 8B/10B Reset	After 8B/10B Reset Release
Single Width	Continuously sends the /K28.5/ code from the RD– column	Some “don’t cares” due to pipelining in the transmitter channel, followed by three /K28.5/ codes with proper disparity—starts with negative disparity—before sending encoded 8-bit data at its input.
Double Width	Continuously sends the /K28.5/ code from the RD– column on LSByte and the /K28.5/ code from the RD+ column on MSByte	Some “don't cares” due to pipelining in the transmitter channel, followed by: • Three /K28.5/ codes from the RD– column before sending encoded 8-bit data at its input on LSByte. • Three /K28.5/ codes from the RD+ column before sending encoded 8-bit data at its input on MSByte.

Transmitter Bit-Slip

The transmitter bit-slip enables a bit-level delay insertion to the data prior to serialization for the serial transmission. The transmitter bit-slip supports operation in single- and double-width modes. Each bit slipped at the transmitter incurs one serial bit of datapath latency.

Table 1-20: Bits Slip Allowed with the `tx_bitslipboundaryselect` Signal

Operation Mode	Maximum Bit-Slip Setting
Singe width (8- or 10-bit)	9
Double width (16- or 20-bit)	19

Receiver PCS Datapath for Arria V GX, SX, GT, and ST Devices and Arria V GZ Standard PCS

Table 1-21: Functional Blocks in the Arria V GX/GT/SX/ST 6-Gbps Receiver PCS and Arria V GZ Standard PCS Datapaths

Block	Functionality
Word Aligner	• Searches for a predefined alignment pattern in the deserialized data to identify the correct boundary and restores the word boundary during link synchronization • Supports an alignment pattern length of 7, 8, 10, 16, 20, or 32 bits • Supports operation in four modes—manual alignment, bit-slip, automatic synchronization state machine, and deterministic latency state machine—in single- and double-width configurations • Supports the optional programmable run-length violation detection, polarity inversion, bit reversal, and byte reversal features
Rate Match FIFO	• Compensates for small clock frequency differences of up to ±300 ppm—600 ppm total—between the upstream transmitter and the local receiver clocks by inserting or deleting skip symbols when necessary • Supports operation that is compliant to the clock rate compensation function in supported protocols
8B/10B Decoder	• Receives 10-bit data and decodes the data into an 8-bit data and a 1-bit control identifier—in compliance with Clause 36 of the IEEE 802.3 specification • Supports operation in single- and double-width modes
Byte Deserializer	• Divides the FPGA fabric–transceiver interface frequency in half at the receiver channel by doubling the receiver output datapath width • Allows the receiver channel to operate at higher data rates with the FPGA fabric–transceiver interface frequency that is within the maximum limit • Supports operation in double-width modes

Block	Functionality
Byte Ordering	• Searches for a predefined pattern that must be ordered to the LSByte position in the parallel data going to the FPGA fabric when you enable the byte deserializer
Receiver Phase Compensation FIFO	• Compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock when interfacing the receiver PCS with the FPGA fabric directly or with the PCIe hard IP block • Supports operation in phase compensation and registered modes

Word Aligner

The Word Aligner provides word boundary restoration during link synchronization.

Parallel data at the input of the receiver PCS loses the word boundary of the upstream transmitter from the serial-to-parallel conversion in the deserializer. The word aligner provides word boundary restoration during link synchronization with the following four modes:

- Manual alignment mode
- Bit-slip mode
- Automatic synchronization state machine mode
- Deterministic latency state machine mode

The word aligner searches for a predefined alignment pattern in the deserialized data to identify the correct boundary and restores the word boundary during link synchronization. The alignment pattern is predefined for standard serial protocols according to the respective protocol specifications to achieve synchronization or you can specify the settings with a custom word alignment pattern for proprietary protocol implementations. Except for bit-slip mode, after completing word alignment, the deserialized data is synchronized to have the word alignment pattern at the LSB portion of the aligned data.

In addition to restoring the word boundary, the word aligner also supports optional features.

Table 1-22: Optional Word Aligner Features

Feature	Availability
Programmable Run-Length Violation Detection	All transceiver configurations
Receiver Polarity Inversion	All transceiver configurations except PCIe
Receiver Bit Reversal	Custom single- and double-width configurations only
Receiver Byte Reversal	Custom double-width configuration only

The operation mode and alignment pattern length support varies depending on the word aligner configurations.

Table 1-23: Word Aligner Operation Mode and Pattern Length Support

PCS Mode	PMA–PCS Interface Width	Word Aligner Mode	Alignment Pattern Length
Single Width	8 bits	Manual alignment	8 bits or 16 bits
		Bit-slip	–
	10 bits	Manual alignment	7 or 10 bits
		Bit-slip	–
		Automatic synchronization state machine	7 or 10 bits [6]
		Deterministic latency state machine	10 bits [7]
Double Width	16 bits	Manual alignment	8, 16, or 32 bits
		Bit-slip	–
	20 bits	Manual alignment	7, 10, 20, or 40 bits
		Bit-slip	–
		Deterministic latency state machine	10 bits [7]

When the 8B/10B encoder/decoder is enabled, the word aligner detects both positive and negative disparities of the alignment pattern. For example, if you specify a /K28.5/ (b'0011111010) pattern as the comma, `rx_patterndetect` is asserted if b'0011111010 or b'1100000101 is detected in the incoming data.

Word Aligner in Manual Alignment Mode

In manual alignment mode, the word alignment operation is manually controlled with the `rx_std_wa_patternalign` input signal or the `rx_enapatternalign` register.

Depending on the configuration, controlling the `rx_std_wa_patternalign` signal enables the word aligner to look for the predefined word alignment pattern in the received data stream. A value 1 at the `rx_patterndetect` register indicates that the word alignment pattern is detected . A value 1 at the `rx_syncstatus` register indicates that the word aligner has successfully synchronized to the new word boundary.

Manual word alignment can be also triggered by writing a value 1 to `rx_enapatternalign` register. The word alignment is triggered in the next parallel clock cycle when a 0 to 1 transition occurs on the `rx_enapatternalign` register.

[6] For PCIe implementation, the word aligner is configured using the automatic synchronization state machine with alignment pattern length of 10 bits.

[7] For more information about CPRI in deterministic latency state machine, refer to the *CPRI Enhancements* section of the **Transceiver Protocol Configurations in Arria V Devices** chapter.

After rx_syncstatus is asserted and if the incoming data is corrupted causing an invalid code group, rx_syncstatus remains asserted. The rx_errdetect register will be set to 1 (indicating RX 8B/10B error detected). When this happens, the manual alignment mode is not be able to de-assert the rx_syncstatus signal. You must manually assert rx_digitalreset or manually control rx_std_wa_patternalign to resynchronize a new word boundary search whenever rx_errdetect shows an error.

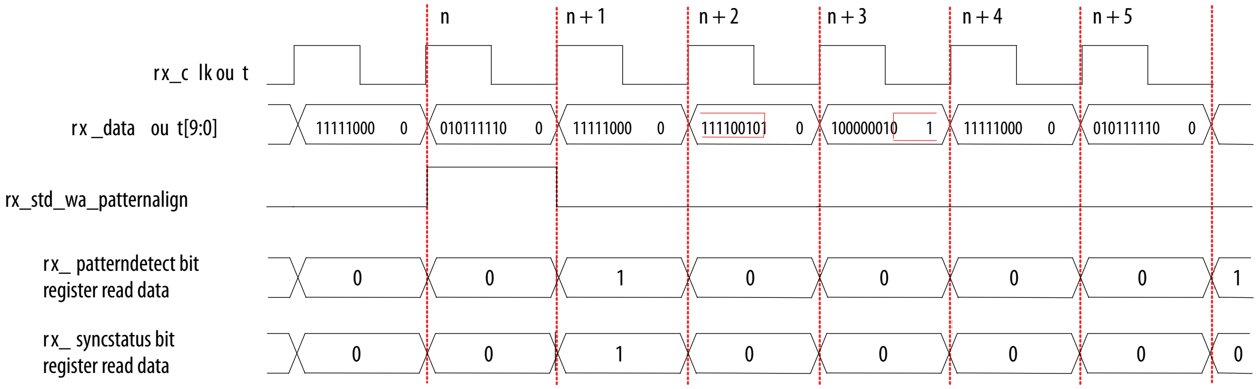
Table 1-24: Word Aligner Operations in Manual Alignment Mode

PCS Mode	PMA–PCS Interface Width	Word Alignment Operation
Single Width	8 bits or 10 bits	**1.** After the rx_digitalreset signal deasserts, assert rx_std_wa_patternalign signal for one parallel clock cycle for the word aligner to look for the predefined word alignment pattern in the received data stream and synchronize to the new word boundary. **2.** Any alignment pattern found thereafter in a different word boundary does not cause the word aligner to resynchronize to this new word boundary if the rx_std_wa_patternalign signal is deasserted. **3.** To resynchronize to a new word boundary, assert the rx_std_wa_patternalign signal for another parallel clock cycle. **4.** If you keep the rx_std_wa_patternalign signal asserted continuously, before the signal rx_digitalreset is deasserted, the word aligner updates the word boundary when the first alignment pattern is found, even though rx_std_wa_patternalign signal was not asserted for one parallel clock cycle to trigger the word alignment operation. **5.** If you keep the rx_std_wa_patternalign signal deasserted, the word aligner maintains the current word boundary even when it finds the alignment pattern in a new word boundary. **6.** When the word aligner synchronizes to the new word boundary, the rx_syncstatus register will have a value 1 for one parallel clock cycle. The rx_patterndetect register will have a value 1 whenever a word alignment pattern is found for one parallel clock cycle regardless of whether the word aligner is triggered to align to the new word boundary or not.

PCS Mode	PMA–PCS Interface Width	Word Alignment Operation
Double Width	16 bits or 20 bits	**1.** After the `rx_digitalreset` signal deasserts, regardless of whether `rx_std_wa_patternalign` status signal is asserted or deasserted, the word aligner synchronizes to the first predefined alignment pattern found. **2.** Any alignment pattern found thereafter in a different word boundary does not cause the word aligner to resynchronize to this new word boundary. **3.** To resynchronize to the new word boundary, assert the `rx_std_wa_patternalign` signal for one parallel clock cycle. **4.** When the word aligner synchronizes to the new word boundary, the `rx_syncstatus` register will have a value 1 till the `rx_digitalreset` signal is deasserted or `rx_std_wa_patternalign` signal is asserted again. The `rx_patterndetect` register will have a value 1 whenever a word alignment pattern is found for one parallel clock cycle regardless of whether the word aligner is triggered to align to the new word boundary or not.

The configuration selected for this example is word aligner operation in single width with 10-bit PMA-PCS interface mode. In this example, a /K28.5/ (10'b0101111100) is specified as the word alignment pattern. The word aligner aligns to the /K28.5/ alignment pattern in the nth cycle because the `rx_std_wa_patternalign` signal is asserted during the nth cycle. The `rx_syncstatus` register has a value 1 in the next parallel clock cycle, indicating alignment to a new word boundary. The `rx_patterndetect` signal also goes high for one clock cycle indicating word alignment pattern detection. At time n + 1, the `rx_std_wa_patternalign` signal is deasserted to instruct the word aligner to lock the current word boundary. The alignment pattern is detected again in a new word boundary across cycles n + 2 and n + 3. The word aligner does not align to this new word boundary because the `rx_std_wa_patternalign` signal is set to 0. The /K28.5/ word alignment pattern is detected again in the current word boundary during cycle n + 5, causing the `rx_patterndetect` to have a value 1 in the next parallel clock cycle.

Figure 1-28: Word Aligner in Manual Alignment Mode



Note: If the word alignment pattern is known to be unique and does not appear between word boundaries, you can continuously assert `rx_std_wa_patternalign` signal or set the `rx_enapatternalign` register to 1 at all times because there is no possibility of false word alignment. If there is a possibility of the word alignment pattern occurring across word boundaries, you must control the number of clock cycles that the `rx_std_wa_patternalign` is asserted or the number of clock cycles the `rx_enapatternalign` register is set to 1 to avoid re-alignment to an incorrect word boundary.

Bit-Slip Mode

In bit-slip mode, the word alignment is achieved by manually controlling the data slip with the `rx_std_bitslip` signal.

Slipping the received data by one bit effectively shifts the word boundary by one bit. You can implement a controller in the FPGA fabric to iteratively control the `rx_std_bitslip` signal until the word aligner output matches the predefined word alignment pattern to achieve synchronization.

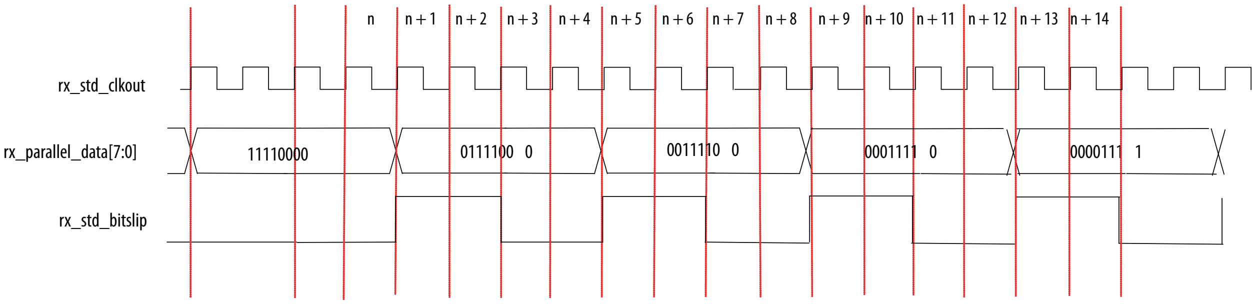
Table 1-25: Word Aligner in Bit-Slip Mode

PCS Mode	PMA–PCS Interface Width (bits)	Word Alignment Operation
Single Width	8	**1.** At every rising edge to the `rx_std_bitslip` signal, the word aligner slips one bit into the received data. **2.** When bit-slipping shifts a complete round of the data bus width, the word boundary is back to the original boundary. **3.** Check the received data, `rx_parallel_data` after every bit slip operation whether the predefined word alignment pattern is visible in the new word boundary. When the word alignment pattern is visible in the new word boundary, the alignment process is completed and the bit-slip operation can be stopped.
	10	
Double Width	16	
	20	

Note: For every bit slipped in the word aligner, the earliest bit received is lost.

For this example, consider that 8'b11110000 is received back-to-back and 16'b0000111100011110 is the predefined word alignment pattern. A rising edge on the `rx_std_bitslip` signal at time n + 1 slips a single bit 0 at the MSB position, forcing the `rx_parallel_data` to 8'b01111000. Another rising edge on the `rx_std_bitslip` signal at time n + 5 forces `rx_parallel_data` to 8'b00111100. Another rising edge on the `rx_std_bitslip` signal at time n + 9 forces `rx_parallel_data` to 8'b00011110. Another rising edge on the `rx_std_bitslip` signal at time n + 13 forces the `rx_parallel_data` to 8'b00001111. At this instance, `rx_parallel_data` in cycles n + 12 and n + 13 is 8'b00011110 and 8'b00001111, respectively, which matches the specified 16-bit alignment pattern 16'b0000111100011110.

Figure 1-29: Word Aligner Configured in Bit Slip Mode



Note: Bit slip operation can also be triggered by a 0 to 1 transition in the `rx_bitslip` register.

Word Aligner in Automatic Synchronization State Machine Mode

In automatic synchronization state machine mode, a programmable state machine determines the moment that the word aligner has either achieved synchronization or lost synchronization.

You can configure the state machine to provide hysteresis control during link synchronization and throughout normal link operation. Depending on your protocol configurations, the state machine parameters are automatically configured so they are compliant with the synchronization state machine in the respective protocol specification.

Table 1-26: State Machine Parameters for the Word Aligner in Automatic Synchronization State Machine Mode

Parameter	Values
Number of valid synchronization code groups or ordered sets received to achieve synchronization	1–256
Number of erroneous code groups received to lose synchronization	1–64
Number of continuous good code groups received to reduce the error count by one	1–256

Table 1-27: Word Aligner Operation in Automatic Synchronization State Machine Mode

PCS Mode	PMA–PCS Interface Width	Word Alignment Operation
Single Width	10 bits	**1.** After the `rx_digitalreset` signal deasserts, the word aligner starts looking for the predefined word alignment pattern, or its complement, in the received data stream and automatically aligns to the new word boundary. **2.** Synchronization is achieved only after the word aligner receives the programmed number of valid synchronization code groups in the same word boundary and is indicated by the value 1 in `rx_syncstatus` register. The `rx_syncstatus` register contains a value 0 if the word aligner has lost synchronization. **3.** Loss of synchronization occurs when the word aligner receives the programmed number of erroneous code groups without receiving the intermediate good code groups and is indicated by the value 0 in the `rx_syncstatus` register. **4.** The word aligner may achieve synchronization again after receiving a new programmed number of valid synchronization code groups in the same word boundary.

Word Aligner in Deterministic Latency State Machine Mode

In deterministic latency state machine mode, word alignment is achieved by performing a clock-slip in the deserializer until the deserialized data coming into the receiver PCS is word-aligned.

The state machine controls the clock-slip process in the deserializer after the word aligner has found the alignment pattern and identified the word boundary. Deterministic latency state machine mode offers a reduced latency uncertainty in the word alignment operation for applications that require deterministic latency.

After `rx_syncstatus` is asserted and if the incoming data is corrupted causing an invalid code group, `rx_syncstatus` remains asserted. The `rx_errdetect` register will be set to 1 (indicating RX 8B/10B error detected). When this happens, the manual alignment mode is not be able to de-assert the `rx_syncstatus` signal. You must manually assert `rx_digitalreset` or manually control `rx_std_wa_patternalign` to resynchronize a new word boundary search whenever `rx_errdetect` shows an error.

PCS Mode	PMA–PCS Interface Width	Word Alignment Operation
Single Width	10 bits	1. After `rx_digitalreset` deasserts, the word aligner starts looking for the predefined word alignment pattern, or its complement, in the received data stream and automatically aligns to the new word boundary. 2. After the pattern is found and the word boundary is identified, the state machine controls the clock-slip process in the deserializer. 3. When the clock-slip is complete, the deserialized data coming into the receiver PCS is word-aligned and is indicated by the value 1 in the `rx_syncstatus` register until `rx_digitalreset` is deasserted. 4. To resynchronize to the new word boundary, the Avalon-MM register `rx_enapatternalign` (not available as a signal) must be reasserted to initiate another pattern alignment. Asserting `rx_enapatternalign` may cause the extra shifting in the RX datapath if `rx_enablepatternalign` is asserted while bit slipping is in progress. Consequently, `rx_enapatternalign` should only be asserted under the following conditions: - `rx_syncstatus` is asserted - `rx_bitslipboundaryselectout` changes from a non-zero value to zero or 1 5. When the word aligner synchronizes to the new word boundary, `rx_syncstatus` has a value of 1 until `rx_digitalreset` is deasserted or `rx_enapatternalign` is set to 1. `rx_patterndetect` has a value of 1 whenever a word alignment pattern is found for one parallel clock cycle regardless of whether or not the word aligner is triggered to align to the new word boundary.
Double Width	20 bits	

Programmable Run-Length Violation Detection

The programmable run-length violation detection circuit detects if the number of consecutive 1s or 0s in the received data exceeds the user-specified threshold.

Table 1-28: Detection Capabilities of the Run-Length Violation Circuit

PCS Mode	PMA–PCS Interface Width (bits)	Run-Length Violation Detector Range	
		Minimum	Maximum
Single Width	8	4	128
	10	5	160
Double Width	16	8	512
	20	10	640

Receiver Polarity Inversion

The positive and negative signals of a serial differential link might erroneously be swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic can be expensive. The polarity inversion feature at the receiver corrects the swapped signal error without requiring board re-spin or major updates to the logic in the FPGA fabric. The polarity inversion feature inverts the polarity of every bit at the input to the word aligner, which has the same effect as swapping the positive and negative signals of the serial differential link.

Inversion is controlled dynamically with the `rx_invpolarity` register. When you enable the polarity inversion feature, initial disparity errors may occur at the receiver with the 8B/10B-coded data. The receiver must be able to tolerate these disparity errors.

Caution: If you enable polarity inversion midway through a word, the word will be corrupted.

Bit Reversal

You can reverse the bit order at the output of the word aligner for receiving a MSB-to-LSB transmission using the bit reversal feature at the receiver.

Table 1-29: Bit Reversal Feature

	Transmission Bit Order	
Bit Reversal Option	**8- or 10-bit Serialization Factor**	**16- or 20-bit Serialization Factor**
Disabled (default)	LSB to MSB	LSB to MSB
Enabled	MSB to LSB For example: 8-bit—`D[7:0]` rewired to `D[0:7]` 10-bit—`D[9:0]` rewired to `D[0:9]`	MSB to LSB For example: 16-bit—`D[15:0]` rewired to `D[0:15]` 20-bit—`D[19:0]` rewired to `D[0:19]`

Note: When receiving the MSB-to-LSB transmission, the word aligner receives the data in reverse order. The word alignment pattern must be reversed accordingly to match the MSB-first incoming data ordering.

You can dynamically control the bit reversal feature using the `rx_bitreversal_enable` register with the word aligner in bit-slip mode. When you dynamically enable the bit reversal feature in bit-slip mode, you can ignore the pattern detection function in the word aligner because the word alignment pattern cannot be dynamically reversed to match the MSB-first incoming data ordering.

Receiver Byte Reversal

The two symbols of incoming data at the receiver in double-width mode may be accidentally swapped during transmission.

For a 16-bit input data width at the word aligner, the two symbols are `bits[15:8]` and `bits[7:0]`. For a 20-bit input data width at the word aligner, the two symbols are `bits[19:10]` and `bits[9:0]`. The byte reversal feature at the word aligner output can correct this error by swapping the two symbols in double-width mode at the word aligner output.

Table 1-30: Byte Reversal Feature

Byte Reversal Option	Word Aligner Output	
	16-bit Data Width	20-bit Data Width
Disabled	`D[15:0]`	`D[19:0]`
Enabled	`D[7:0], D[15:8]`	`D[9:0], D[19:10]`

The reversal is controlled dynamically using the `rx_bytereversal_enable` register. Enabling the byte reversal option may cause initial disparity errors at the receiver with 8B/10B-coded data. The receiver must be able to tolerate these disparity errors.

Note: When receiving swapped symbols, the word alignment pattern must be byte-reversed to match the incoming byte-reversed data.

Rate Match FIFO

The Rate Match FIFO compensates for the small clock frequency differences between the upstream transmitter and the local receiver clocks.

In a link where the upstream transmitter and local receiver can be clocked with independent reference clock sources, the data can be corrupted by any frequency differences (in ppm count) when crossing the data from the recovered clock domain—the same clock domain as the upstream transmitter reference clock—to the local receiver reference clock domain.

The rate match FIFO is 20 words deep, which compensates for the small clock frequency differences of up to ±300 ppm (600 ppm total) between the upstream transmitter and the local receiver clocks by performing symbol insertion or deletion, depending on the ppm difference on the clocks.

The rate match FIFO operation requires that the transceiver channel is in duplex configuration (both transmit and receive functions) and a predefined 20-bit pattern (consisting of a 10-bit control pattern and a 10-bit skip pattern). The 10-bit skip pattern must be chosen from a code group with neutral disparity.

The FIFO operates by looking for the 10-bit control pattern, followed by the 10-bit skip pattern in the data after the word aligner has restored the word boundary. After finding the pattern, the FIFO performs the following operations to ensure the FIFO does not underflow or overflow:

- Inserts the 10-bit skip pattern when the local receiver reference clock frequency is greater than the upstream transmitter reference clock frequency
- Deletes the 10-bit skip pattern when the local receiver reference clock frequency is less than the upstream transmitter reference clock frequency

The rate match FIFO supports operations in single- and double-width modes. You can define the 20-bit pattern for custom configurations. For protocol configurations, the FIFO is automatically configured to support a clock rate compensation function as required by the following specifications:

- The PCIe protocol per clock tolerance compensation requirement, as specified in the PCI Express Base Specification 2.0 for Gen1 and Gen2 signaling rates and PCI Express Base Specification 3.0 for Gen1, Gen2, and Gen3 signaling rates (Arria V GZ only).
- The Gbps Ethernet (GbE) protocol per clock rate compensation requirement using an idle ordered set, as specified in Clause 36 of the IEEE 802.3 specification

Note: For the Gigabit Ethernet protocol, if you enabled rate match FIFO in the autonegotiation state machine in an FPGA core, refer to the "Rate Match FIFO" section in the "Gigabit Ethernet" section in the *Transceiver Protocol Configurations in Arria V Devices* chapter.

Related Information

- **Transceiver Custom Configurations in Arria V Devices**
  For more information about the rate match FIFO operation in custom-specific configurations.
- **Transceiver Protocol Configurations in Arria V Devices**
  For more information about the rate match FIFO operation in protocol-specific configurations.

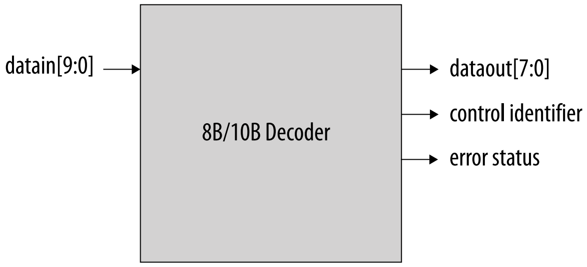
8B/10B Decoder

The 8B/10B decoder supports operation in single- and double-width modes.

8B/10B Decoder in Single-Width Mode

In single-width mode, the 8B/10B decoder decodes the received 10-bit code groups into an 8-bit data and a 1-bit control identifier in compliance with Clause 36 in the IEEE 802.3 specification. The 1-bit control identifier indicates if the decoded 8-bit code is a valid data or special control code.

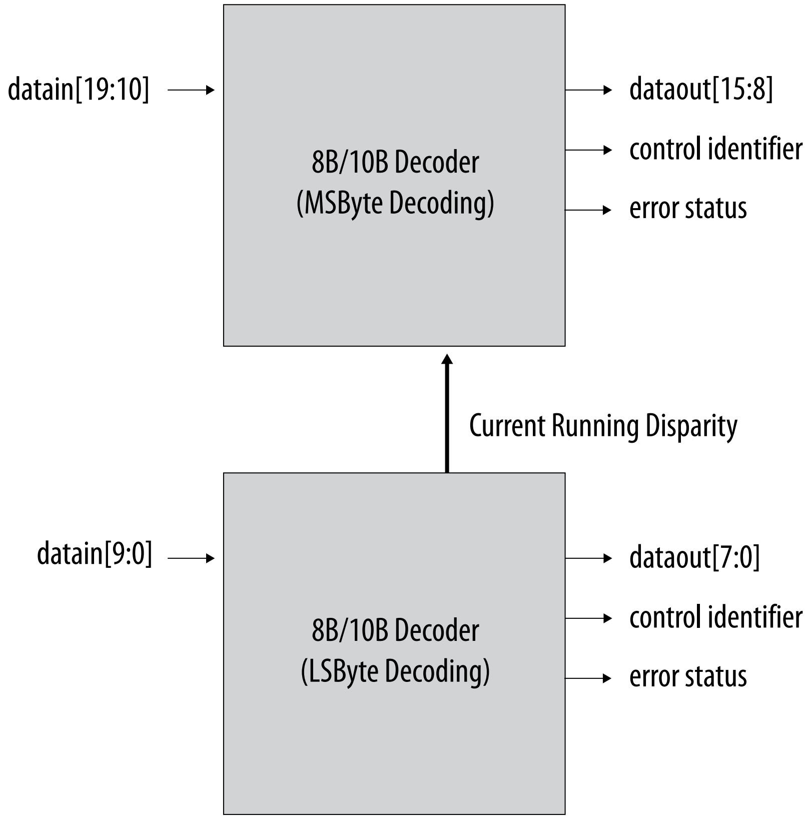
Figure 1-30: 8B/10B Decoder Block Diagram in Single-Width Mode



8B/10B Decoder in Double-Width Mode

In double-width mode, two 8B/10B decoders are cascaded to decode the 20-bit code groups into two sets of 8-bit data and two 1-bit control identifiers. When receiving the 20-bit code group, the 10-bit LSByte is decoded first and the ending running disparity is forwarded to the other 8B/10B decoder for decoding the 10-bit MSByte.

Figure 1-31: 8B/10B Decoder Block Diagram in Double-Width Mode



Byte Deserializer

The byte deserializer allows the receiver channel to operate at higher data rates in a configuration that exceeds the FPGA fabric–transceiver interface frequency limit.

The byte deserializer supports operation in single- and double-width modes. The datapath clock rate at the input of the byte deserializer is twice the FPGA fabric–receiver interface clock frequency.

Note: You must use the byte deserializer in configurations that exceed the maximum frequency limit of the FPGA fabric–transceiver interface.

After byte deserialization, the word alignment pattern may be ordered in the MSByte or LSByte position.

Table 1-31: Byte Deserializer Input Datapath Width Conversion

Data is assumed to be received as LSByte first—the least significant 8 or 10 bits in single-width mode or the least significant 16 or 20 bits in double-width mode.

Mode	Byte Deserializer Input Datapath Width	Receiver Output Datapath Width
Single Width	8	16
	10	20

Mode	Byte Deserializer Input Datapath Width	Receiver Output Datapath Width
Double Width	16	32
	20	40

Byte Ordering

When you enable the byte deserializer, the output byte order may not match the originally transmitted ordering. For applications that require a specific pattern to be ordered at the LSByte position of the data, byte ordering can restore the proper byte order of the byte-deserialized data before forwarding it to the FPGA fabric.

Byte ordering operates by inserting a predefined pad pattern to the byte-deserialized data if the predefined byte ordering pattern found is not in the LSByte position.

Byte ordering requires the following:

- A receiver with the byte deserializer enabled
- A predefined byte ordering pattern that must be ordered at the LSByte position of the data
- A predefined pad

Byte ordering supports operation in single- and double-width modes. Both modes support operation in word aligner-based and manual ordering modes.

Byte Ordering in Single-Width Mode

Byte ordering is supported only when you enable the byte deserializer.

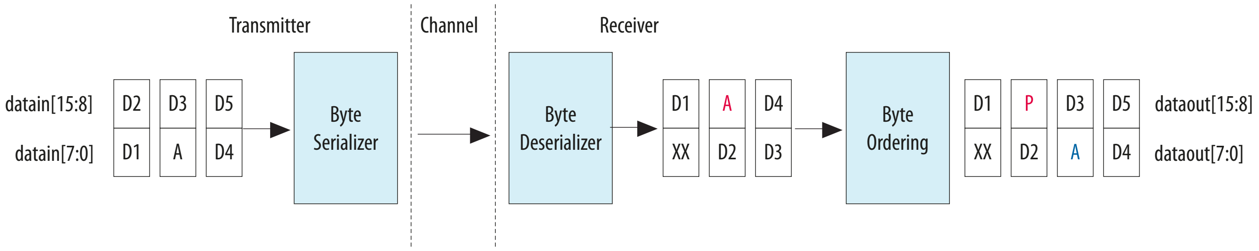
Table 1-32: Byte Ordering Operation in Single-Width Mode

PMA–PCS Interface Width	FPGA Fabric–Transceiver Interface Width	8B/10B Decoder	Byte Ordering Pattern Length	Pad Pattern Length
8 bits	16 bits	Disabled	8 bits	8 bits
10 bits	16 bits	Enabled	9 bits[8]	9 bits [8]
	20 bits	Disabled	10 bits	10 bits

[8] The MSB of the 9-bit pattern represents the 1-bit control identifier of the 8B/10B-decoded data. The lower 8 bits represent the 8-bit decoded code.

Figure 1-32: Byte Ordering Operation Example in Single-Width Mode

An example of a byte ordering operation in single-width mode (8-bit PMA-PCS interface width) where A is the predefined byte ordering pattern and P is the predefined pad pattern.



Byte Ordering in Double-Width Mode

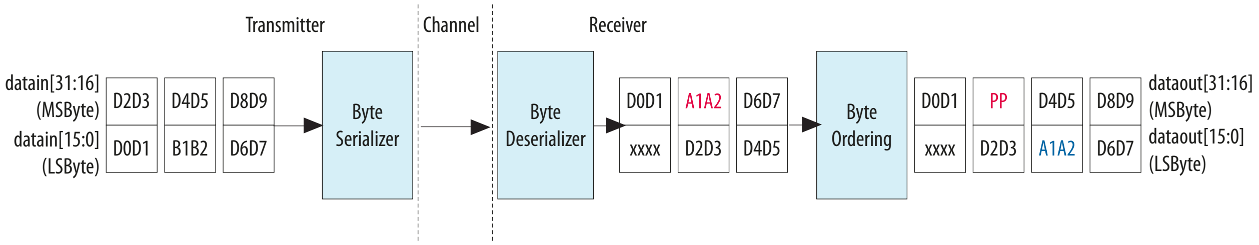
Byte ordering is supported only when you enable the byte deserializer.

Table 1-33: Byte Ordering Operation in Double-Width Mode

PMA–PCS Interface Width	FPGA Fabric–Transceiver Interface Width	8B/10B Decoder	Byte Ordering Pattern Length	Pad Pattern Length
16 bits	32 bits	Disabled	8 or 16 bits	8 bits
20 bits	32 bits	Enabled	9[9] or 18 bits[10]	9 bits[9]
	40 bits	Disabled	10 or 20 bits	10 bits

Figure 1-33: Byte Ordering Operation Example in Double-Width Mode

An example of a byte ordering operation in double-width mode (16-bit PMA-PCS interface width) where A1A2 is the predefined byte ordering pattern and P is the predefined pad pattern.



Word Aligner-Based Ordering Mode

In word aligner-based ordering mode, the byte ordering operation is controlled by the word aligner synchronization status signal, `rx_syncstatus`.

[9] The MSB of the 9-bit pattern represents the 1-bit control identifier of the 8B/10B-decoded data. The lower 8 bits represent the 8-bit decoded code.

[10] The 18-bit pattern consists of two sets of 9-bit patterns, individually represented as in the previous note.

After a rising edge on the rx_syncstatus signal, byte ordering looks for the byte ordering pattern in the byte-deserialized data.

When the first data byte that matches the byte ordering pattern is found, the byte ordering performs the following operations:

- If the pattern is not in the LSByte position—byte ordering inserts the appropriate number of pad patterns to push the byte ordering pattern to the LSByte position and indicates the byte alignment.
- If the pattern is in the LSByte position—byte ordering indicates the byte alignment.

Any byte misalignment found thereafter is ignored unless another rising edge on the rx_syncstatus signal, indicating resynchronization, is observed.

Manual Ordering Mode

In manual ordering mode, the byte ordering operation is controlled using the rx_enabyteord signal.

A rising edge on the rx_enabyteord signal triggers byte ordering to look for the byte ordering pattern in the byte-deserialized data.

When the first data byte that matches the byte ordering pattern is found, the byte ordering performs the following operations:

- If the pattern is not in the LSByte position—byte ordering inserts the appropriate number of pad patterns to push the byte ordering pattern to the LSByte position and indicates the byte alignment.
- If the pattern is in the LSByte position—byte ordering indicates the byte alignment.

Any byte misalignment found thereafter is ignored unless another rising edge on the rx_enabyteord signal is observed.

Receiver Phase Compensation FIFO

The receiver phase compensation FIFO is four words deep and interfaces the status and data signals between the receiver PCS and the FPGA fabric or the PCIe hard IP block. The FIFO supports the following operations:

- Phase compensation mode with various clocking modes on the read clock and write clock
- Registered mode with only one clock cycle of datapath latency

Phase Compensation Mode

The receiver phase compensation FIFO compensates for any phase difference between the read and write clocks for the receiver status and data signals.

The low-speed parallel clock feeds the write clock; the FPGA fabric interface clock feeds the read clock. The clocks must have 0 ppm difference in frequency or a FIFO underrun or overflow condition may result.

The receiver phase compensation FIFO supports various clocking modes on the read and write clocks depending on the transceiver configuration.

Related Information

For a detailed description of the receiver datapath interface clocking modes when using the receiver phase compensation FIFO, seeTransceiver Clocking in Arria V Devices.

Registered Mode

To eliminate the FIFO latency uncertainty for applications with stringent datapath latency uncertainty requirements, bypass the FIFO functionality in registered mode to incur only one clock cycle of datapath latency when interfacing the receiver channel to the FPGA fabric. Configure the FIFO to registered mode when interfacing the receiver channel to the PCIe hard IP block to reduce datapath latency. In registered mode, the low-speed parallel clock that is used in the receiver PCS clocks the FIFO.

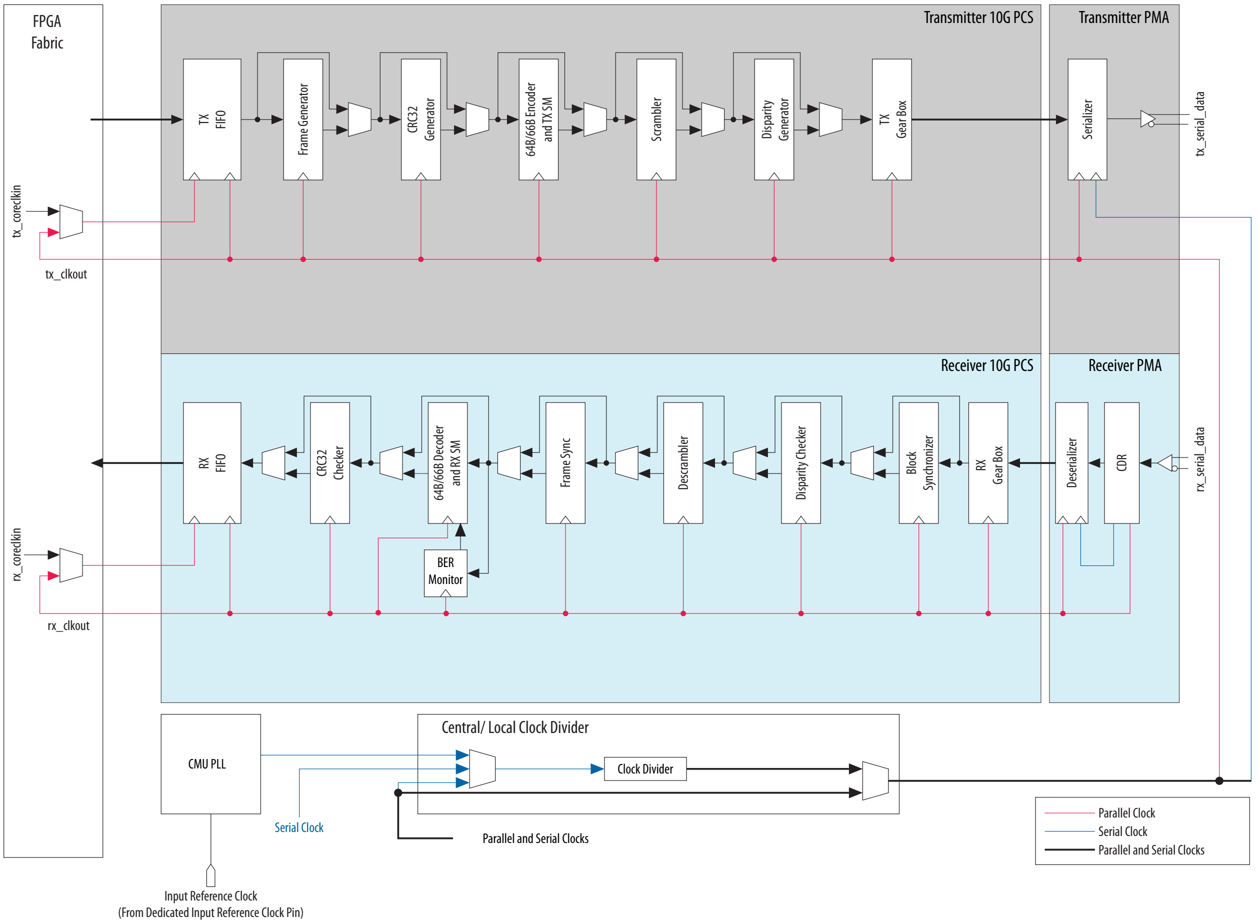
10G PCS Architecture for Arria V GZ Devices

The 10G PCS architecture offers a full duplex (transmitter and receiver) transceiver channel that supports serial data rates up to 12.5 Gbps for Arria V GZ devices.

Several functional blocks are customized for various protocols. The different datapath configurations for these protocols are available through the different PHY IPs instantiated through the Parameter Editor. Currently, the only supported configurations for the 10G PCS are Interlaken, 10GBASE-R and low-latency 10G PCS.

Figure 1-34: 10G PCS Datapath in Arria V GZ Channels

Not all the blocks shown in the 10G PCS datapath are available in every configuration.



Related Information

Altera Transceiver PHY IP Core User Guide

Transmitter 10G PCS Datapath

The sub-blocks in the transmitter 10G PCS datapath are described in order from the transmitter FIFO to the transmitter gearbox.

Transmitter FIFO

The transmitter FIFO provides an interface between the transmitter channel PCS and the FPGA fabric.

In 10GBASE-R configurations, the transmitter FIFO receives data from the FPGA fabric. The data output from the transmitter FIFO block goes to the 64B/66B encoder.

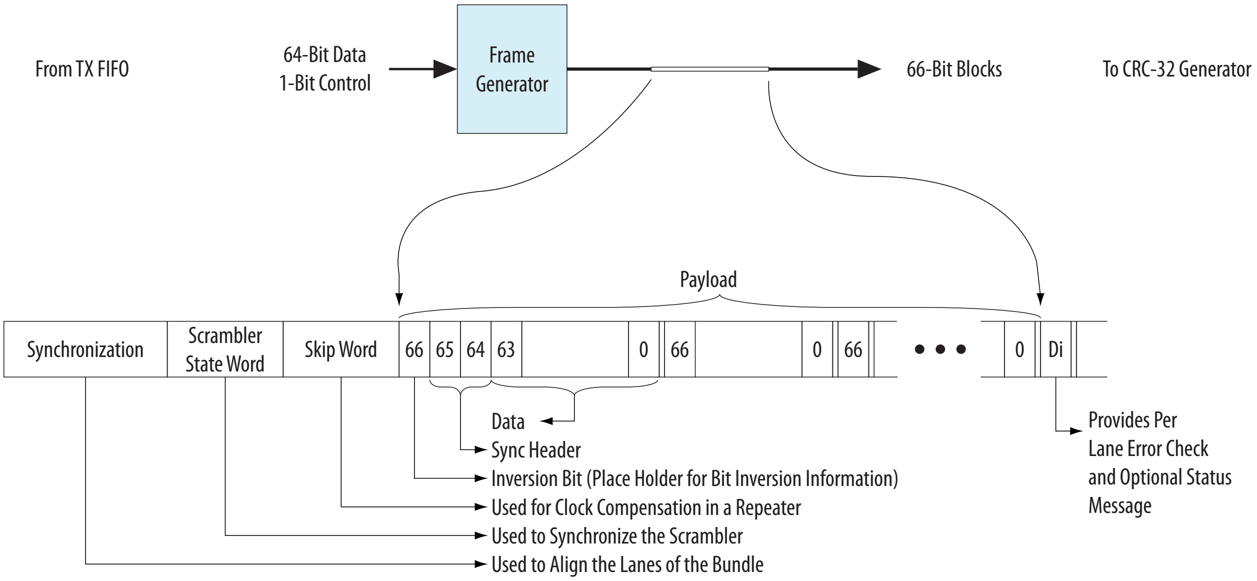
In Interlaken configurations, the transmitter FIFO sends a control signal to indicate whether it is ready to receive data from the FPGA fabric. The user logic sends the data to the transmitter FIFO only if this control signal is asserted. In this configuration, data output from the transmitter FIFO block goes to the frame generator.

Frame Generator

The frame generator block supports the Interlaken protocol. The frame generator block takes the data from the transmitter FIFO and encapsulates the payload and burst/idle control words from the FPGA fabric with the framing layer's control words, such as the synchronization word, scrambler state word, skip word, and diagnostic word, to form a metaframe. The Interlaken PHY IP core Parameter Editor allows you to set the metaframe length.

Note: The frame generator is used only in Interlaken configurations.

Figure 1-35: Frame Generator


From TX FIFO
64-Bit Data
1-Bit Control
Frame Generator
66-Bit Blocks
To CRC-32 Generator
Payload
Synchronization
Scrambler State Word
Skip Word
66
65
64
63
0
66
0
66
• • •
0
Di
Data
Sync Header
Inversion Bit (Place Holder for Bit Inversion Information)
Used for Clock Compensation in a Repeater
Used to Synchronize the Scrambler
Used to Align the Lanes of the Bundle
Provides Per Lane Error Check and Optional Status Message

CRC-32 Generator

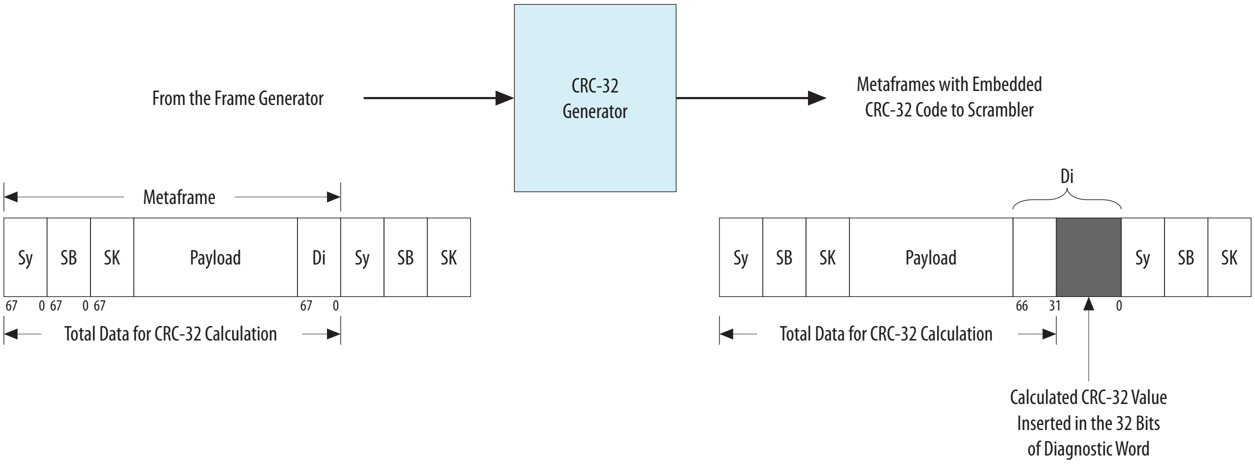
The CRC-32 generator block receives data from the frame generator and calculates the cyclic redundancy check (CRC) code for each block of data. This CRC code value is stored in the CRC32 field of the diagnostic word.

Note: The CRC-32 generator is used only in Interlaken configurations.

The CRC-32 calculation covers most of the metaframe, including the diagnostic word, except the following:

- bits `[66:64]` of each word
- 58-bit scrambler state within the scrambler state word
- 32-bit CRC-32 field within the diagnostic word

Figure 1-36: CRC-32 Generator


From the Frame Generator
CRC-32 Generator
Metaframes with Embedded CRC-32 Code to Scrambler
Metaframe
Sy
SB
SK
Payload
Di
Sy
SB
SK
67
0
67
0
67
67
0
Total Data for CRC-32 Calculation
Di
Sy
SB
SK
Payload
Sy
SB
SK
66
31
0
Total Data for CRC-32 Calculation
Calculated CRC-32 Value Inserted in the 32 Bits of Diagnostic Word

64B/66B Encoder

The 64B/66B encoder conforms to the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

Note: The 64B/66B encoder is used only in 10GBASE-R configurations.

This block contains the 64B/66B encoder sub-block and the transmitter state machine sub-block. The 64B/66B encoder sub-block receives data from the transmitter FIFO and encodes the 64-bit data and 8-bit control characters to the 66-bit data block required by the 10GBASE-R configuration. The transmit state machine in the 64B/66B encoder sub-block checks the validity of the 64-bit data from the MAC layer and ensures proper block sequencing.

Scrambler

The scrambler operates in frame synchronous mode and self synchronous mode. Frame synchronous mode operates in Interlaken configurations. Self synchronous mode operates in 10GBASE-R configurations, as specified in IEEE 802.3-2008 clause-49.

Disparity Generator

The disparity generator block conforms to the Interlaken protocol specification and provides a DC-balanced data output. The disparity generator receives data from the scrambler and inverts the running disparity to stay within the ±96-bit boundary. To ensure this running disparity requirement, the disparity generator inverts bits [63:0] and sets bit [66] to indicate the inversion.

Note: The disparity generator is used only in Interlaken configurations.

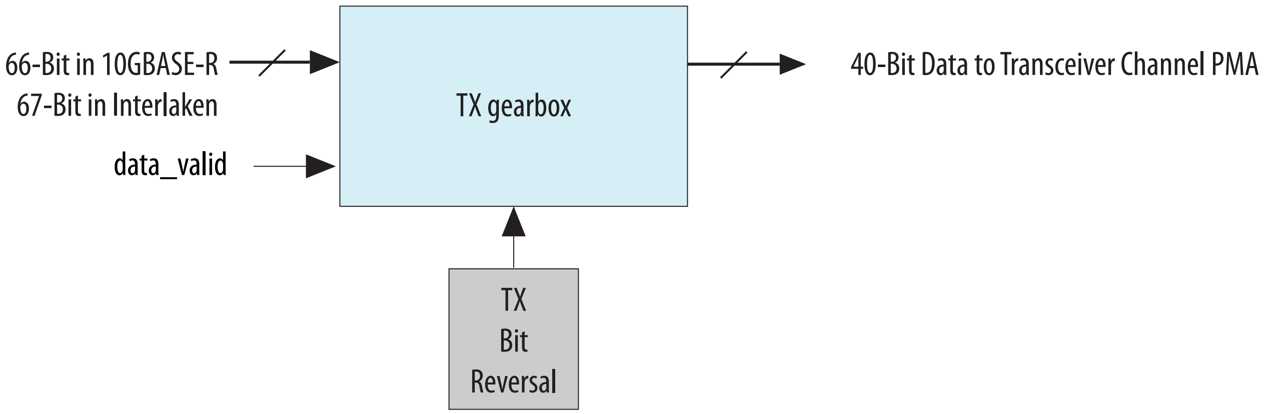
Table 1-34: Interpretation of the MSB in the 67-Bit Payload for Arria V Devices

MSB	Interpretation
0	Bits [63:0] are not inverted; the disparity generator processes the word without modification
1	Bits [63:0] are inverted; the disparity generator inverts the word before processing it

Transmitter Gearbox

The transmitter gearbox adapts the PCS data width to a smaller bus width for interfacing with the PMA. Because of the transmitter gearbox, the difference in the bus widths between the PCS and the PMA is transparent to the logic in the FPGA fabric.

Figure 1-37: Transmitter Gearbox



In addition to providing bus width adaptation, the transmitter gearbox provides the transmitter polarity inversion, bit reversal and bit-slip features.

Transmitter Bit Reversal

The transmitter gearbox can reverse the order of transmitted bits. By default, the transmitter first sends out the LSB of a word. Some protocols, such as Interlaken, require that the MSB of a word (bit 66 in a word [66:0]) is transmitted first. When you enable the transmitter bit reversal, the parallel input to the gearbox is swapped and the MSB is sent out first. The Quartus® II software automatically sets the bit reversal for Interlaken configurations.

Transmitter Polarity Inversion

Transmitter polarity can be used to reverse the positive and negative differential buffer signals. This is useful if these signals are reversed on the board or backplane layout.

A high value on the tx_invpolarity register, which is accessed via the Avalon-MM PHY management interface, inverts the polarity of every bit of the input data word to the serializer in the transmitter datapath. Because inverting the polarity of each bit has the same effect as swapping the positive and negative signals of the differential link, correct data is sent to the receiver. Dynamically changing the tx_invpolarity register value might cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

If polarity inversion is asserted midway through a serializer word, the word will be corrupted.

Transmitter Bit-Slip

The transmitter bit-slip allows you to compensate for the channel-to-channel skew between multiple transmitter channels by slipping the data sent to the PMA. The maximum number of bits slipped is controlled from the FPGA fabric and is equal to the width of the PMA-PCS interface, minus one.

Receiver 10G PCS Datapath

The sub-blocks in the receiver 10G PCS datapath are described in order from the receiver gearbox to the receiver FIFO.

Receiver Gearbox

The receiver gearbox adapts the PMA data width to a larger bus width to interface with the PCS. The PMA bus width is smaller than the PCS bus width; therefore, the receiver gearbox expands the data bus width from the PMA to the PCS. Because bus width adaptation is transparent, you can continuously feed data to the receiver gearbox. In addition to providing bus width adaptation, the receiver gearbox provides the receiver bit reversal feature.

Receiver Polarity Inversion

The receiver gearbox can invert the polarity of the incoming data. This is useful if the receive signals are reversed on the board or backplane layout.

Receiver Bit Reversal

The receiver gearbox allows bit reversal of the received data. Some protocols, such as Interlaken, require the bit reversal feature.

Related Information

Transmitter Bit Reversal on page 1-58

Block Synchronizer

The block synchronizer determines the block boundary of a 66-bit word in the case of the 10GBASE-R protocol or a 67-bit word in the case of the Interlaken protocol. The incoming data stream is slipped one bit at a time until a valid synchronization header (bits 65 and 66) is detected in the received data stream. After the predefined number of synchronization headers (as required by the protocol specification) is detected, the block synchronizer asserts the status signal to other receiver PCS blocks down the receiver datapath and to the FPGA fabric.

The block synchronizer is designed in accordance with both the Interlaken protocol specification and the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

Disparity Checker

The design of the disparity checker is based on the Interlaken protocol specifications. After word synchronization is achieved, the disparity checker monitors the status of the 67th bit of the incoming word and determines whether or not to invert bits [63:0] of the received word.

Note: The disparity checker is only used in Interlaken configurations.

Table 1-35: Interpretation of the MSB in the 67-Bit Payload for Arria V Devices

MSB	Interpretation
0	Bits `[63:0]` are not inverted; the disparity checker processes the word without modification
1	Bits `[63:0]` are inverted; the disparity checker inverts the word to achieve the original word before processing it

Descrambler

The descrambler descrambles data per the protocol specifications supported by the 10G PCS. The descrambler operates either in frame synchronous or self synchronous mode.

Frame Synchronous Mode

Frame synchronous mode is used in Interlaken configurations only. When block synchronization is achieved, the descrambler uses the scrambler seed from the received scrambler state word. This block also forwards the current descrambler state to the frame synchronizer.

Self Synchronous Mode

Self synchronous mode is used in 10GBASE-R configurations only.

Frame Synchronizer

The frame synchronizer block achieves lock by looking for four synchronization words in consecutive metaframes. After synchronization, the frame synchronizer monitors the scrambler word in the metaframe and deasserts the lock signal after three consecutive mismatches and starts the synchronization process again. Lock status is available to the FPGA fabric.

Note: The frame synchronizer is only used in Interlaken configurations.

Bit-Error Rate (BER) Monitor

The BER monitor block conforms to the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49. After block lock is achieved, the BER monitor starts to count the number of invalid synchronization headers within a 125-ms period. If more than 16 invalid synchronization headers are observed in a 125-ms period, the BER monitor provides the status signal to the FPGA fabric, indicating a high bit error rate condition.

64B/66B Decoder

The 64B/66B decoder block contains a 64B/66B decoder sub-block and a receiver state machine sub-block. The 64B/66B decoder sub-block converts the received data from the descrambler into 64-bit data and 8-bit control characters. The receiver state machine sub-block monitors the status signal from the BER monitor.

If the status signal is asserted, the receiver state machine sends local fault ordered sets to the FPGA interface.

Note: The 64B/66B encoder is used only in 10GBASE-R configurations.

The 64B/66B decoder block is designed towards the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

CRC-32 Checker

The CRC-32 checker block supports the Interlaken protocol. The CRC-32 checker calculates the CRC from the incoming data and compares the result to the CRC value sent in the diagnostic word. The CRC error signal is sent to the FPGA fabric.

Receiver FIFO

The receiver FIFO block operates in different modes based on the transceiver datapath configuration.

The Quartus II software automatically selects receiver FIFO mode for the configuration you use.

Clock Compensation Mode

The receiver FIFO is configured in clock compensation mode for the 10GBASE-R configuration. In clock compensation mode, the FIFO deletes idles or ordered sets and inserts only idles to compensate up to a ±100 ppm clock difference between the remote transmitter and the local receiver.

Generic Mode

The receiver FIFO is configured in generic mode for the Interlaken configuration. In generic mode, the receiver FIFO provides the FIFO partially empty and FIFO full status signals to the FPGA fabric to control the read side of the FIFO.

Phase Compensation Mode

The receiver FIFO is configured in phase compensation mode for the 10G custom configuration. In phase compensation mode, the FIFO compensates for the phase difference between the FIFO write clock and the read clock.

XAUI Mode

In XAUI mode, the receiver FIFO compensates for up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The XAUI protocol requires the transmitter to send /R/ (/K28.0/) code groups simultaneously on all four lanes (denoted as ||R|| column) during inter-packet gaps (IPGs), conforming to the IEEE P802.3ae specification. The receiver FIFO operation in XAUI mode is compliant with the IEEE P802.3ae specification.

PCIe Mode

In PCIe mode, the receiver FIFO compensates for up to ±300 ppm (total 600 ppm) difference between the upstream transmitter and the local receiver. The PCIe protocol requires the transmitter to send SKP ordered sets during IPGs, conforming to the PCIe base specification 2.0. The SKP ordered set is defined as a /K28.5/ COM symbol followed by three consecutive /K28.0/ SKP symbol groups.

The PCIe protocol requires the receiver to recognize a SKP ordered set as a /K28.5/ COM symbol followed by one to five consecutive /K28.0/ SKP symbols. The rate match FIFO operation is compliant with PCIe Base Specification 2.0.

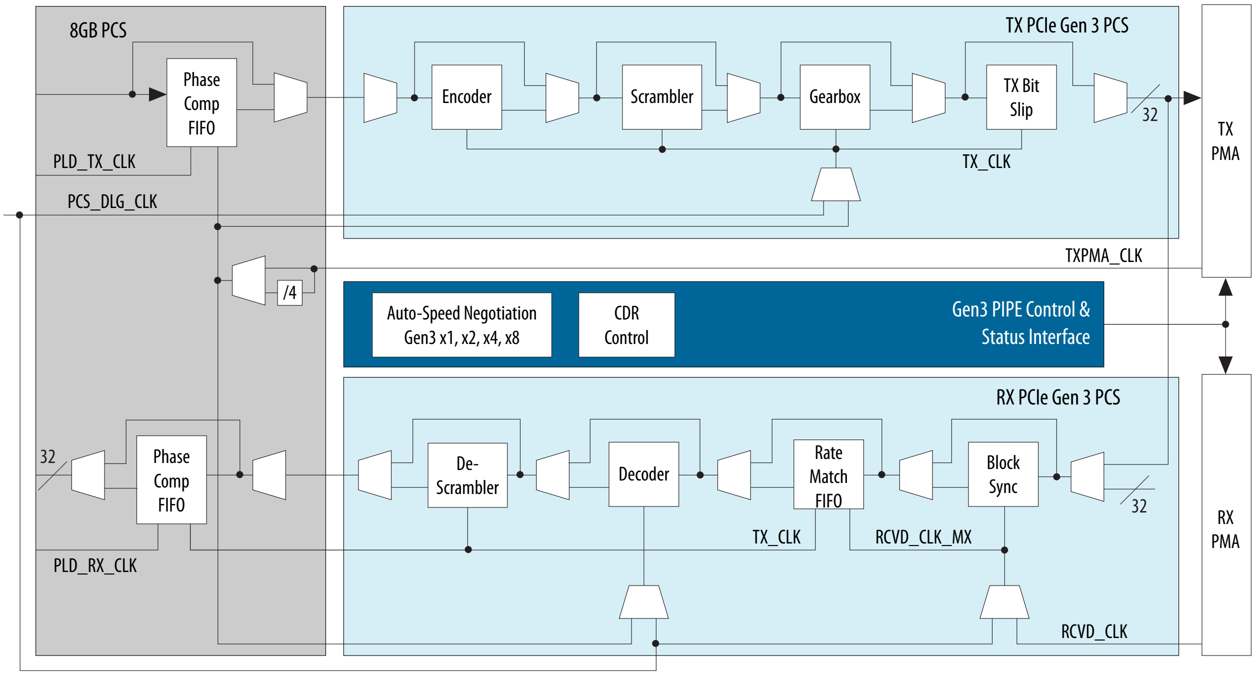
PCIe Gen3 PCS Architecture

Arria V supports the PCIe Gen3 Base specification. The PCIe Gen3 uses a 128/130 bit block encoding/decoding scheme which is different from the 8B/10B scheme used in Gen1 and Gen2. The 130-bit block contains a 2-bit sync header and 128-bit data payload. For this reason, the PCIe Gen3 PCS has a separate data path as compared to the PCIe Gen1 or Gen2 PCS. The PCIe Gen3 PCS supports the PHY Interface for the PCI Express (PIPE) interface with the hard IP enabled and with the hard IP bypassed.

This PIPE interface supports the seamless switching of Data and Clock between the Gen1, Gen2, and Gen3 PCS, and provides support for PIPE 3.0 features.

The overall simplified PCIe Gen3 PCS the following architecture diagram. Note that the RX/TX Phase Comp FIFOs are physically placed in, and shared with the standard 8GB PCS.

Figure 1-38: PCIe Gen3 PCS Top Level Block Diagram



Transmitter PCIe Gen3 PCS Datapath

This section describes the transmitter channel PCIe Gen3 PCS datapath architecture.

Phase Compensation FIFO (Shared with Standard PCS)

Related Information

Transmitter Phase Compensation FIFO on page 1-35

Scrambler

The Scrambler is an additive scrambler on a per-lane basis with degree 23 polynomial linear feedback shift register (LFSR), with different taps for the 8 adjacent lanes. The scrambler is used to provide enough edge density, since there is no 8B/10B encoding in PCIe Gen 3, so that the RX PMA CDR can lock to the incoming data stream and generate the recovered clock.

Encoder

The PCIe Gen 3 base specification defines that the data packets have to be scrambled and descrambled, whereas the Ordered Set packets (except the first symbol of TS1 and TS2 Ordered Set) do not have to be scrambled or descrambled. The Encoder/Decoder continuously checks the header and payload of the packet and generates a signal to enable the scrambler/descrambler based upon whether the payload is an ordered set, or data packet. It also generates a signal to reset the scrambler/descrambler to the initial seed value if an Electrical Idle Exit Ordered Set or a Fast Training Sequence Ordered Set is received or transmitted. In addition, the encoder/decoder logic monitors the Ordered Set and the header for invalid values, and generates an error flag if they do.

Gearbox

The PCIe 3.0 base specification specifies a block size of 130 bits, with the exception of SKP Ordered Sets which can be variable length. An implementation of a 130-bit data path would take up significant resources, so the PCIe Gen 3 PCS data path is implemented as 32 bits wide. As the TX PMA data width is fixed to 32 bits, and the block size is 130 bits with variations, a gearbox is needed to convert the 130 bits to 32 bits. This gearbox has a transmitter bit-slip feature.

Receiver PCIe Gen3 PCS Datapath

This section describes the receiver channel PCIe Gen3 PCS datapath architecture.

Block Sync

PMA parallelization occurs at arbitrary word boundaries. Consequently, the parallel data from the RX PMA CDR must be realigned to meaningful character boundaries. The Block Sync module searches for the Electrical Idle Exit Sequence Ordered Set (or the last number of fast training sequences (NFTS) Ordered Set) and skip (SKP) Ordered Set to identify the correct boundary for the incoming stream and achieve the block alignment. The block is realigned to the new block boundary following the receipt of a SKP Ordered Set, as it can be of variable length.

Rate Match FIFO

The Rate Match FIFO (or clock compensation FIFO) compensates for minute frequency differences between the local clock (sometimes referred to as PLD soft IP clock, or PLD system clock) and the recovered clock. This is achieved by inserting and deleting SKP characters in the data stream to keep the FIFO from going empty or full respectively.

Decoder

The Decoder checks for decode errors in the data stream. It also enables or disables the Descrambler based on the Data and Ordered Set received.

Descrambler

The Descrambler descrambles data per the PCIe Gen3 specification.

PIPE Interface

The PIPE Interface block provides PIPE Gen3 compliant control and status interfaces between the high-speed serial interface (HSSI) and upper layer logic. It is PIPE 3.0 compliant.

Auto Speed Negotiation Block

The Auto Speed Negotiation block controls the operating speed of the HSSI when operating under PIPE 3.0 modes. By monitoring the rate control signal from the PhyMac, this block will change the HSSI from

PCIe Gen 1 operation mode, to Gen 2 operation mode, or from PCIe Gen 1 operation mode to Gen 2 operation mode to Gen 3 operation mode, or vice versa, with all the appropriate settings.

Electrical Idle Inference Block

In conjunction with side band signals from the PLD side, the Electrical Idle Inference Block infers Electrical Idle assuming that the signal detect is not reliable. This is based on the PCIe Base Specification Revision 2.0/3.0.

Clock Data Recovery (CDR) Control Block

The CDR control block is used for Rx.L0s fast exit when operating in PIPE/PCIe Gen 3 mode. Upon detecting an Electrical Idle Ordered Set (EIOS), it takes manual control of the CDR by forcing it into a lock to reference mode. When an exit from electrical idle is detected, this block moves the CDR into lock to data mode to achieve fast data lock.

Channel Bonding

The following factors contribute to the transmitter channel-to-channel skew:

- High-speed serial and low-speed parallel clock skew between channels
- Unequal latency in the transmitter phase compensation FIFO

Bonded transmitter datapath clocking provides low channel-to-channel skew when compared with non-bonded channel configurations.

Related Information

For more information about the bonded and non-bonded channel clocking, see Transceiver Clocking in Arria V Devices.

Bonded Channel Configurations

In bonded channel configurations, the serial and parallel clocks are generated by the transmit PLL and the central clock divider.

There is equal latency in the transmitter phase compensation FIFO of all bonded channels because they share common pointers and control logic generated in the central clock divider.

The lower transceiver clock skew and equal latency in the transmitter phase compensation FIFO in all channels result in a lower channel-to-channel skew.

Note: Bonded channel configurations are available only for serial data rates up to 6.5536 Gbps in Arria V GX/GT/SX/ST devices and up to 12.5 Gbps in Arria V GZ devices. You can bond (up to) all channels on the same side.

Non-Bonded Channel Configurations

In a non-bonded channel configuration, the parallel clock in each channel is generated independently by its local clock divider.

There may be unequal latency in the transmitter phase compensation FIFO of each channel because each channel has its own pointers and control logic. The higher transceiver clock skew and unequal latency in the transmitter phase compensation FIFO in each channel may result in a higher channel-to-channel skew.

PLL Sharing

In a Quartus II design, you can merge two different protocol configurations to share the same CMU PLL resources. These configurations must fit in the same transceiver bank. The input `refclk` and PLL output frequencies must be identical.

Document Revision History

The revision history for this chapter.

Table 1-36: Document Revision History

Date	Version	Changes
May 2020	2020.05.29	Made the following change: • Removed continuous as an AEQ mode option for Continuous Time Linear Equalization (CTLE).
April 2019	2019.04.04	Made the following changes: • Updated "8B/10B Decoder Block Diagram in Single-Width Mode" figure.
January 2016	2016.01.28	Made the following change: • Added the "Calibration Block Boundary" section.
September 2014	2014.09.30	• Changed the *Transceiver Bank and PCIe Hard IP Location for GT Devices* figure to show Ch0 of GXB_L1 as a 10-Gbps channel. Also added notes to the figure. • Added an example calculation for the maximum supported data rate and a related link to the *Arria V Device Datasheet* in the *Transceiver Channel Architecture* section. • Added a description of `rx_syncstatus` to the *Word Aligner in Manual Alignment Mode* section. • Added a description of `rx_syncstatus` to the *Word Aligner in Deterministic Latency State Machine Mode* section. • Changed "MegaWizard Plug-in Manager" to "Parameter Editor" in the *10G PCS Architecture for Arria V GZ Devices* and *Frame Generator* sections.

Date	Version	Changes
March 2014	2014.03.07	• Updated Table 1-7. • Updated the *Transmitter Buffer* section. • Updated the *Word Aligner* section. • Updated the *Word Aligner in Deterministic Latency State Machine Mode* section. • Updated the *Clock Divider* section. • Updated Figure 1-28. • Added a note to the *Rate Match FIFO* section.
October 2013	2013.10.18	• Updated the *Transceiver Architecture in Arria V Devices* section. • Updated the *Enhanced Small Form-Factor Pluggable (SFP+) Interface* section. • Updated the *Channel PLL Architecture* section. • Updated Table 1-1. • Updated Table 1-6. • Updated Table 1-22.
May 2013	2013.05.06	• Added link to the known document issues in the Knowledge Base • Updated Figure 1-1. • Updated Table 1-6. • Updated the *Adaptive Equalization Mode* section. • Updated the *Transmitter Buffer* section. • Updated the *Receiver Buffer* section. • Updated the *Clock Divider* section. • Updated the *Word Aligner in Manual Alignment Mode* section. • Updated the *Bit Slip Mode* section. • Updated the *Auxiliary Transmit (ATX) PLL Architecture* section.

Date	Version	Changes
March 2013	2013.03.15	• Updated Figure 1-2. • Updated Figure 1-3, and clarified the data rate for rx-only channels. • Updated Figure 1-7, and clarified the data rate for rx-only channels. • Updated *Transceiver Banks* . • Added *10-Gbps Support Capability in GT and ST Devices*. • Added *Enhanced Small Form-Factor Pluggable (SFP+) Modules*. • Added *10GBase-KR Support*. • Added *9.8 Gbps CPRI Application*. • Added *Transceiver Channel Architecture*.
November 2012	2012.11.19	Reorganized content and updated template
June 2012	1.2	• Merged information from Transceiver Basics for Arria V Devices, version 1.1 into this chapter.

Transceiver Clocking in Arria V Devices

2

2016.01.08

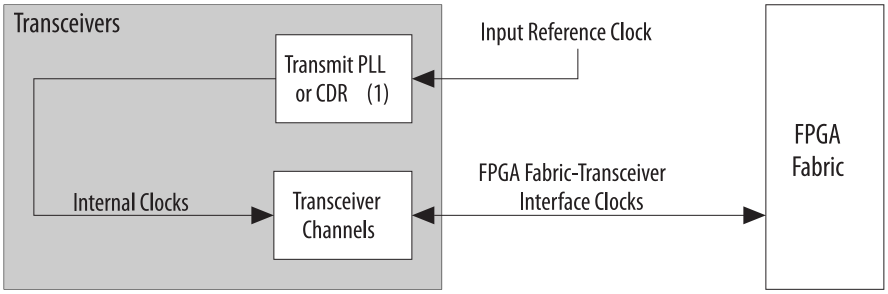
AV53002 Subscribe Send Feedback

This chapter provides information about the Arria V transceiver clocking architecture. The chapter describes the clocks that are required for operation, internal clocking architecture, and clocking options when the transceiver interfaces with the FPGA fabric.

Notes:

- Bonded configuration refers to PMA bonding for Arria V devices. The split between PCS and PMA bonding is done in Arria 10 devices only.
- Channels need to be contiguous when using PMA bonding.

Figure 2-1: Transceiver Clocking Architecture Overview



Note: (1) The transmit phase-locked loop (PLL) can be a CMU PLL (channel PLL), fPLL (fractional PLL Clock) or an ATX PLL (for Arria V GZ devices only.)

Related Information

Arria V Device Handbook: Known Issues
Lists the planned updates to the *Arria V Device Handbook* chapters.

Input Reference Clocking

The reference clock for the transmitter PLL and CDR generates the clocks required for transceiver operation.

Table 2-1: Input Reference Clock Sources

Sources	Transmitter PLL				CDR	Jitter Performance [11]
	ATX PLL [12]	CMU PLL > 6.5536 Gbps [13]	CMU PLL <= 6.5536 Gbps	fPLL		
Dedicated refclk pin	Yes	Yes	Yes	Yes	Yes	1
REFCLK network	Yes	Yes	Yes	Yes	Yes	2
Dual-purpose `RX/REFCLK` pin	Yes	No [14]	Yes	Yes	Yes	3
Fractional PLL	Yes [15]	Yes	Yes	Yes	Yes	4
Generic `CLK` pin	No	No	No	No	No	5
Core clock network (GCLK, RCLK, PCLK)	No	No	No	No	No	6

[11] The lower number indicates better jitter performance.

[12] ATX PLL is available only in Arria V GZ devices.

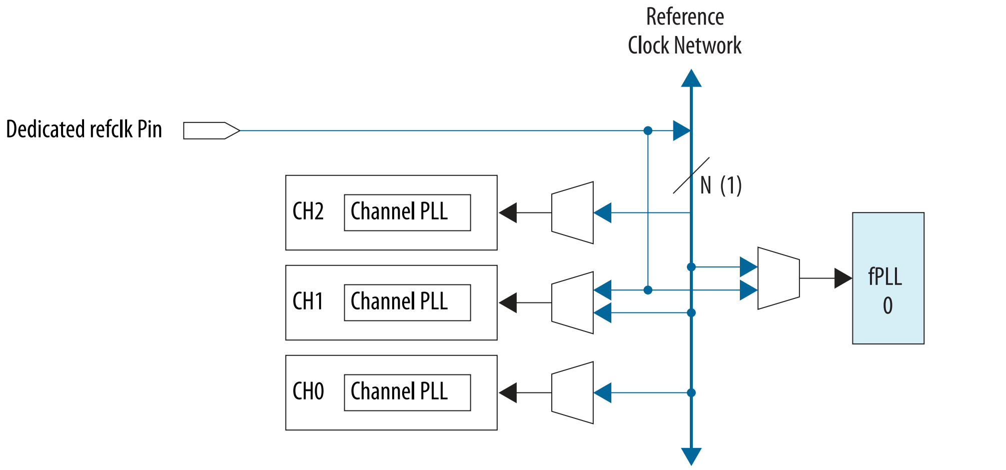
[13] Applicable for 10 Gbps channels only in GT and ST devices and for 12.5 Gbps channels in GZ devices. For better jitter performance, use dedicated refclk pins for data rates > 6.5536 Gbps.

[14] For Arria V GZ devices, the dual-purpose `RX/REFCLK` pin can be used as a reference clock source for CMU PLL with data rates > 6.5536 Gbps.

[15] fPLL to ATX PLL cascading is only enabled for SDI applications

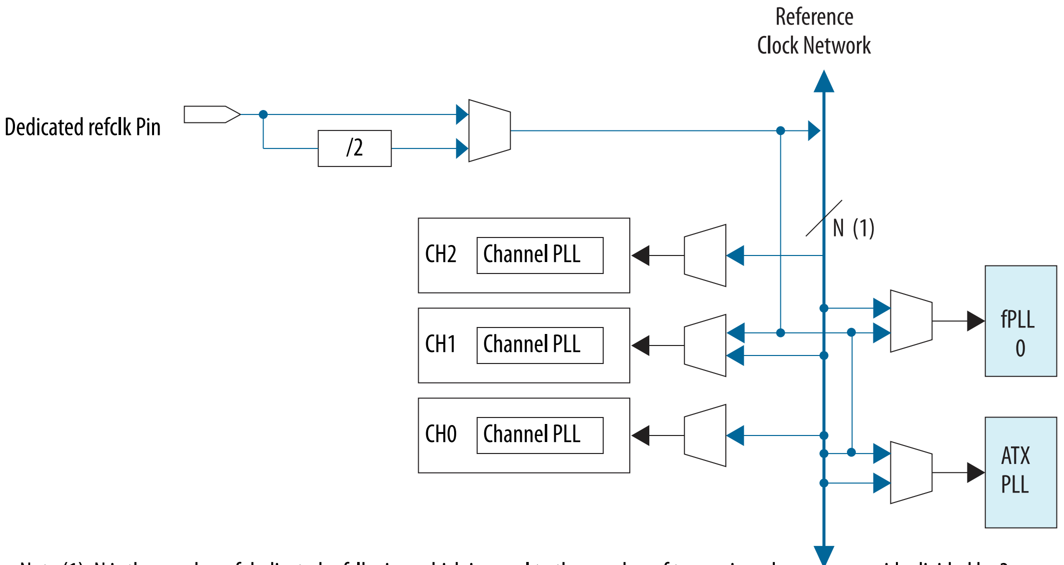
Figure 2-2: Dedicated refclk Pin and Reference Clock Network

The following figure shows the dedicated refclk pin connection to channel PLL. The direct refclk pin connection to channel PLL (which can either be configured as CMU PLL or CDR) is only available in channel 1 and 4 in a bank.



Note (1): N is the number of dedicated refclk pins, which is equal to the number of transceiver channels on a side divided by 3.

Figure 2-3: Dedicated refclk Pin and Reference Clock Network for Arria V GZ Devices



Note (1): N is the number of dedicated refclk pins, which is equal to the number of transceiver channels on a side divided by 3.

Note: ATX PLL is available only for Arria V GZ devices.

Figure 2-4: Input Reference Clock Source for CMU PLL Driving Channels with Serial Data Rates Beyond 6.5536 Gbps

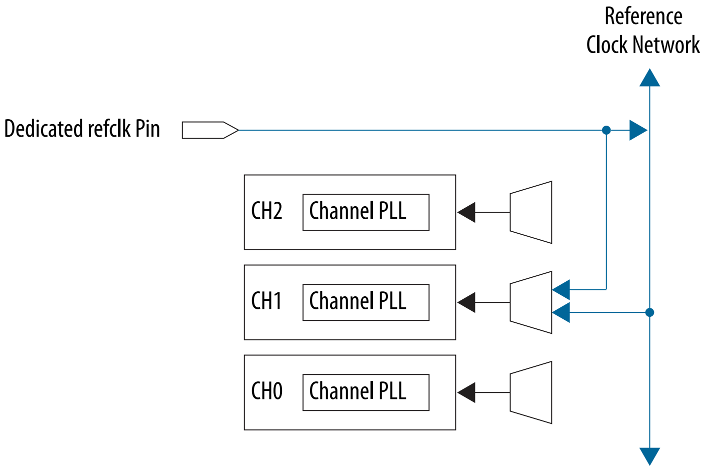


Table 2-2: Electrical Specifications for the Input Reference Clock for Arria V GZ Devices

Protocol	I/O Standard	Coupling	Termination
PCI Express (PCIe)	• 1.2V PCML, 1.4 PCML • 1.4V PCML • 1.5V PCML • 2.5V PCML • Differential LVPECL • LVDS	AC	On - Chip [16]
	• HCSL [17]	DC	Off - Chip [18]
All other protocols	• 1.2V PCML, 1.4 PCML • 1.4V PCML • 1.5V PCML • 2.5V PCML • Differential LVPECL • LVDS	AC	On - Chip [16]

[16] For more information about termination values supported, refer to the *DC Characteristics* section in *Arria V Device Datasheet.*

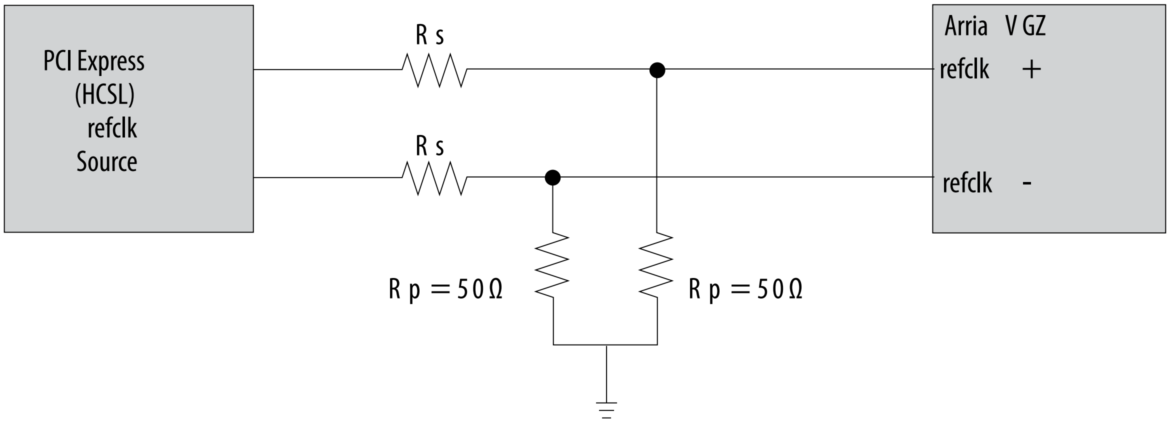
[17] In PCIe mode, you have the option of selecting the HCSL standard for the reference clock if compliance to the PCIe protocol is required. You can select this I/O standard option only if you have configured the transceiver in PCIe mode.

[18] For an example termination scheme refer to **Figure 2-5**

Note: If you select the HCSL I/O standard for the PCIe reference clock, add the following assignment to your project's Quartus settings file (.qsf):

```
set_instance_assignment -name XCVR_REFCLK_PIN_TERMINA-
TION_DC_COUPLING_EXTERNAL_RESISTOR -to <refclk_pin_name>
```

Figure 2-5: Termination Scheme for a Reference Clock Signal When Configured as HCSL for Arria V GZ Devices



Note:
- No biasing is required if the reference clock signals are generated from a clock source that conforms to the PCIe specification.
- Select Rs and / or Rp resistor values as recommended by the PCIe clock source vendor.

Related Information

Arria V Device Datasheet

Reference Clock Network

The dedicated refclk pin can provide the reference clock to multiple channel PLLs, fractional PLLs or an ATX PLL (for Arria V GZ devices).

Designs that use multiple transmitter PLL and CDRs with the same input reference clock frequency can share the same dedicated refclk pin. Each dedicated refclk pin can drive any transmitter PLL or CDR on the same side of device through the reference clock network.

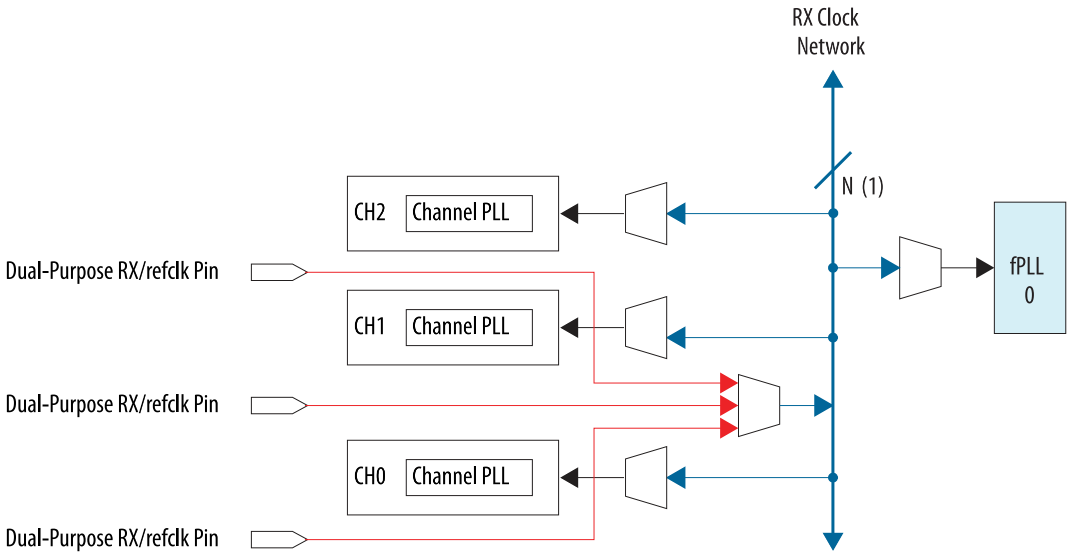
Dual-Purpose RX/refclk Pin

When not used as receiver, an RX differential pair can be used as an additional input reference clock source. The clock from the RX pins feed the RX clock network that spans all the channels on one side of the device.

Only one RX differential pair for every three channels can be used as input reference clock at a time. The following figure shows the use of dual-purpose RX/refclk differential pin as input reference clock source and the RX clock network.

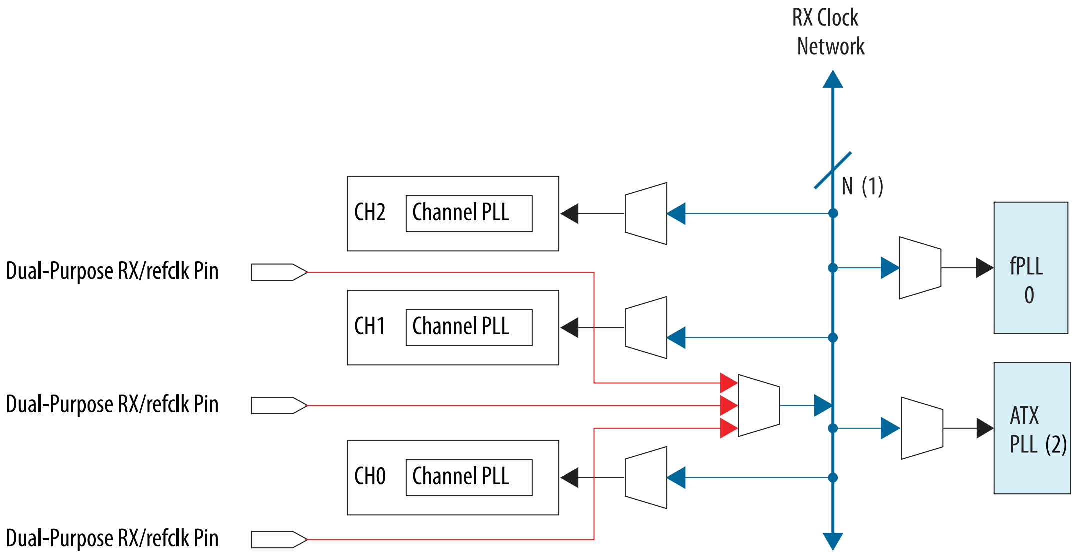
Note:
- An RX differential pair from another bank can be used as an input reference clock pin on the same side of the device.
- refclk switching cannot be performed when dual-purpose RX differential pins are used as refclk pins.

Figure 2-6: Dual-Purpose RX/refclk Pin as an Input Reference Clock


RX Clock Network
N (1)
CH2
Channel PLL
Dual-Purpose RX/refclk Pin
fPLL 0
CH1
Channel PLL
Dual-Purpose RX/refclk Pin
CH0
Channel PLL
Dual-Purpose RX/refclk Pin

Note (1): N is the number of transceiver channels on a side divided by 3.

Figure 2-7: Dual-Purpose RX/refclk Pin as an Input Reference Clock for GZ Devices



Note (1): N is the number of transceiver channels on a side divided by 3.
Note (2): ATX PLL is available only for Arria V GZ devices.

Fractional PLL (fPLL)

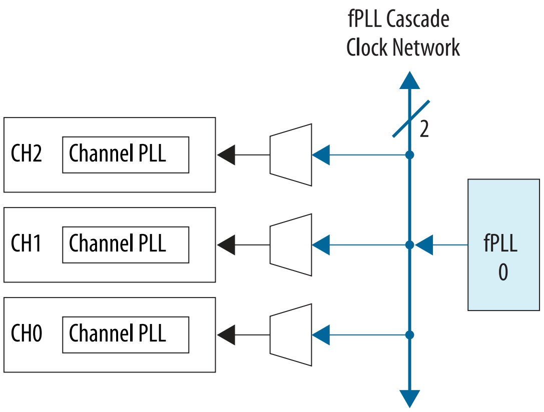
The fPLL clock output can be used as input reference clock source to transmitter PLL or CDR.

Cascading the fPLL to transmitter PLL or CDR enables you to use an input reference clock that is not supported by the transmitter PLL or CDR. The fPLL synthesizes a supported input reference clock for the transmitter PLL or CDR.

A fPLL is available for each group of three transceiver channels. Each fPLL drives one of two fPLL cascade clock network lines that can provide an input reference clock to any transmitter PLL or CDR on the same side of a device.

Note: An fPLL can also be used as a transmit PLL.

Figure 2-8: fPLL Clock Output as Input Reference Clock



Note: It is not recommended to use a fractional PLL in fractional mode for transceiver applications as a TX PLL or for PLL cascading.

Note: Transceiver associated power pins must be powered up. If a dedicated transceiver refclk pin is used as a clock reference for a core fractional PLL, then at least one transceiver must be instantiated in the design.

Internal Clocking

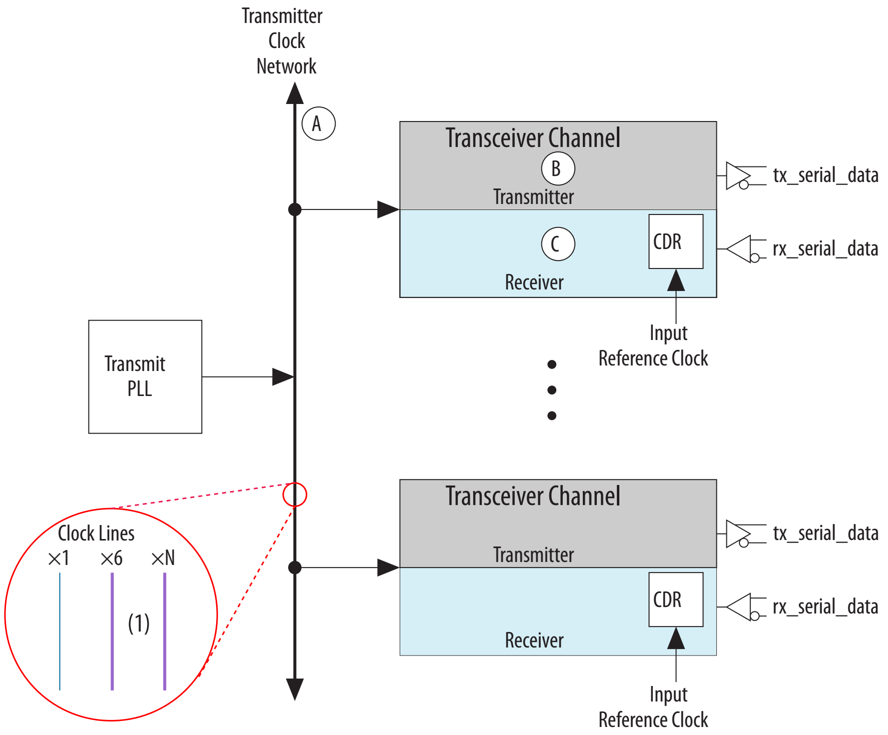
In the internal clocking architecture, different physical coding sublayer (PCS) configurations and channel bonding options result in various transceiver clock paths.

Table 2-3: Internal Clocking Subsections

The labels listed in the following table and shown in the figure following mark the three sections of the transceiver internal clocking.

Label	Scope	Description
A	Transmitter Clock Network	Clock distribution from transmitter PLLs to channels
B	Transmitter Clocking	Clocking architecture within transmitter channel datapath
C	Receiver Clocking	Clocking architecture within receiver channel datapath

Figure 2-9: Internal Clocking



Note:
(1) The x6 and xN clock lines are supported only by the Arria V 6-Gbps transceivers.

Note: For Arria V GZ devices, the x6 clock lines can support data rates up to 12.5 Gbps and the xN clock lines can support data rates up to 9.8304 Gbps.

Transmitter Clock Network

The transmitter PLL is comprised of the ATX PLL (for GZ devices only) , CMU PLL, and fPLL.

All CMU PLLs are identical in architecture, but vary in the following:

- Usage capability: CMU PLL in channel 1 and 4 are capable of distributing a clock with accessibility to the transmitter clock network, while the rest only are able to clock the transmitter in the same channel only.
- Performance: Up to 6.5536 Gbps in GX and SX, except for the CMU PLL of channel 1 and 4 in GT and ST devices, which are capable of up to 10.3125 Gbps. In GZ devices, the CMU PLL in channel 1 and channel 4 can drive any transceiver channel up to 12.5 Gbps.

Table 2-4: Usage Capability of Each ATX PLL (for GZ devices) and CMU PLL Within a Transceiver Bank

CMU PLL Location in a Transceiver Bank	Clock Network Access	Maximum ATX / CMU PLL Performance (Gbps)	Maximum CMU PLL Performance (Gbps)		Usage Capability
		GZ Devices only	GX and SX Devices	GT and ST Devices	
CH 0	No	12.5	6.5536	6.5536	Clock transmitter within same channel only
CH 1	Yes	12.5	6.5536	10.3125	Clock transmitter within same channel only and other channels via clock network
CH 2	No	12.5	6.5536	6.5536	Clock transmitter within same channel only
CH 3	No	12.5	6.5536	6.5536	Clock transmitter within same channel only
CH 4	Yes	12.5	6.5536	10.3125	Clock transmitter within same channel only and other channels via clock network
CH 5	No	12.5	6.5536	6.5536	Clock transmitter within same channel only

The fPLLs adjacent to the transceiver banks provide an additional transmitter PLL source for clocking transceivers up to 3.125 Gbps. Two fPLLs are available as the transmitter PLL for every transceiver bank of six channels, or one fPLL for a bank of three channels.

The transmitter clock network routes the clock from the transmitter PLL to the transmitter channel. As shown in Figure 2-9 , the transmitter clock network routes the clock from the transmit PLL to the transmitter channel. A clock divider provides two clocks to the transmitter channel:

- Serial clock—high-speed clock for the serializer
- Parallel clock—low-speed clock for the serializer and the PCS

Arria V transceivers support non-bonded and bonded transceiver clocking configurations:

- Non-bonded configuration—Only the serial clock from the transmit PLL is routed to the transmitter channel. The clock divider of each channel generates the local parallel clock. The x1 and xN (for Native PHY IP only) clock lines are used for non-bonded configurations. This configuration is available for both the 6-Gbps, 10-Gbps and 12.5 Gbps (GZ devices only) transceivers.
- Bonded configuration—Both the serial clock and parallel clock are routed from the central clock divider in channel 1 or 4 to the bonded transmitter channels. The x6 and xN clock lines are used for bonded configurations. This configuration is only available for 6 Gbps transceivers. Arria V GZ devices can support data rates upto 12.5 Gbps on the x6 clock lines and 8 Gbps using PCIe or 9.8304 Gbps using Native PHY IP on the xN clock lines.

The transmitter clock network is comprised of x1 (x1 and x1_fPLL), x6 and xN clock lines.

Table 2-5: Characteristics of x1, x6, and xN Clock Lines

Characteristics	x1	x1_fPLL	x6	x6_fPLL	xN
Clock Source	CMU PLL from CH 1 or CH 4 in a bank (serial clock only)	fPLL adjacent to transceivers (serial clock only)	Central clock divider from Ch 1 or Ch 4 in a bank (serial and parallel clock)	fPLL through the `x1_fPLL` line. The central clock divider resource of Ch 1 or Ch 4 in a bank is used (serial and parallel clock). However, the Ch 1 or Ch 4 can still be used as the receiver CDR.	x6 clock lines (serial and parallel clock)
Max Data Rate (Gbps)	10.3125 (GT and ST) 6.5536 (GX and SX)	3.125	6.5536	3.125	3.25 [19]
Clock Line Span	Within a transceiver bank	Within a group of 3 channels (0, 1, 2 or 3, 4, 5)	Within a transceiver bank	Within a transceiver bank	Across all channels in the same side of device
Non-bonded Configuration	Yes	Yes	Yes	No	Yes
Bonded Configuration	No	No	Yes	Yes	Yes

[19] Only for PCIe Gen2 configurations, xN clock lines can support a maximum data rate of 5 Gbps. There is no xN data rate limit check in the Quartus II software. The limit in the handbook is the golden reference.

Table 2-6: Data Rates and Spans Supported by Clock Sources and Clock Networks in Arria V GZ Devices

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
x1	GX	ATX PLLs in a transceiver bank	12.5 Gbps [20]	No	Transceiver bank
		CMU PLLs in a transceiver bank	12.5 Gbps [20]		Transceiver bank
		Fractional PLLs in a transceiver bank	3.125 Gbps		fPLLs can only span upper or lower 3 channels in a transceiver bank.
xN (Native PHY)	GX	ATX PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive only the serial clock from the x6 clock lines.	8 Gbps	No	xN lines span a side of the device. Specified datarate can drive up to 13 data channels above and up to 13 data channels below TX PLL.
		Channel PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive only the serial clock from the x6 clock lines.	7.99 Gbps		
		Fractional PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive only the serial clock from the x6 clock lines.	3.125 Gbps		
x6	GX	ATX PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The x6 clock lines receive both the serial and	12.5 Gbps [20]	Yes	Transceiver bank

[20] For the fastest speed grade only. For the remaining speed grades, refer to the *Arria V Device Datasheet* .

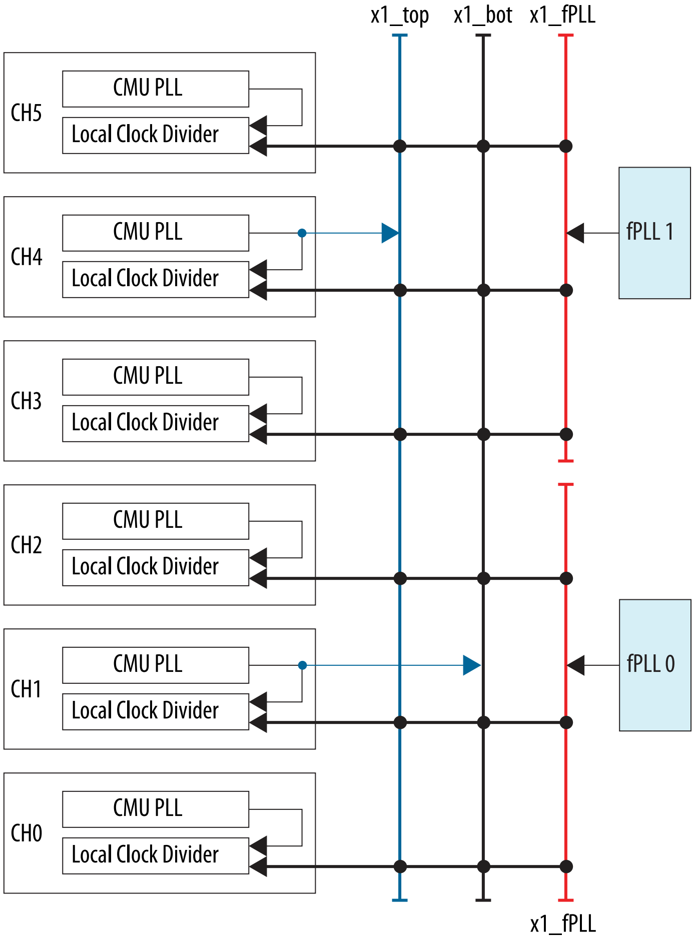
Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
		parallel clock from the central clock dividers.			
		The channel (CMU) PLLs provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The x6 clock lines receive both the serial and parallel clock from the central clock dividers.	12.5 Gbps [20]		
		Fractional PLLs provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The x6 clock lines receive both the serial and parallel clock from the central clock dividers.	3.125 Gbps		
x6 PLL Feedback Compensation [21]	GX	One ATX PLL per bonded transceiver bank provides a serial clock to the central clock dividers of Ch 1 and Ch 4. The central clock dividers in the transceiver bank drive the x6 clock lines and provide feedback path to the ATX PLL. The x6 clock lines receive both the serial and parallel clocks from the central clock dividers.	12.5 Gbps [20]	Yes	x6 lines span a transceiver bank. The x6 lines across multiple transceiver banks can be bonded together through PLL feedback compensation path to span the entire side of the device.
		One CMU PLL per bonded transceiver bank provides a serial clock to the central clock dividers of Ch 1 and Ch 4. The central clock dividers in the transceiver bank drive the x6 clock lines and provide feedback path to the CMU PLL. The x6 clock lines receive both the serial and parallel clocks from the central clock dividers.	12.5 Gbps [20]		

[21] The input reference clock frequency of the transmit PLL must be the same as the parallel clock frequency which clock the PCS bonded channels.

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
xN (PCIe)[22]	GX	The ATX or channel (CMU) PLL provides a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive the serial and parallel clocks from the x6 clock lines.	8 Gbps	Yes	xN lines span a side of the device, but can bond only up to eight contiguous data channels.
xN (Native PHY)	GX	ATX PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive the serial and parallel clocks from the x6 clock lines.	9.8304 Gbps	Yes	xN lines span a side of the device. Specified datarate can bond up to 7 contiguous data channels above and up to 7 contiguous data channels below TX PLL.
			8 Gbps	Yes	xN lines span a side of the device. Specified datarate can bond up to 13 contiguous data channels above and up to 13 contiguous data channels below TX PL
		Channel (CMU) PLLs in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive the serial and parallel clocks from the x6 clock lines.	7.99 Gbps	Yes	xN lines span a side of the device. Specified datarate can bond up to 13 contiguous data channels above and up to 13 contiguous data channels below TX PL
		Fractional PLLs (fPLLs) in a transceiver bank provide a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive the serial and parallel clocks from the x6 clock lines.	3.125 Gbps		

[22] For more information about PCIe x8 configurations, refer to the *Transceiver Configurations in Arria V GZ Devices* chapter.

Figure 2-10: x1 Clock Line Architecture (up to 6.5536 Gbps)

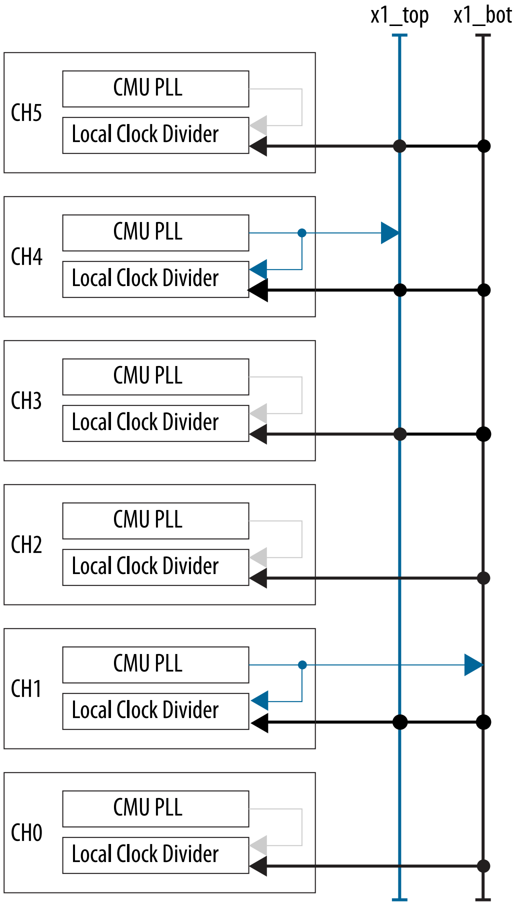


Note: All clock lines shown in this figure carry the serial clock only. x1_fPLL can support data rates upto 3.125 Gbps only.

The x1 clock lines are driven by serial clocks of CMU PLLs from channels 1 and 4. The serial clock in the x1 clock line is then distributed to the local and central clock dividers of every channel within a transceiver bank.

The x1_fPLL clock lines are driven by the serial clocks of the adjacent fPLL. The serial clock in the x1_fPLL clock lines, is then distributed to the local and central clock dividers of channels within a group of three channels (0, 1, 2 or 3, 4, 5).

Figure 2-11: x1 Clock Line Architecture (more than 6.5536 Gbps)

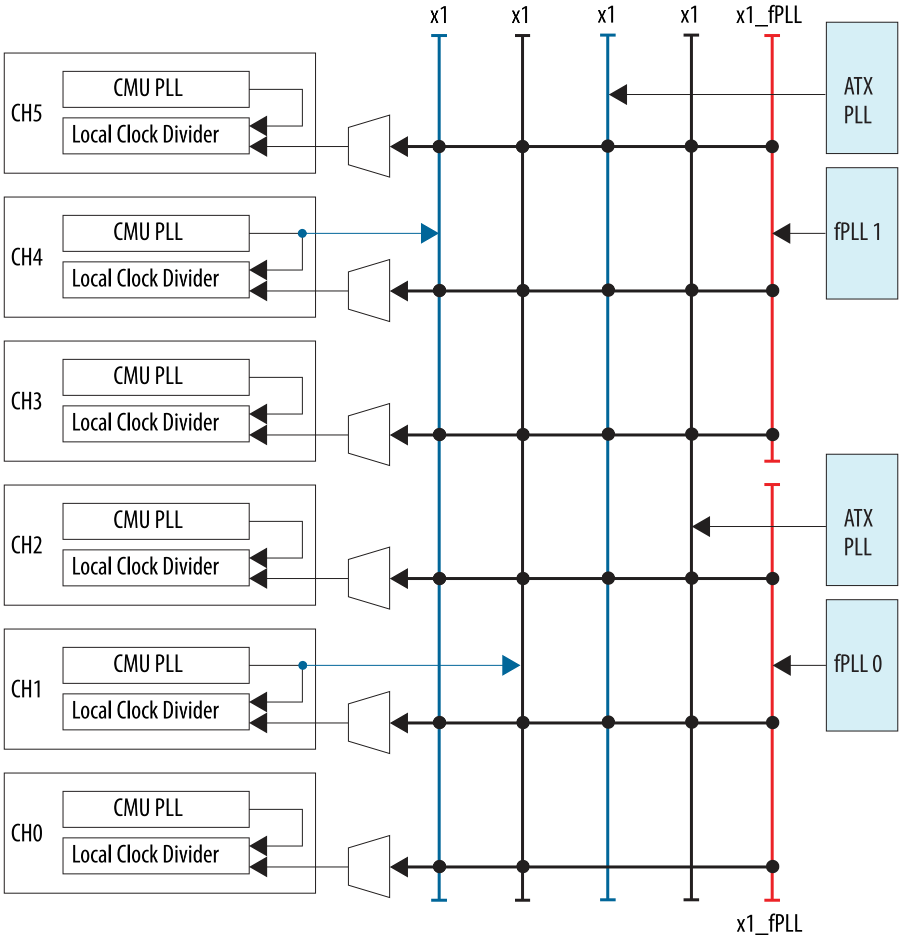


Note: All clock lines shown in this figure carry the serial clock only.

For serial data rates beyond 6.5536 Gbps (10-Gbps channels only in GT and ST devices). The x1 clock lines are driven by the serial clocks of CMU PLLs from channels 1 and 4. The serial clock in the x1 clock line is then distributed to the local and central clock dividers of every channel within a transceiver bank.

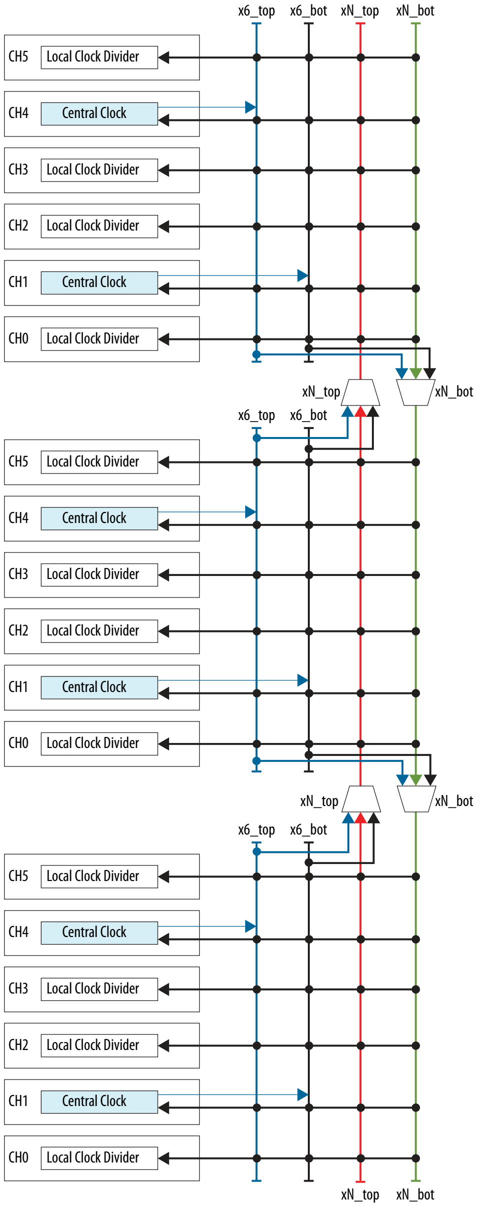
Note: When you configure the channel PLL as a CMU PLL to drive the local clock divider, or the central clock divider of its own channel, you cannot use the channel PLL as a CDR. Without a CDR, you can use the channel only as a transmitter channel.

Figure 2-12: x1 Clock Line Architecture (up to 12.5 Gbps) for GZ Devices



Note: All clock lines shown in this figure carry the serial clock only. In Arria V GZ devices, fPLLs support data rate up to 3.125 Gbps only.

Figure 2-13: x6 and xN Clock Line Architecture



Note: All the clock lines shown in this figure carry both the serial and parallel clocks.

The x6 clock lines are driven by serial and parallel clocks from the central clock divider in channels 1 and 4. For channels within a bank, the serial and parallel clocks in the x6 clock line is then distributed to every channel within a transceiver bank.

The xN clock lines extend the clocking reach of the x6 clock line to all channels within the same side of the device. To reach a xN clock line, the clocks must be provided on the x6 clock line. The serial and parallel clocks in the x6 clock line is distributed to every channel within a transceiver bank. The serial and parallel clocks are distributed to other channels beyond the bank using the xN clock line.

In bonded configurations, serial and parallel clocks from the x6 or xN clock lines are received by the clock divider of every bonded channel and fed directly to the serializer. In a non-bonded configuration, the clock divider of every non-bonded channel receives the serial clock from the x6 or xN clock lines and generates the individual parallel clock to the serializer.

Note:
- In a bonded configuration, bonded channels must be placed contiguously without leaving a gap between the channels, except when the gap channel is a CMU PLL.
- xN bonded configuration is only supported by PIPE and Native PHY IP.

Related Information

- Transceiver Configurations in Arria V GZ Devices
- Arria V Device Datasheet

Transmitter Clocking

Transmitter (TX) clocking refers to the clocking architecture that is internal to the TX channel of a transceiver.

The following figure shows how the clock divider provides the serial clock to the serializer, and the parallel clock to the serializer and TX PCS. When the byte serializer is not used, the parallel clock is used to clock all the blocks up to the read side of the TX phase compensation FIFO. For configurations with the byte serializer, the parallel clock is divided by a factor of two for the byte serializer and the read side of the TX phase compensation FIFO. The read side clock of the TX phase compensation FIFO is also forwarded to the FPGA fabric to interface the FPGA fabric with the transceiver.

Figure 2-14: Clocking Architecture for Transmitter PCS and PMA Configuration

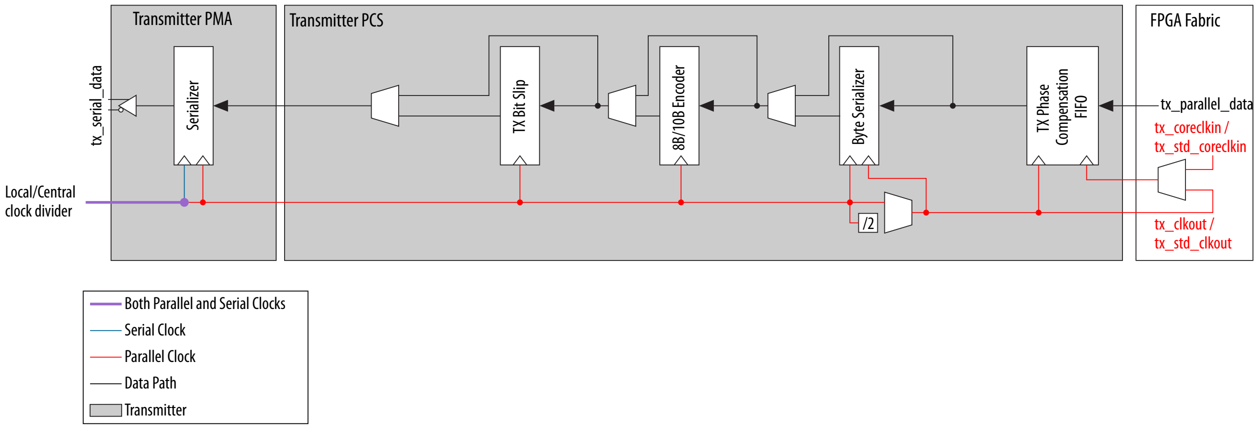


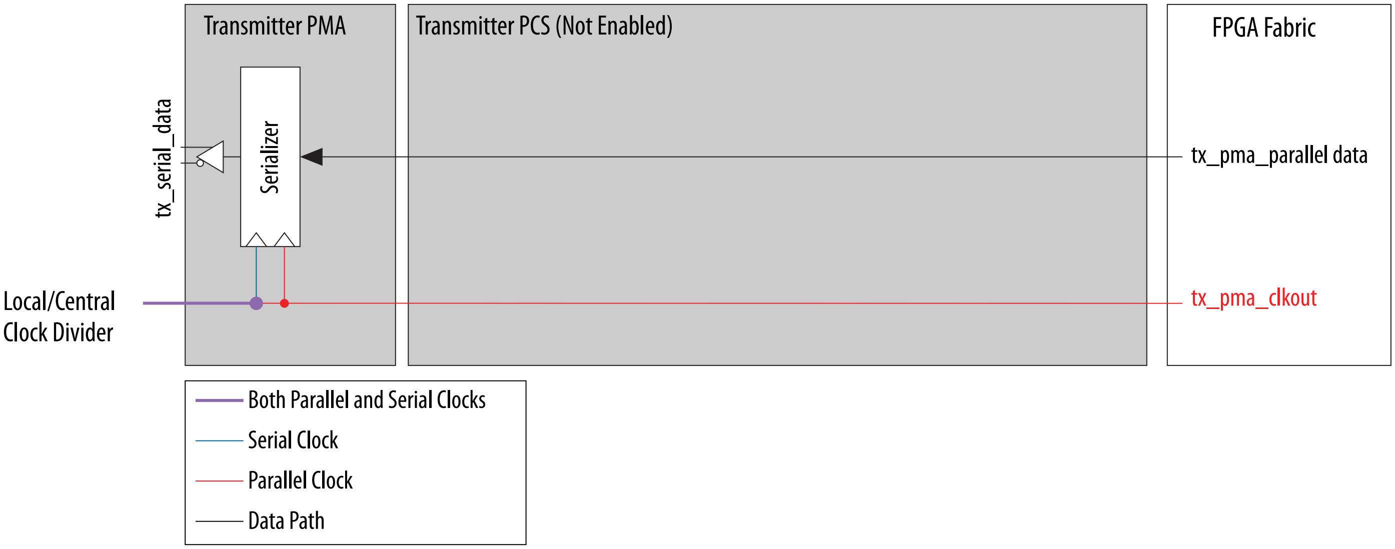
Table 2-7: Clock Sources for All TX PCS Blocks

PCS Block	Side	Clock Source
TX Phase Compensation FIFO	Write	FPGA fabric write clock, driven either by tx_clkout or tx_coreclkin
	Read	Parallel clock (divided). Clock forwarded to FPGA fabric as tx_clkout
Byte Serializer	Write	Parallel clock (divided) either by factor of 1 (not enabled), or factor of 2 (enabled)
	Read	Parallel clock
8B/10B Encoder	—	Parallel clock

PCS Block	Side	Clock Source
TX Bit Slip	—	Parallel clock

The following figure shows how the clock divider provides the serial and parallel clock to the serializer in a transmitter PMA configuration. The parallel clock is forwarded to the FPGA fabric to interface the FPGA fabric with the TX PMA directly, bypassing the PCS blocks.

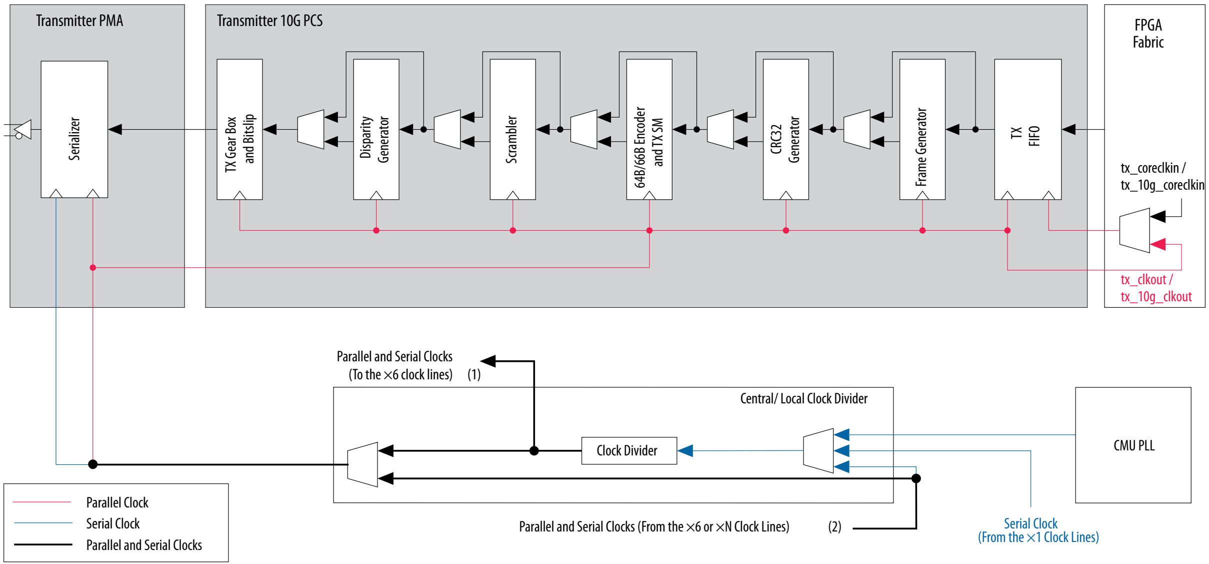
Figure 2-15: Clocking Architecture for Transmitter PMA Only Configuration



Transmitter 10G PCS Clocking for GZ Devices

The following figure shows the clocking scheme for the transmitter 10G PCS and transmitter physical medium attachment (PMA). The clock divider block provides the serial clock to the serializer of the transmitter PMA and the parallel clock to the transmitter PCS. In the 10G PCS channel, the parallel clock is used by all the blocks up to the read side of the transmitter (TX) FIFO.

Figure 2-16: Transmitter 10G PCS Clocking for GZ Devices


Transmitter PMA
Transmitter 10G PCS
FPGA Fabric
Serializer
TX Gear Box and Bitslip
Disparity Generator
Scrambler
64B/66B Encoder and TX SM
CRC32 Generator
Frame Generator
TX FIFO
tx_coreclkin / tx_10g_coreclkin
tx_clkout / tx_10g_clkout
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Central/ Local Clock Divider
Clock Divider
CMU PLL
Parallel Clock
Serial Clock
Parallel and Serial Clocks
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines) (2)
Serial Clock (From the ×1 Clock Lines)

Notes: (1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.
(2) x1 clock lines can be driven by a CMU PLL, ATX PLL, or a fractional PLL.

Non-Bonded Channel Configurations

The channel clock path for non-bonded configurations can be driven by the x1, or the x6 and xN clock lines.

Table 2-8: Clock Path for Non-Bonded Configurations

The following table describes the clock path for non-bonded configuration with the ATX PLL, CMU PLL, and fPLL as TX PLL using various clock lines.

Clock Line	Transmitter PLL	Clock Path
x1	ATX PLL[23]	ATX PLL » x1 » individual clock divider » serializer
	CMU	CMU PLL » x1 » individual clock divider » serializer
	fPLL	fPLL » x1_fPLL » individual clock divider » serializer
x6, xN	ATX PLL[23]	ATX PLL » central clock divider » x6 » xN » individual clock divider » serializer
	CMU	CMU PLL » central clock divider » x6 » xN » individual clock divider » serializer [24]
	fPLL	fPLL » x1_fPLL » central clock divider » x6 » individual clock divider » serializer [24]

[23] ATX PLL is available only for GZ devices.

[24] Non-bonded channels within same bank as TX PLL are driven by clocks from x6 clock line, and channels in other banks are driven from xN clock line.

Figure 2-17: Three Non-Bonded Transmitter Channels Driven by CMU PLL using x1 Clock Line Within a Transceiver Bank

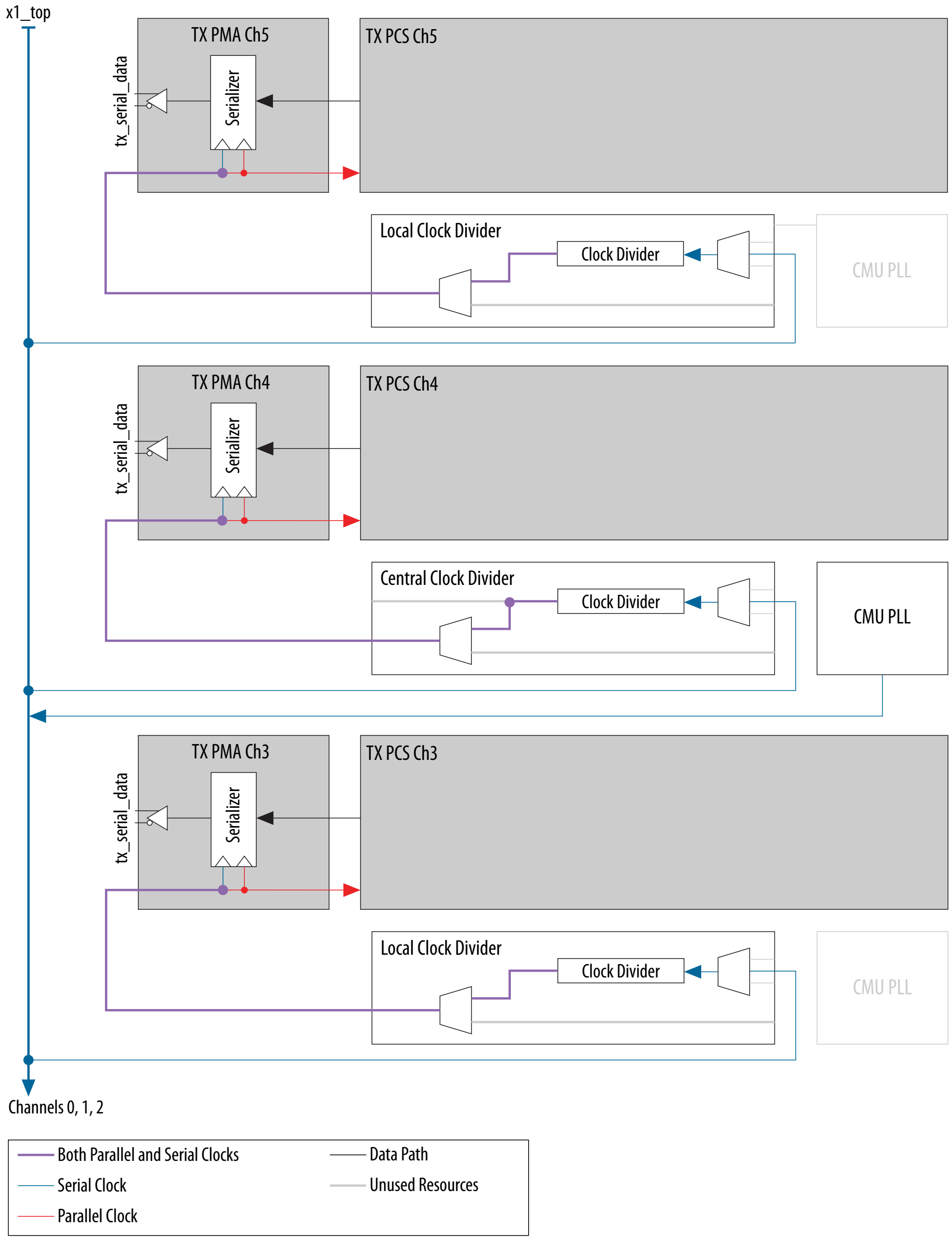


Figure 2-18: Three Non-Bonded Transmitter Channels Driven by fPLL using x1 Clock Line Within a Transceiver Bank

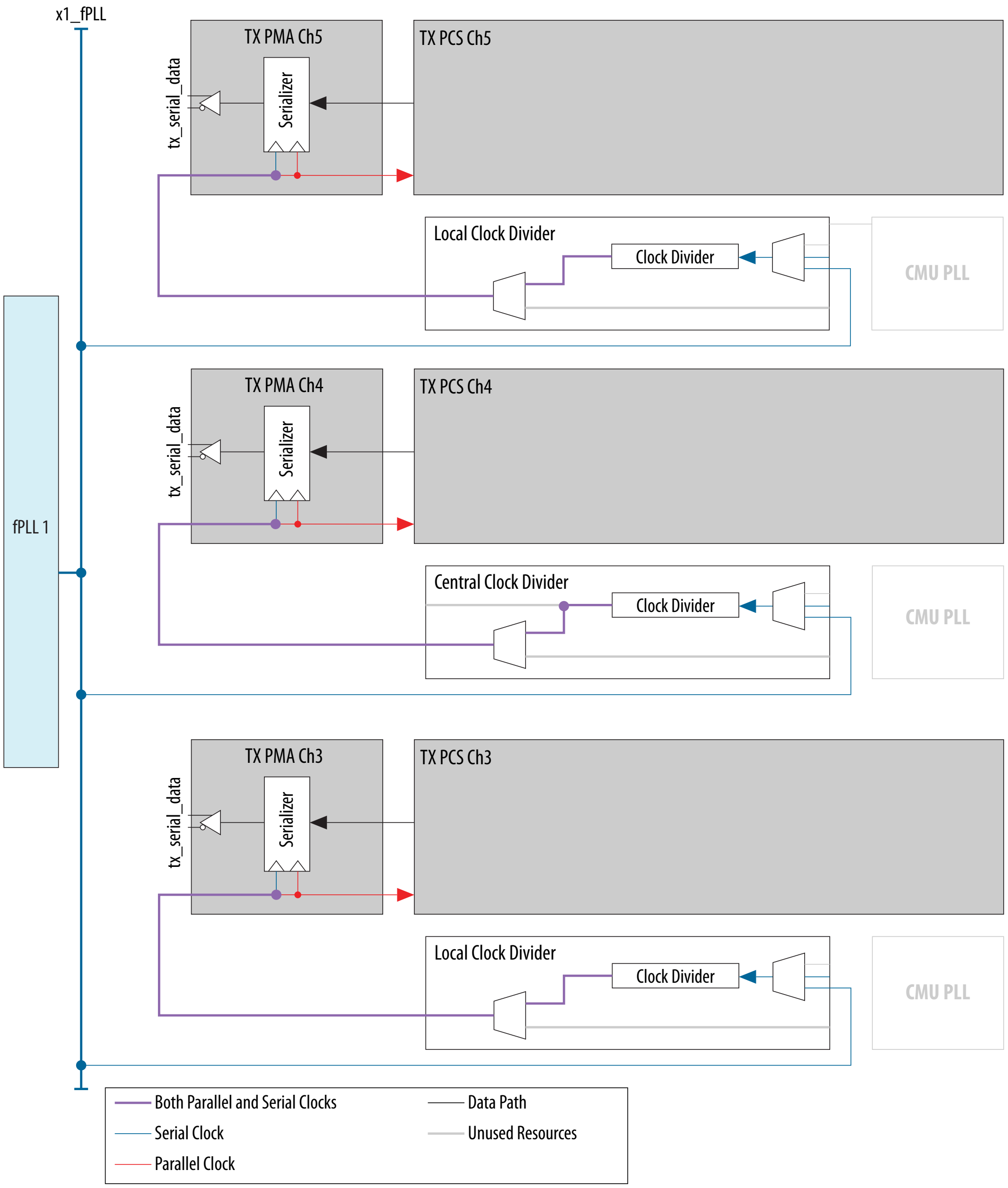
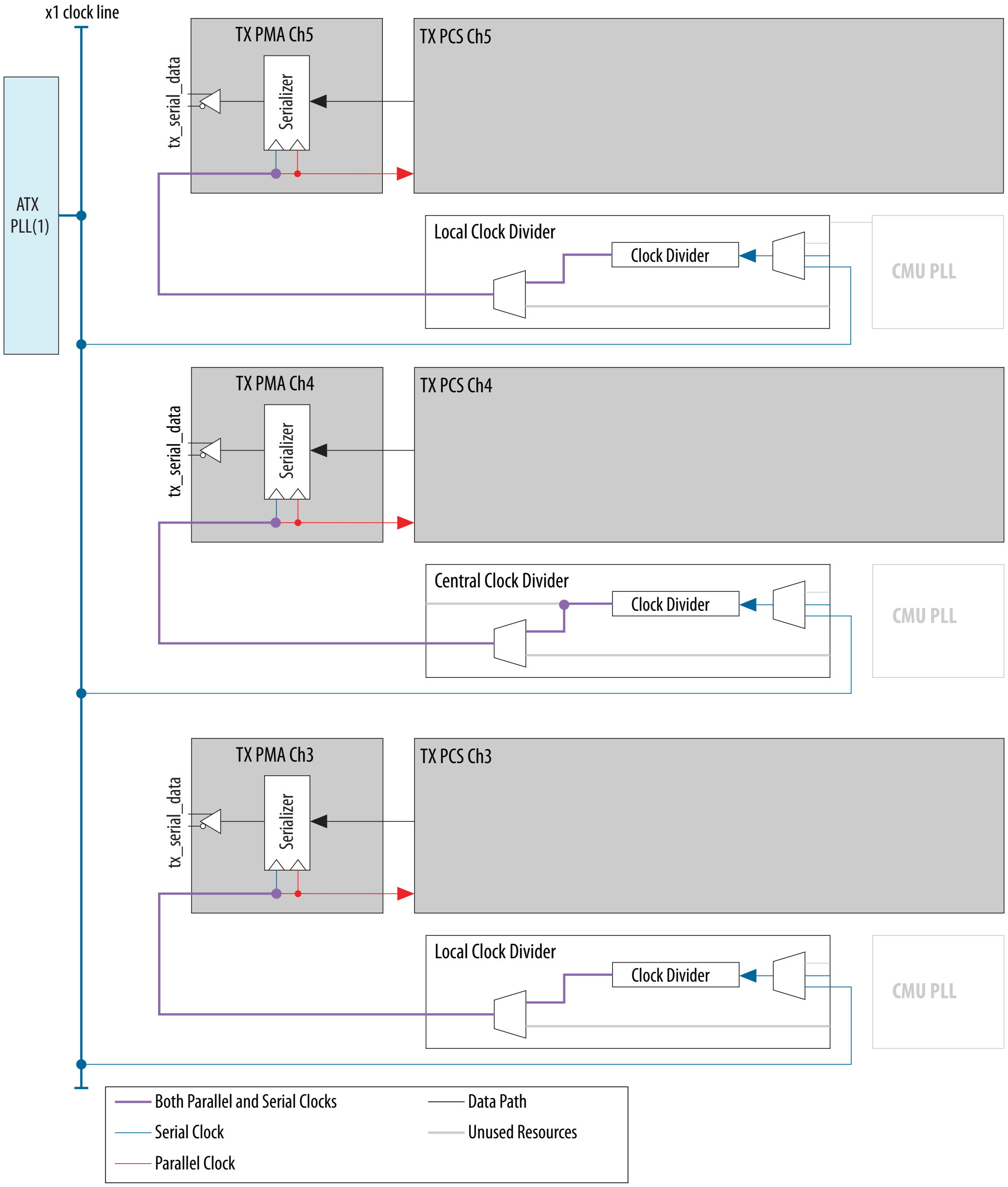


Figure 2-19: Three Non-Bonded Transmitter Channels Driven by ATX PLL using x1 Clock Line Within a Transceiver Bank for GZ Devices.



Note : (1) ATX PLL is available only for Arria V GZ devices.

Figure 2-20: Three Non-Bonded Transmitter Channels Driven by CMU PLL using x6 and xN Clock Lines Across Multiple Transceiver Banks

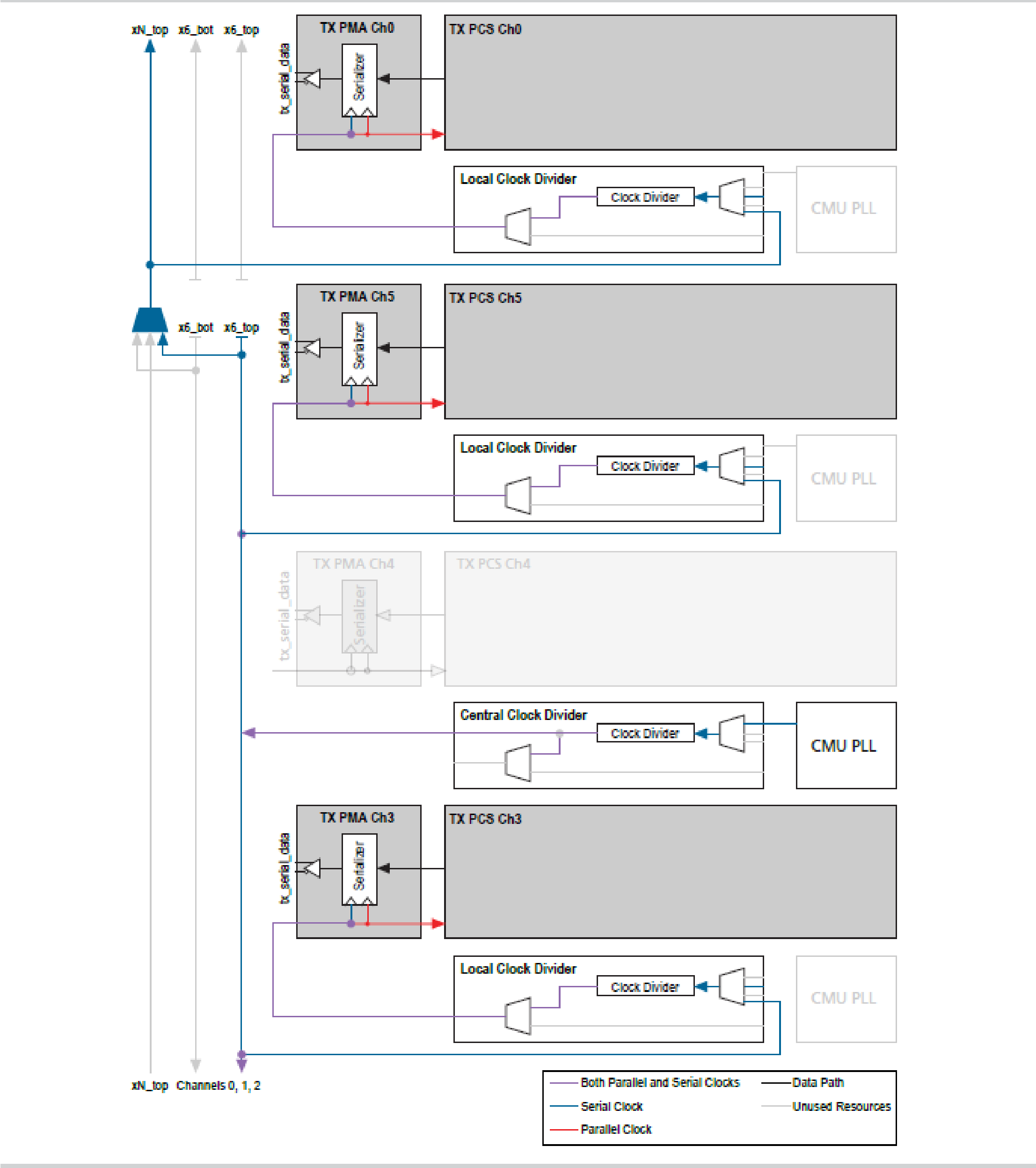
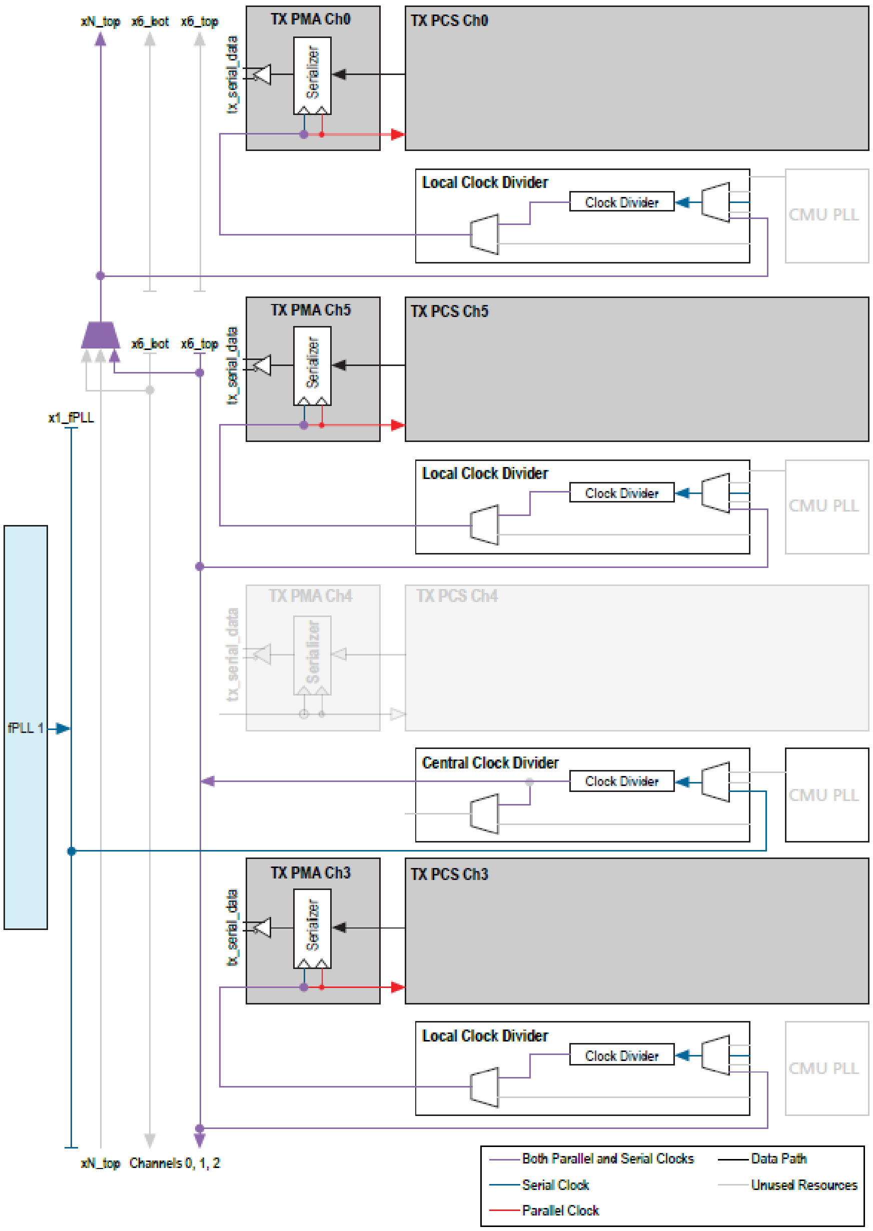


Figure 2-21: Three Non-Bonded Transmitter Channels Driven by fPLL using x6 and xN Clock Line Across Multiple Transceiver Banks



When the fPLL is used to drive more than three non-bonded channels, the channel where the central clock divider resides adjacent to the fPLL cannot be used as a transmitter. The fPLL uses a central clock divider to access the x6 clock network when driving more than three non-bonded channels, so the divider is no longer available to implement a transmitter. For xN non-bonded configurations, the ch 1 or ch 4 transceiver bank where the central clock divider resides cannot be used as a data channel since the parallel clock cannot be generated in this channel.

Bonded Channel Configurations

The channel clock path for bonded configurations is driven by the x6 and xN clock lines.

Table 2-9: Clock Path for Bonded Configurations

The following table describes the clock path for a bonded configuration with ATX PLL, CMU PLL, or fPLL as TX PLL using various clock lines.

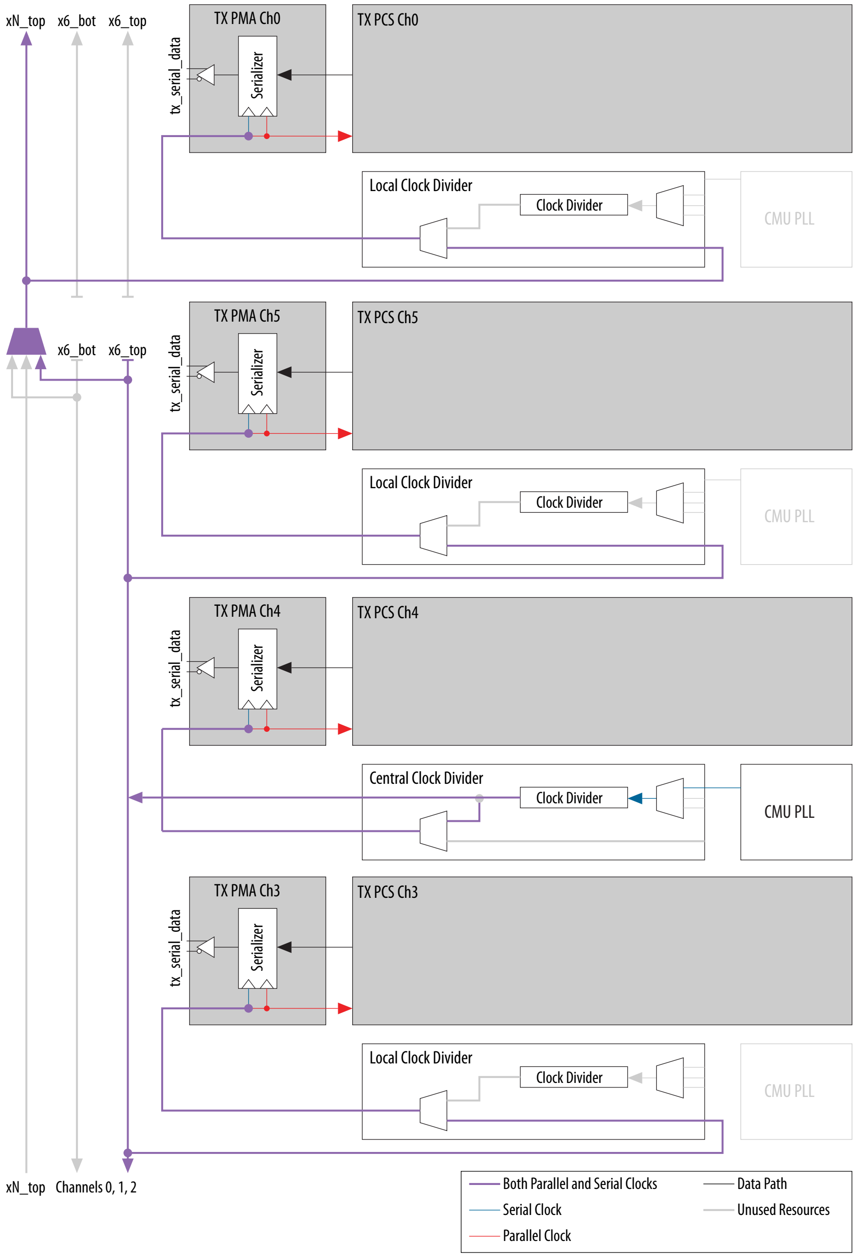
Clock Line	Transmitter PLL	Clock Path
x6, xN	ATX PLL[26]	CMU PLL » central clock divider » x6 » xN » serializer
	CMU	CMU PLL » central clock divider » x6 » xN » serializer [25]
	fPLL	fPLL » x1_fPLL » central clock divider » x6 » serializer [25]
x6 PLL Feedback Compensation [27]	ATX PLL [26]	ATX PLL » central clock divider » x6 » serializer
	CMU	CMU PLL » central clock divider » x6 » serializer

[25] Bonded channels within same bank as the TX PLL are driven by clocks from the x6 clock line, and channels in other banks are driven from the xN clock line.

[26] ATX PLL is available for GZ devices only.

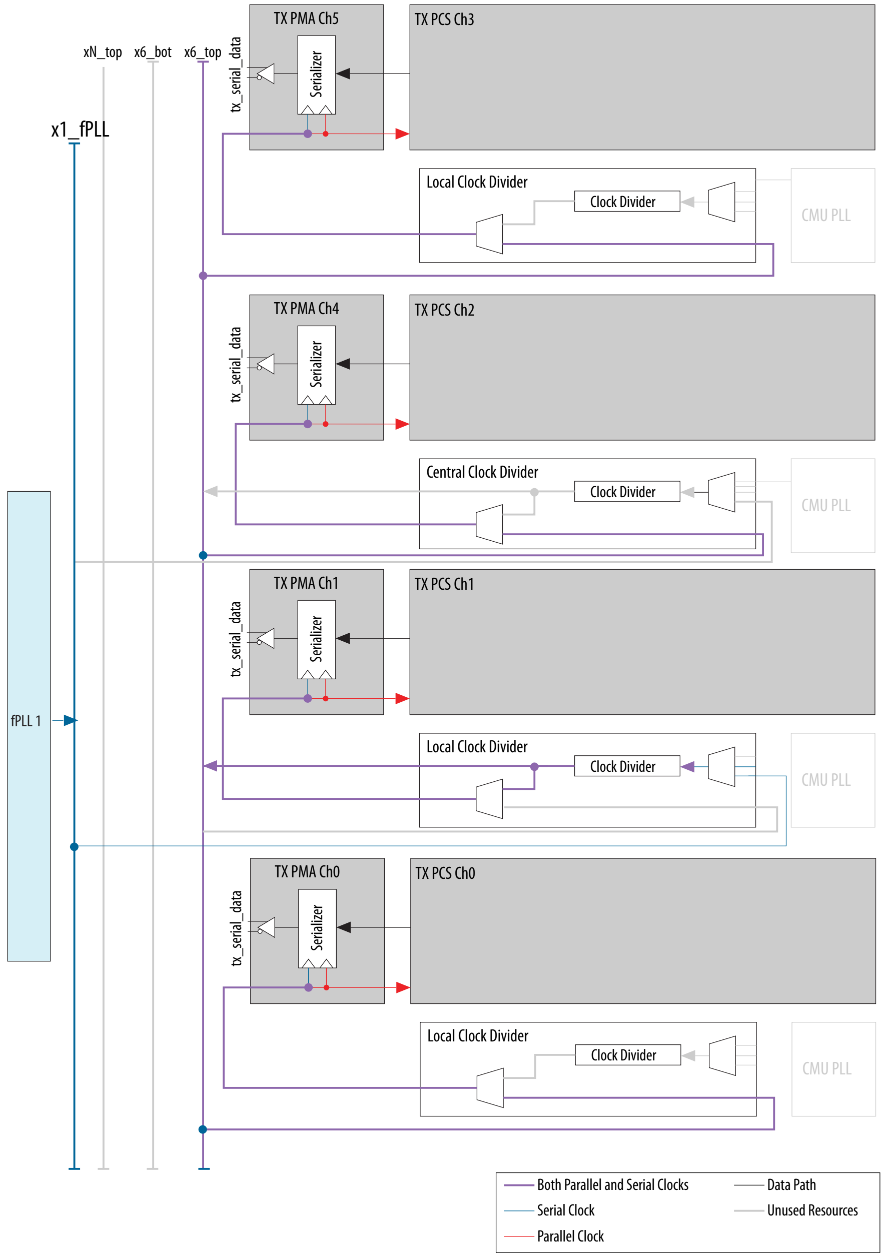
[27] x6 PLL Feedback Compensation is available for GZ devices only.

Figure 2-22: Four Bonded Transmitter Channels Driven by CMU PLL using x6 and xN Clock Lines Across Multiple Transceiver Banks



Note: When channel PLL is configured as a CMU PLL to drive the local clock divider or the central clock divider of its own channel, the channel PLL cannot be used as a CDR. Without a CDR, the channel can be used only as a transmitter.

Figure 2-23: Four Bonded Transmitter Channels Driven by fPLL using x6 Clock Line Within a Transceiver Bank



Note:
- When using the fPLL to drive bonded channels, assign logical channel 0 to the channel where the central clock divider is used for fPLL clocks to access the x6 clock line. Using the preceding figure as an example, assign `tx_serial_data[0]` to the transmitter channel 4 pin location.
- For xN bonded configurations, the channel where the central clock divider resides (ch 1 or ch 4) can be used as a data channel as the parallel clock can be generated in this channel.

Bonded Channel Configurations Using the PLL Feedback Compensation Path for GZ Devices

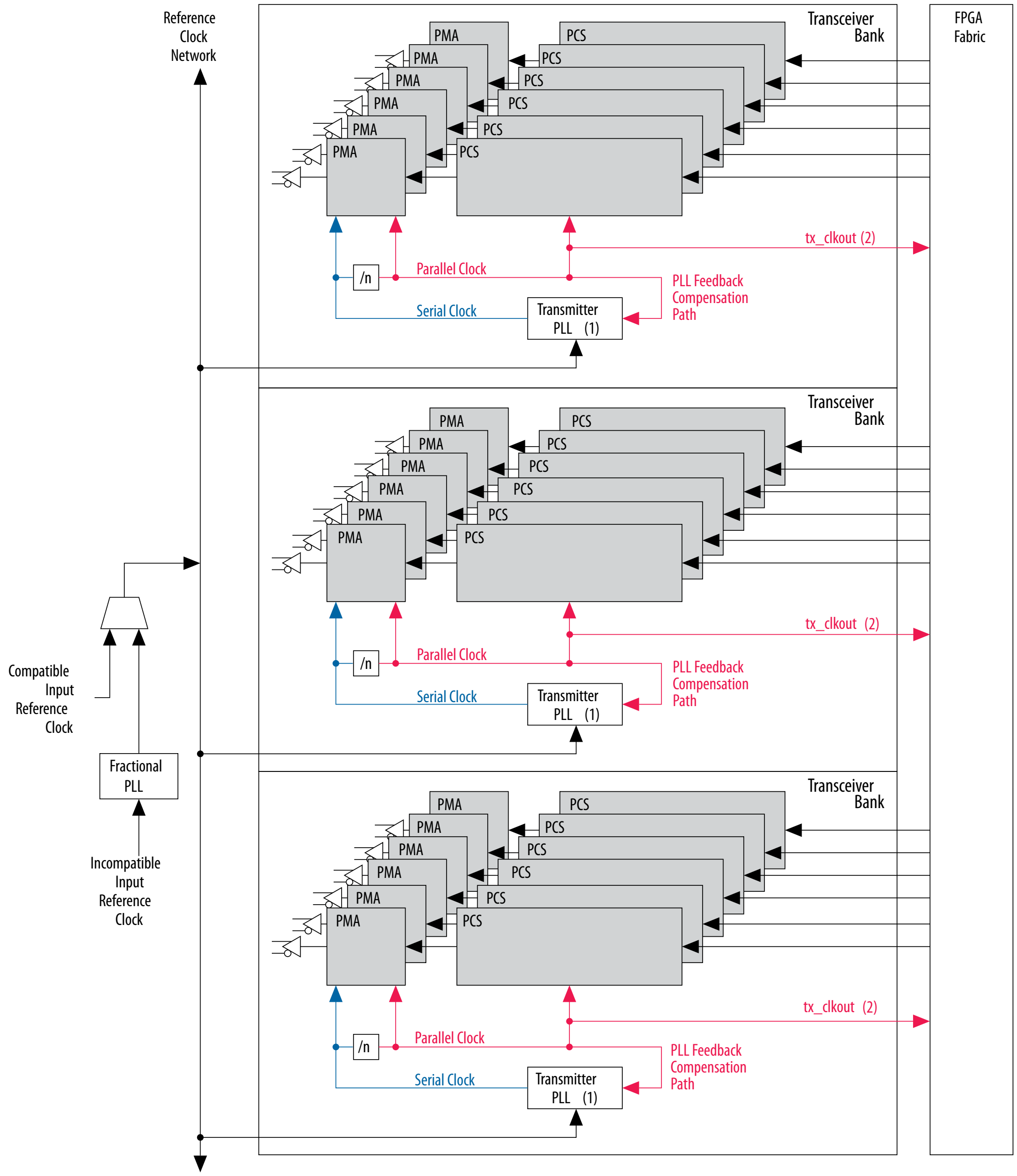
You can bond channels across multiple banks by using the PLL feedback compensation path.

The PLL feedback compensation path loops the parallel clock, which is used by the PCS blocks, back to the transmitter PLL. The PLL feedback compensation path synchronizes the parallel clock used to clock the PCS blocks in all transceiver banks with the `refclk`. You can use the PLL feedback compensation path to reduce channel-to-channel skew, which is introduced by the clock divider in each transceiver bank.

To bond channels using the PLL feedback compensation path, the input reference clock frequency used by the transmitter PLL must be the same as the parallel clock that clocks the PCS of the same channel.

Note: If the input reference clock frequency is not equal to the parallel clock frequency, use a fractional PLL to synthesize an input reference clock with the same frequency as the parallel clock.

Figure 2-24: Three Transceiver Bank Channels Bonded Using the PLL Feedback Compensation Path for GZ Devices



Notes: (1) The transmitter PLL can be an ATX PLL, or a CMU PLL. You can have up to six channels per bank with an ATX PLL and four channels per bank with a CMU PLL.
(2) tx_clkout from any of the banks can be used with the FPGA fabric-transceiver interface for all the bonded channels.

Note:
- Every transceiver bank with a bonded channel configured using the PLL feedback compensation path consumes a transmit PLL.
- fPLL does not support PLL feedback compensation when used as a TX PLL.

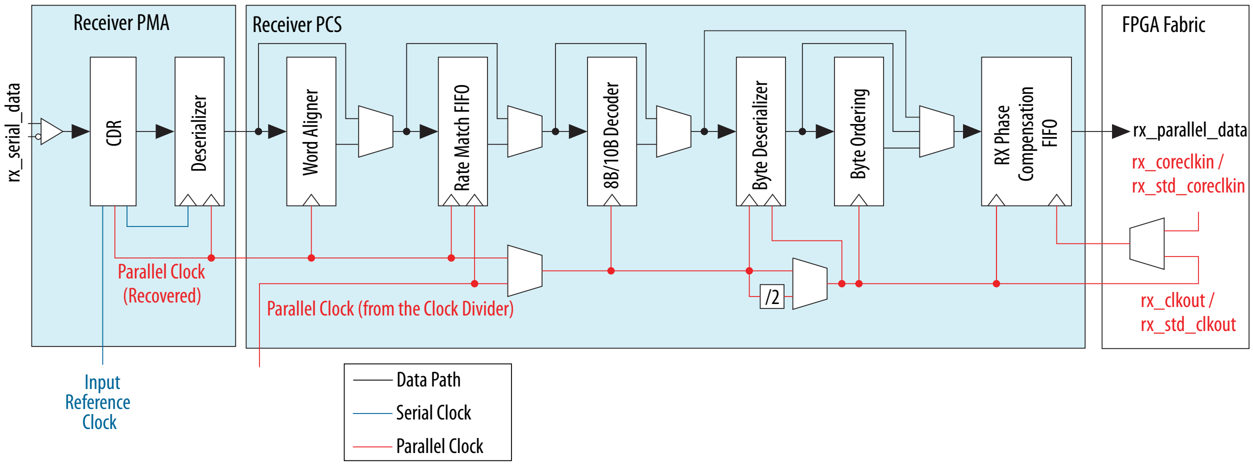
Transceiver Channel Placement Guidelines for fPLL in Transmit PLL Bonded Configuration (Except GZ Devices)

The fPLL as transmit PLL, when configured in bonded configuration, has placement restrictions. All channels need to be placed within one transceiver bank. A link cannot span across two banks. The channel placement must also be contiguous.

Receiver Clocking

Receiver clocking refers to the clocking architecture internal to the receiver channel of a transceiver.

Figure 2-25: Clocking Architecture for Receiver PCS and PMA Configuration



The CDR in the PMA of each channel recovers the serial clock from the incoming data and generates the parallel clock (recovered) by dividing the serial clock (recovered). The deserializer uses both clocks. The receiver PCS can use the following clocks depending on the configuration of the receiver channel:

- Parallel clock (recovered) from the CDR in the PMA
- Parallel clock from the clock divider that is used by the channel's transmitter PCS

[28] Available for Arria V GZ devices only.

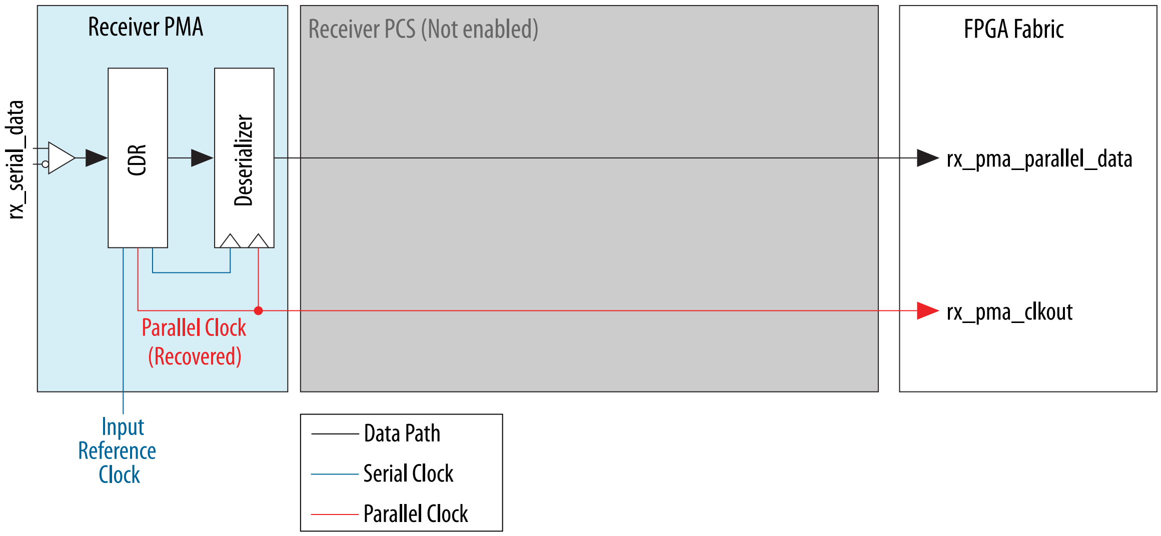
[29] For more information about loopback mode, refer to the *Transceiver Loopback Support* chapter in Arria V Devices.

Table 2-10: Clock Sources for All Receiver PCS Blocks

PCS	Block	Side	Clock Source
Standard	Word aligner	-	Parallel clock (recovered)
	Rate match FIFO	Write	Parallel clock (recovered)
		Read	Parallel clock from the clock divider
	8B/10B decoder	-	• Rate match FIFO is not used-Parallel clock (recovered) • Rate match FIFO is used-Parallel clock from the clock divider
	Byte deserializer	Write	• Rate match FIFO is not used-Parallel clock (recovered) • Rate match FIFO is used-Parallel clock from the clock divider
		Read	Divided down version of the write side clock depending on the deserialization factor of 1 or 2, also called the parallel clock (divided)
	Byte ordering	-	Parallel clock (divided)
	Receiver (RX) phase compensation FIFO	Write	Parallel clock (divided). This clock is also forwarded to the FPGA fabric.
		Read	Clock sourced from the FPGA fabric
10G[28]	All other PCS blocks		• Regular mode: parallel clock (recovered) • Loopback mode: parallel clock from the clock divider.[29]

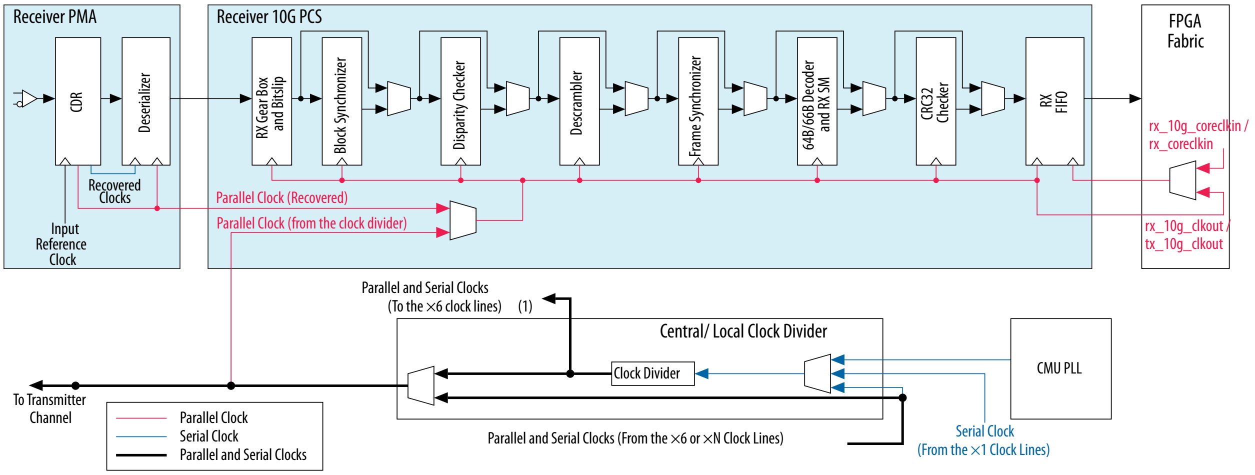
Figure 2-26: Clocking Architecture for Receiver PMA Only Configuration

The parallel recovered clock from the CDR and deserializer is forwarded to the FPGA fabric to interface FPGA fabric with the receiver PMA directly, bypassing the PCS blocks.



Clocking Architecture for Receiver 10G PCS and the Receiver PMA for GZ Devices.

Figure 2-27: Clocking Architecture for 10G PCS and the Receiver PMA for GZ Devices


Receiver PMA
CDR
Deserializer
Recovered Clocks
Input Reference Clock
Receiver 10G PCS
RX Gear Box and Bitslip
Block Synchronizer
Disparity Checker
Descrambler
Frame Synchronizer
64B/66B Decoder and RX SM
CRC32 Checker
RX FIFO
Parallel Clock (Recovered)
Parallel Clock (from the clock divider)
FPGA Fabric
rx_10g_coreclkin / rx_coreclkin
rx_10g_clkout / tx_10g_clkout
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Central/ Local Clock Divider
Clock Divider
CMU PLL
To Transmitter Channel
Parallel Clock
Serial Clock
Parallel and Serial Clocks
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines)
Serial Clock (From the ×1 Clock Lines)

Note: (1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

Related Information

Transceiver Loopback Support in Arria V Devices

Receiver Non-Bonded Channel Configurations

The receiver clocking in non-bonded mode varies, depending on whether the rate match FIFO is enabled. When the rate match FIFO is not enabled, the receiver PCS in every channel uses the parallel recovered clock. When the rate match FIFO is enabled, the receiver PCS in every channel uses both the parallel recovered clock and parallel clock from the clock divider.

For Arria V GZ devices, in non-bonded configurations the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2-28: Three Non-Bonded Receiver Channels without Rate Match FIFO Enabled

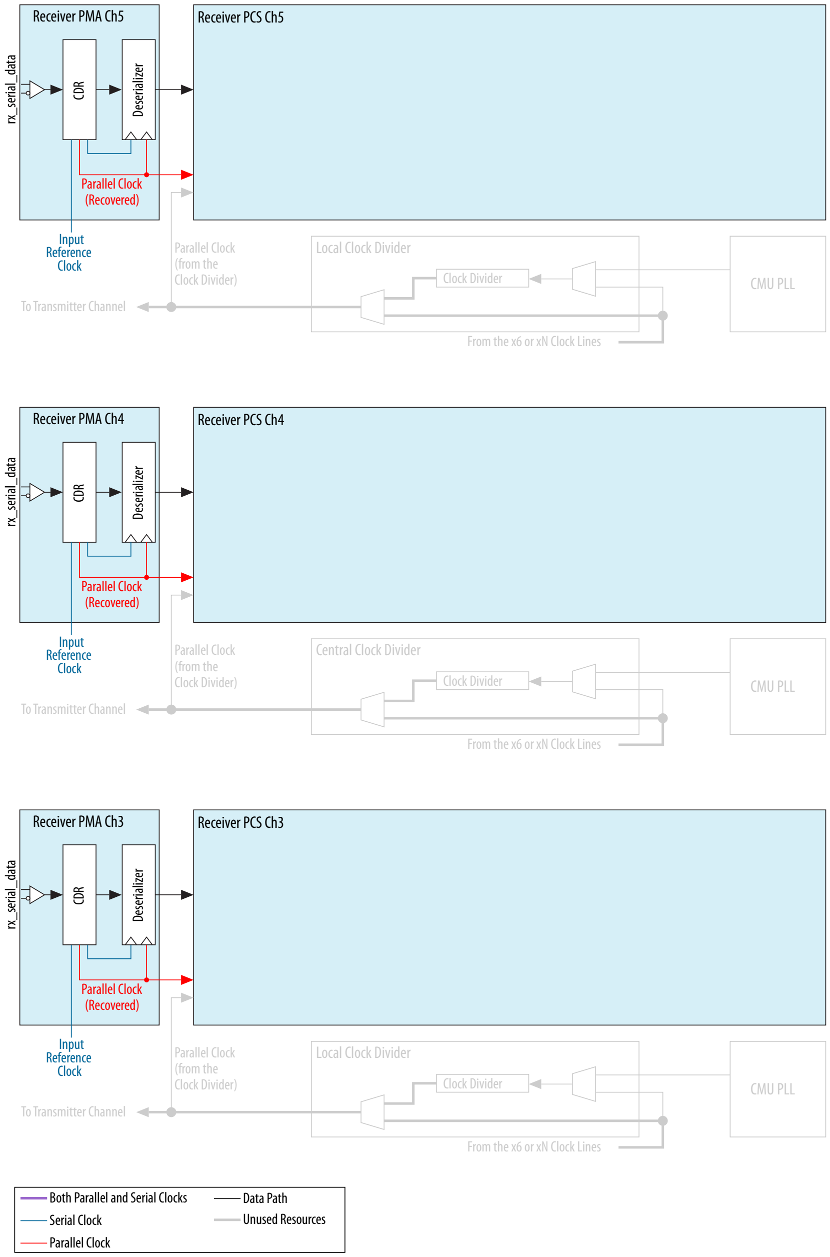
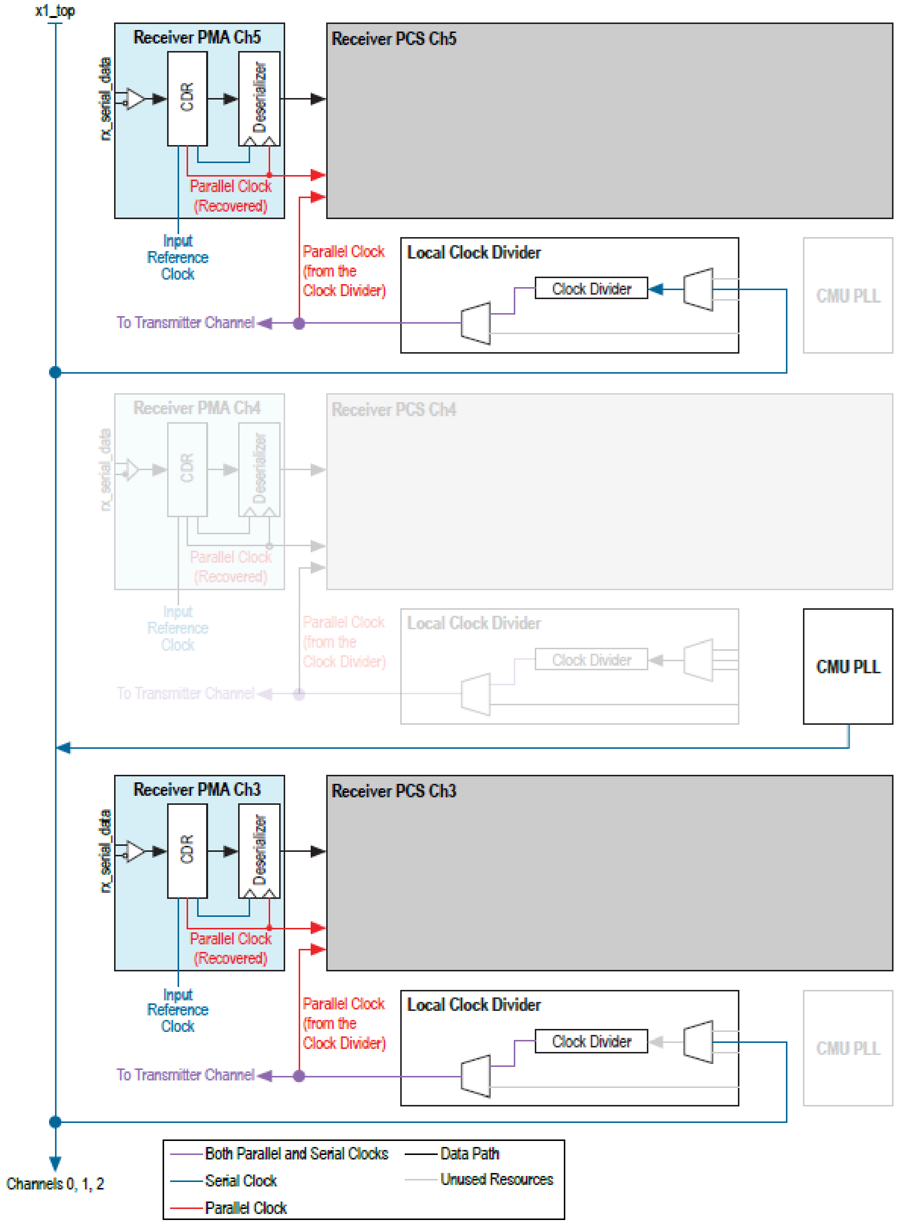


Figure 2-29: Two Non-Bonded Receiver Channels with Rate Match FIFO Enabled



Receiver Bonded Channel Configurations

Receiver channels can only be bonded in configurations where rate match FIFOs are enabled. When bonded, the receiver PCS requires the parallel clock (recovered) and, the parallel clock from the central clock divider in channel 1 or 4.

For Arria V GZ devices, in bonded configurations the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2-30: Five Bonded Receiver Channels with Rate Match FIFO Enabled

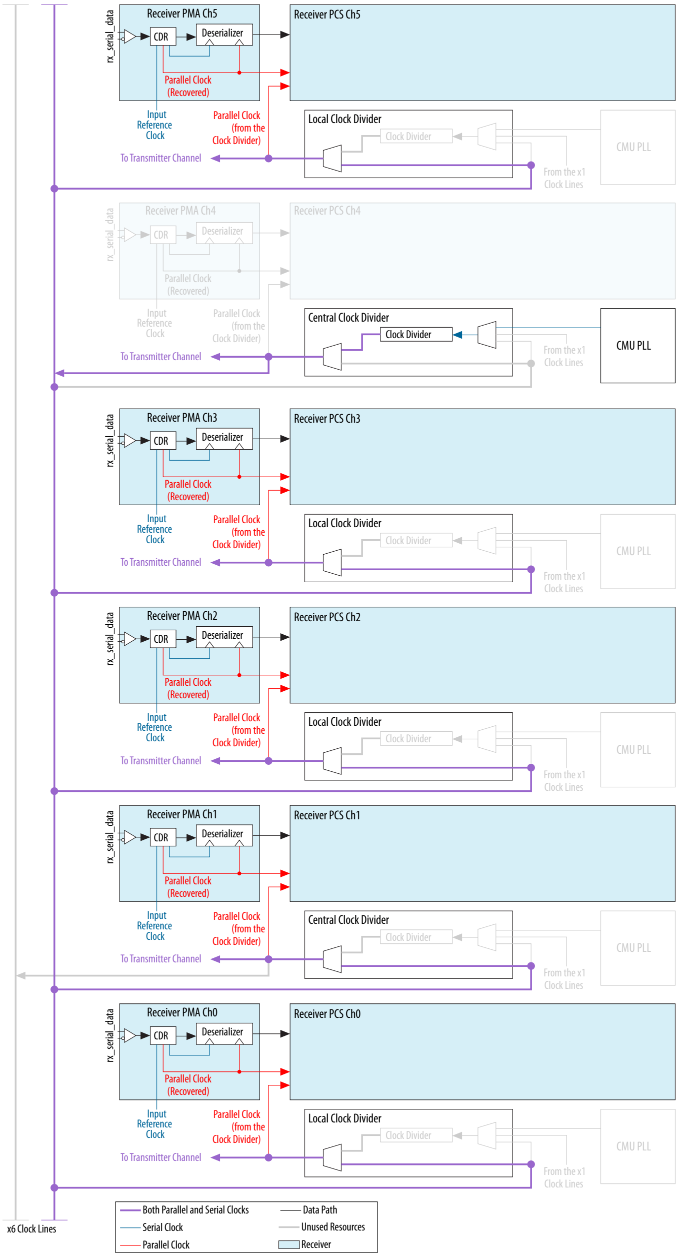


Figure 2-31: Six Bonded Receiver Channels with Rate Match FIFO Enabled Using Fractional PLL

All six channels in the transceiver bank are in a bonded configuration. This configuration is possible because the fractional PLL is used as a transmit PLL instead of a channel PLL in the transceiver bank. Using the fractional PLL enables you to configure the channel PLLs of both channels 1 and 4 as CDRs to perform receiver operations.

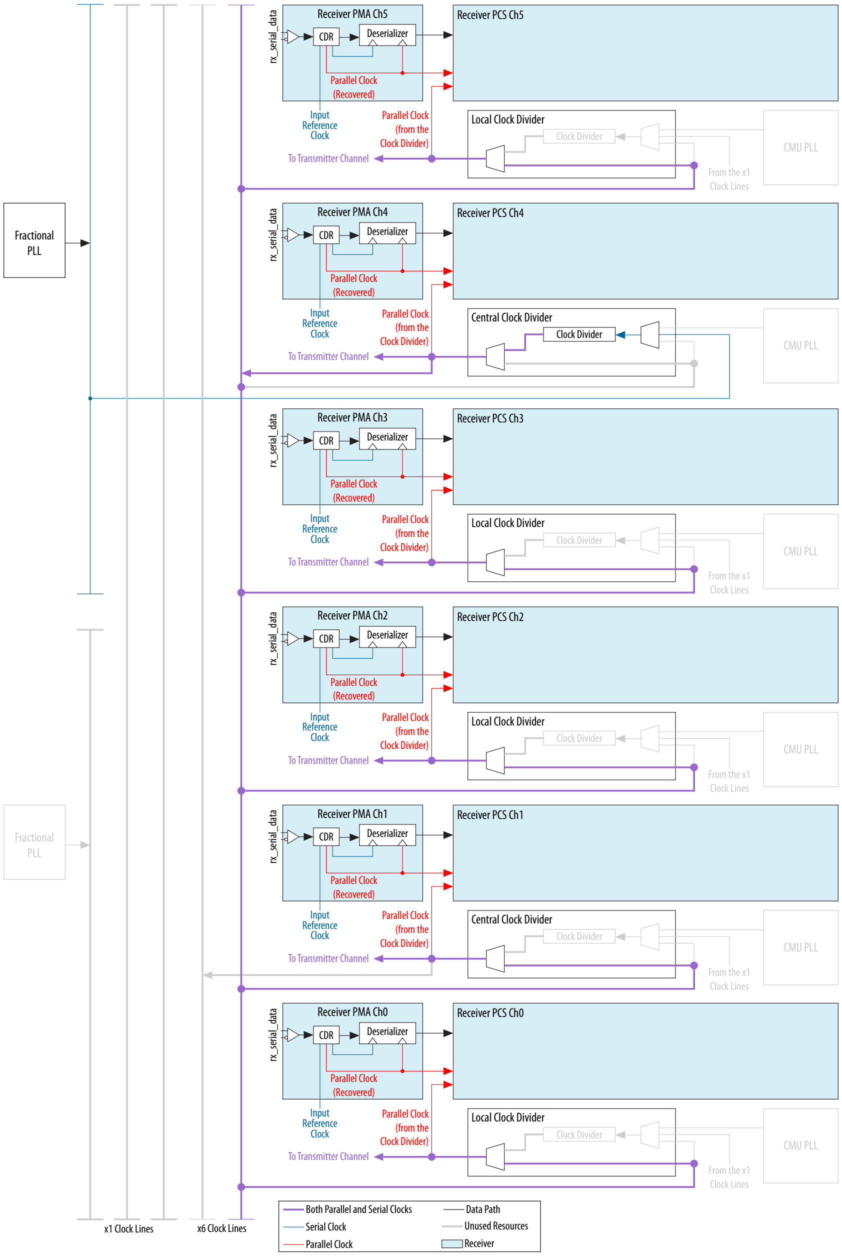
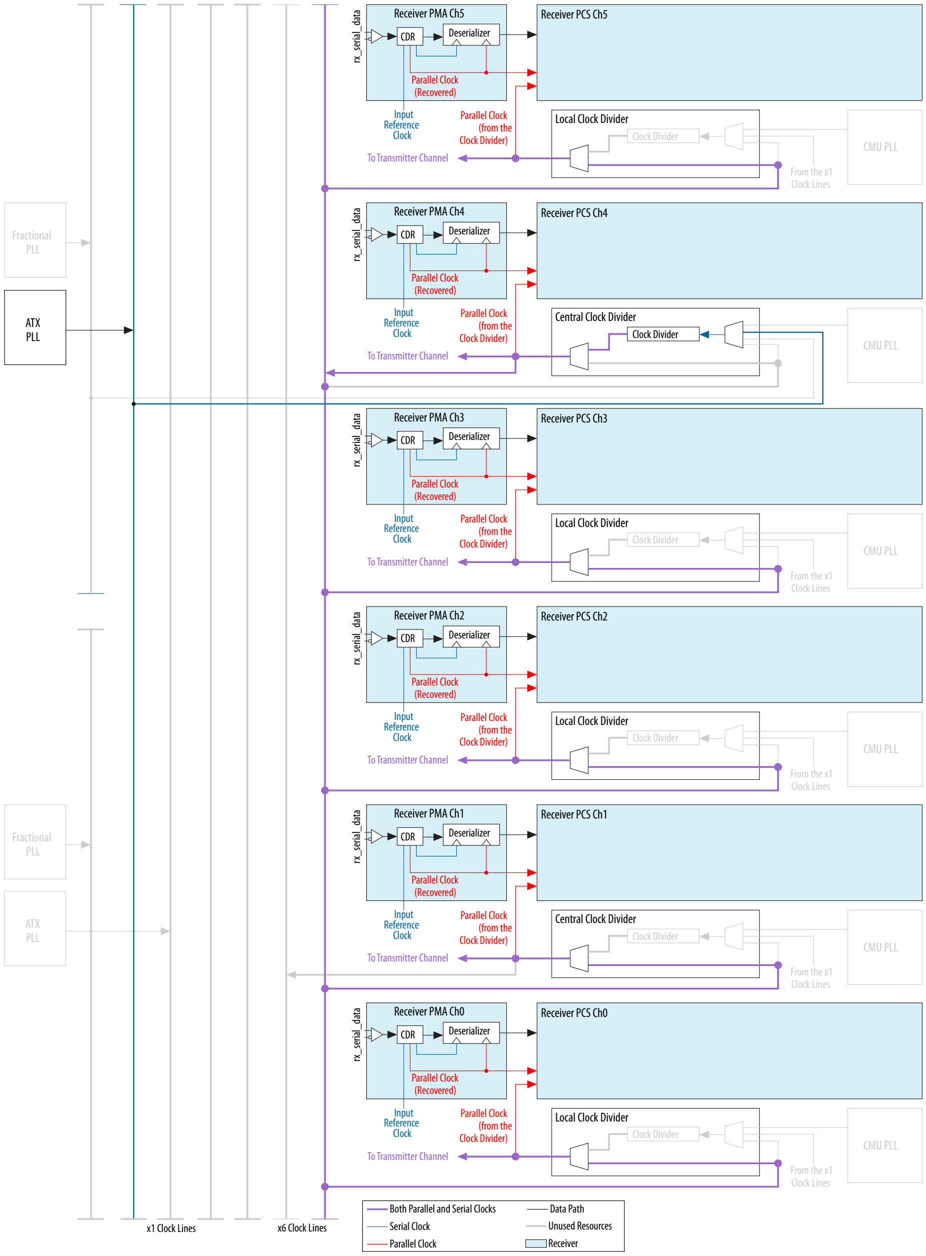


Figure 2-32: Six Channels Configured in Bonded Configuration Using ATX PLL for GZ Devices

All six channels in the transceiver bank in a bonded configuration. Six channel bonding is possible because the ATX PLL is used as a transmitter PLL instead of a channel PLL in the transceiver bank. Using the ATX PLL or fractional PLL allows you to use the channel PLLs of both channels 1 and 4 as CDRs to perform receiver operations.



Related Information

- Transceiver Protocol Configurations in Arria V Devices
- Transceiver Custom Configurations in Arria V Devices

- **Transceiver Configurations in Arria V GZ Devices**
  Information about the clocking scheme used in different configurations.

FPGA Fabric–Transceiver Interface Clocking

This section describes the clocking options available when the transceiver interfaces with the FPGA fabric.

The FPGA fabric–transceiver interface clocks can be subdivided into the following three categories:

- Input reference clocks—Can be an FPGA fabric–transceiver interface clock. This may occur when the FPGA fabric-transceiver interface clock is forwarded to the FPGA fabric, where it can then clock logic.

  Note: The input reference clock can only be routed into the FPGA fabric if a transceiver is also instantiated.
- Transceiver datapath interface clocks—Used to transfer data, control, and status signals between the FPGA fabric and the transceiver channels. The transceiver channel forwards the `tx_clkout` signal to the FPGA fabric to clock the data and control signals into the transmitter. The transceiver channel also forwards the recovered `rx_clkout` clock (in configurations without the rate matcher) or the `tx_clkout` clock (in configurations with the rate matcher) to the FPGA fabric to clock the data and status signals from the receiver into the FPGA fabric.
- Other transceiver clocks—The following transceiver clocks form a part of the FPGA fabric–transceiver interface clocks:
  - `phy_mgmt_clk`—Avalon®-MM interface clock used for controlling the transceivers, dynamic reconfiguration, and calibration
  - `fixed_clk`—the 125 MHz fixed-rate clock used in the PCIe (PIPE) receiver detect circuitry

Table 2-11: FPGA Fabric–Transceiver Interface Clocks

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization
tx_pll_refclk, rx_cdr_refclk	Input reference clock used for clocking logic in the FPGA fabric	Transceiver-to-FPGA fabric	GCLK, RCLK, PCLK
tx_clkout, tx_pma_clkout	Clock forwarded by the transceiver for clocking the transceiver datapath interface		
rx_clkout, rx_pma_clkout	Clock forwarded by the receiver for clocking the receiver datapath interface		
tx_coreclkin	User-selected clock for clocking the transmitter datapath interface	FPGA fabric-to-transceiver	
rx_coreclkin	User-selected clock for clocking the receiver datapath interface		
fixed_clk	PCIe receiver detect clock		
phy_mgmt_clk[30]	Avalon-MM interface management clock		

Note:
- For Arria V GZ devices, you can forward the pll_refclk, tx_clkout, and rx_clkout clocks to a fractional PLL so that the fractional PLL can synthesize a clock for the FPGA logic. A second fractional PLL can be reached by periphery clocks, depending on your device and channel placement, and may require using a RGCLK or GCLK.
- For more information about the GCLK, RCLK, and PCLK resources available in each device, refer to the Clock Networks and PLLs in Arria V Devices chapter

Table 2-12: Configuration Specific Port Names for tx_clkout and rx_clkout

Configuration	Port Name for tx_clkout	Port Name for rx_clkout
Custom	tx_clkout	rx_clkout

[30] The phy_mgmt_clk is a free-running clock that is not derived from the transceiver blocks.

Configuration	Port Name for tx_clkout	Port Name for rx_clkout
Native - 10G PCS[31]	tx_10g_clkout	rx_10g_clkout
Native - Standard PCS	tx_std_clkout	rx_std_clkout
Native - PMA Direct	tx_pma_clkout	rx_pma_clkout
Interlaken [31]	tx_clkout	rx_clkout
Low Latency	tx_clkout	rx_clkout
PCIe	pipe_pclk	pipe_pclk
XAUI	xgmii_tx_clk	xgmii_rx_clk

Related Information

Clock Networks and PLLs in Arria V Devices chapter

Transceiver Datapath Interface Clocking

There are two types of design considerations for clock optimization when interfacing the transceiver datapath to the FPGA fabric:

- PCS with FIFO in phase compensation mode – share clock network for identical channels
- PCS with FIFO in registered mode or PMA direct mode – refer to *AN580: Achieving Timing Closure in Basic (PMA Direct) Functional Mode for additional timing closure techniques between transceiver and FPGA fabric*

Note: For Arria V (GX, GT, ST and SX) devices, the PMA clock for channel 1 and channel 2 of GXB_L0 and GXB_R0 cannot be routed out of the FPGA fabric.

Related Information

AN580: Achieving Timing Closure in Basic (PMA Direct) Functional Mode

Transmitter Datapath Interface Clocking

In 6-Gbps transceivers, the write side of the transmitter phase compensation FIFO makes up the transmitter datapath interface.

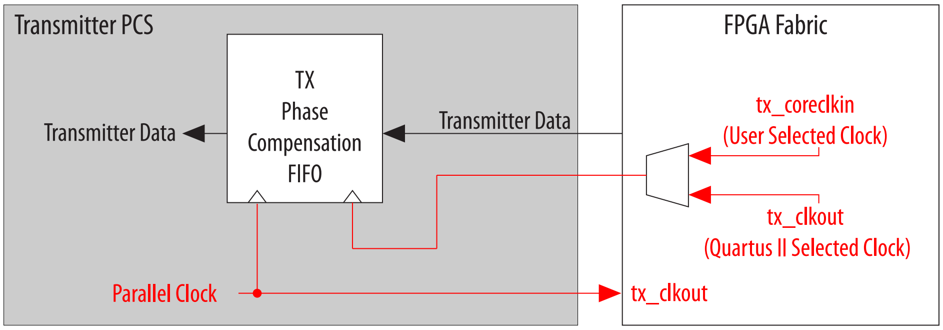
For the 6-Gbps transceivers, the write side of the transmitter phase compensation FIFO makes up the transmitter datapath interface. This interface is clocked with the transmitter datapath interface clock.

The following figure shows the 6-Gbps transmitter datapath interface clocking. The transmitter PCS forwards the following clocks to the FPGA fabric:

- tx_clkout—for each transmitter channel in a non-bonded configuration
- tx_clkout[0]—for all transmitter channels in a bonded configuration

[31] Available for Arria V GZ devices only.

Figure 2-33: Transmitter Datapath Interface Clocking for 6-Gbps Transceivers

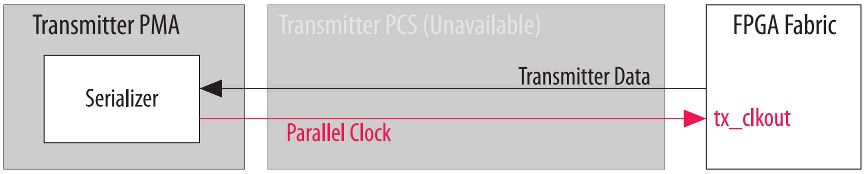


All configurations that use the PCS channel must have a 0 parts per million (ppm) difference between write and read clocks of the transmitter phase compensation FIFO.

For the 10-Gbps transceivers Arria V GT/ST devices, there are no PCS blocks. The only transmit datapath available is a direct connection from the FPGA fabric to the serializer of the transmitter PMA.

The following figure shows the 10-Gbps transmitter datapath interface clocking. For each transmitter channel in a non-bonded configuration, the FPGA fabric forwards the following `tx_clkout` clock to the transmitter PMA.

Figure 2-34: Transmitter Datapath Interface Clocking for 10-Gbps Transceivers (for Arria V GT/ST Devices)



You can clock the transmitter datapath interface by one of the following options:

- The Quartus II-selected transmitter datapath interface clock
- The user-selected transmitter datapath interface clock

Note: To reduce GCLK, RCLK, and PCLK resource utilization in your design, you can select the user-selection option to share the transceiver datapath interface clocks.

Related Information

- **Transceiver Custom Configurations in Arria V Devices.**
- **Transceiver Protocol Configurations in Arria V Devices.**
  Information about interface clocking for each configuration.

Quartus II-Software Selected Transmitter Datapath Interface Clock

The Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the transmitter datapath interface.

Figure 2-35: 6-Gbps Transmitter Datapath Interface Clocking for Non-Bonded Channels

The figure shows the transmitter datapath interface of two 6 Gbps transceiver non-bonded channels clocked by their respective transmitter PCS clocks which are forwarded to the FPGA fabric.

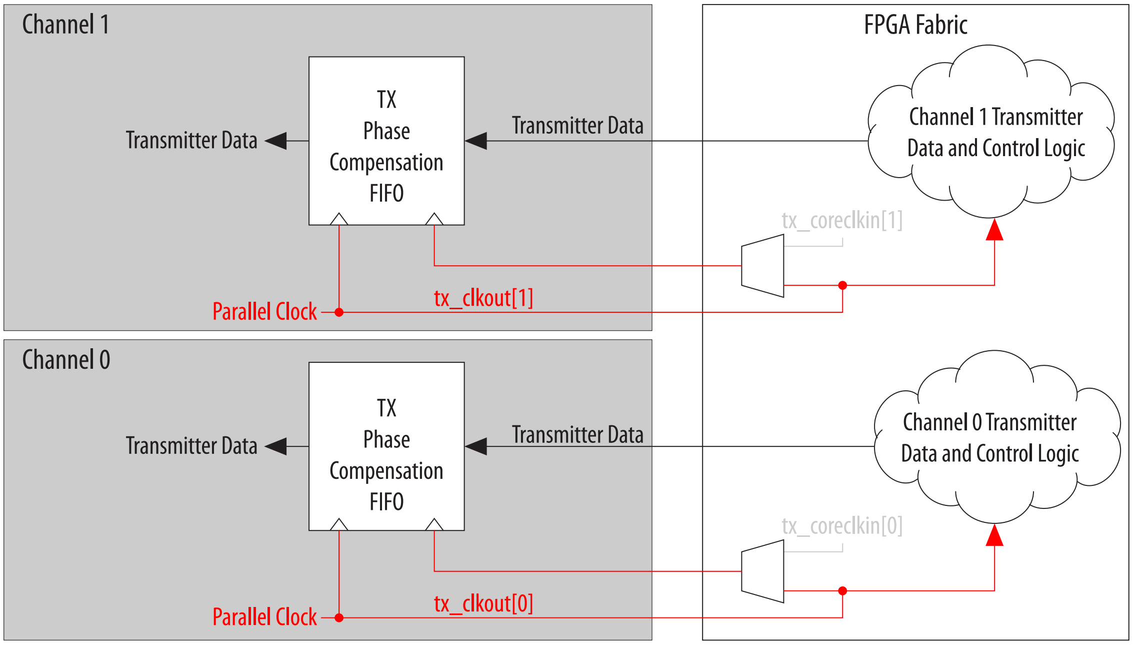


Figure 2-36: Arria V GT/ST 10-Gbps Transmitter Datapath Interface Clocking for Non-Bonded Channels

The figure shows the Arria V GT/ST transmitter datapath interface of two 10-Gbps transceiver non-bonded channels clocked by their respective transmitter PMA clocks, which are forwarded to the FPGA fabric.

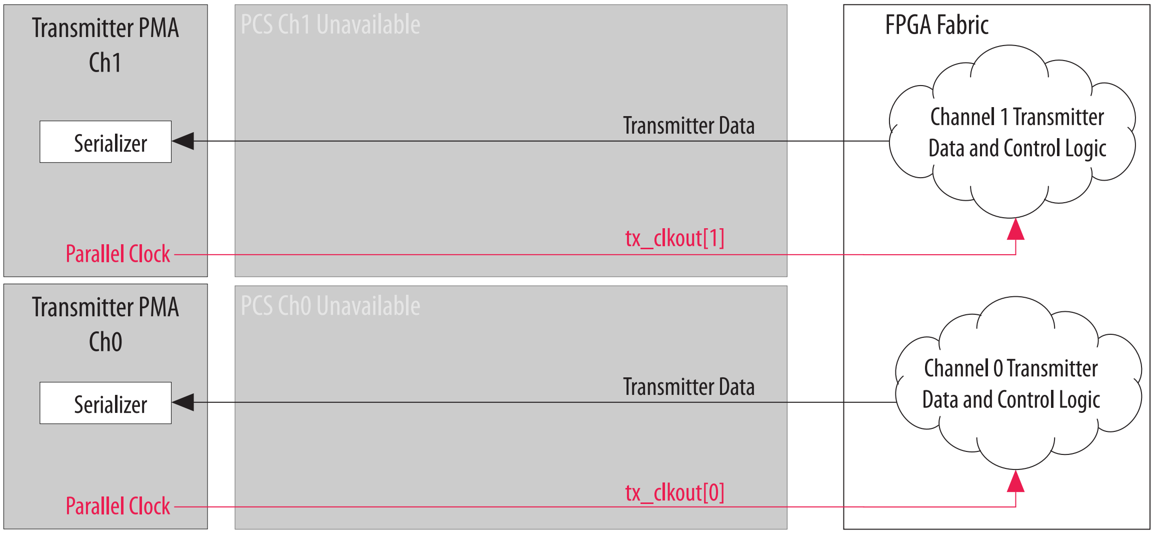
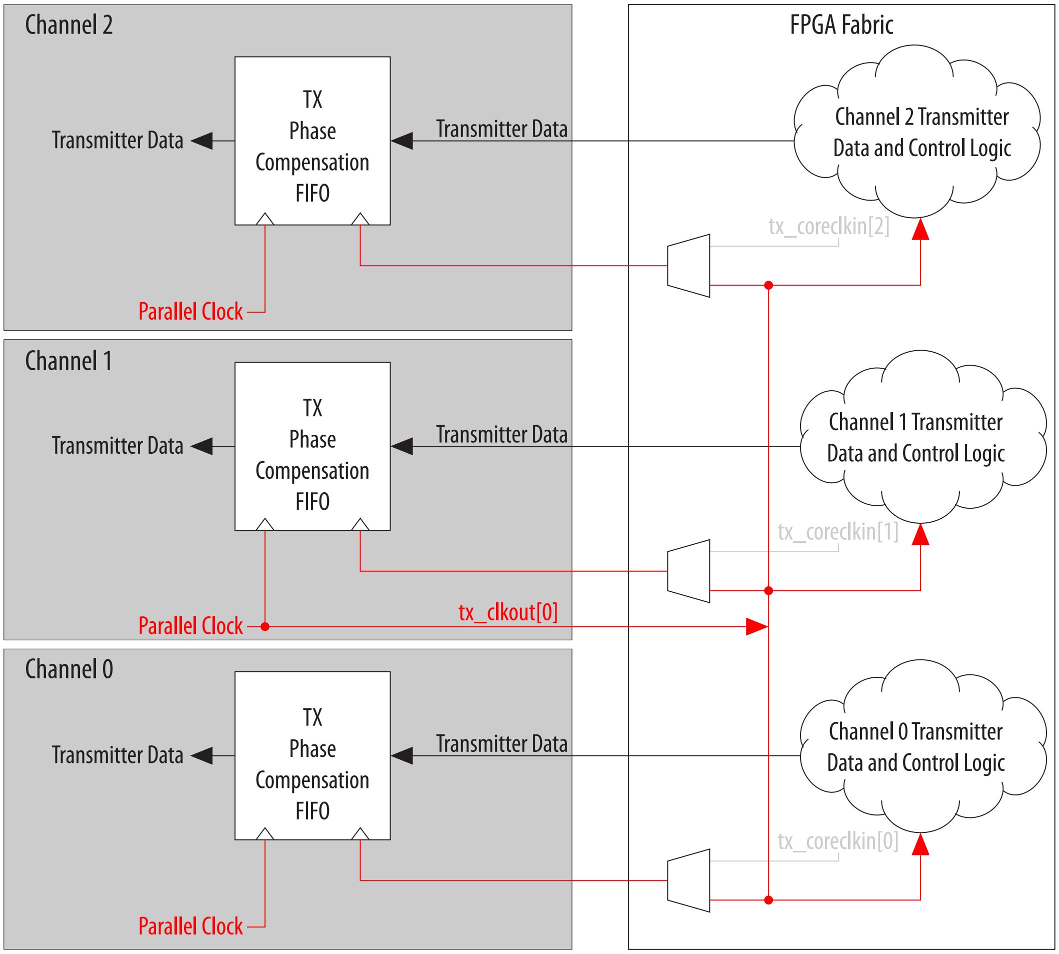


Figure 2-37: 6-Gbps Transmitter Datapath Interface Clocking for Three Bonded Channels

The following figure shows the 6-Gbps transmitter datapath interface of three bonded channels clocked by the `tx_clkout[0]` clock. The `tx_clkout[0]` clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.



Selecting a Transmitter Datapath Interface Clock

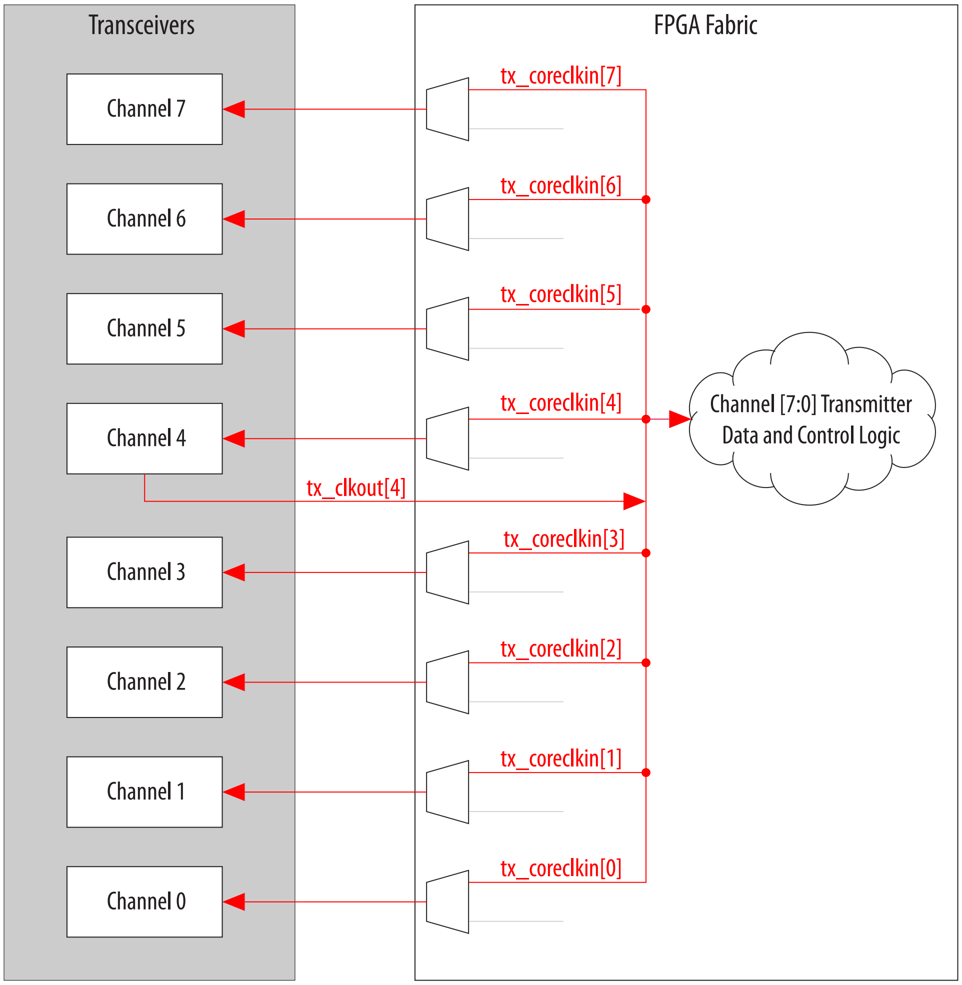
Multiple non-bonded transmitter channels use a large portion of GCLK, RCLK, and PCLK resources. Selecting a common clock driver for the transmitter datapath interface of all identical transmitter channels saves clock resources.

Multiple transmitter channels that are non-bonded lead to high utilization of GCLK, RCLK, and PCLK resources (one clock resource per channel). You can significantly reduce GCLK, RCLK, and PCLK resource use for transmitter datapath clocks if the transmitter channels are identical.

Note: Identical transmitter channels have the same input reference clock source, transmit PLL configuration, transmitter PMA, and PCS configuration, but may have different analog settings, such as transmitter voltage output differential (VOD), transmitter common-mode voltage (VCM), or pre-emphasis.

To achieve the clock resource savings, select a common clock driver for the transmitter datapath interface of all identical transmitter channels. The following figure shows eight identical channels clocked by a single clock (`tx_clkout` of channel 4).

Figure 2-38: Eight Identical Channels with a Single User-Selected Transmitter Interface Clock



To clock eight identical channels with a single clock, perform these steps:

1. Instantiate the `tx_coreclkin` port for all the identical transmitter channels (`tx_coreclkin[7:0]`).
2. Connect `tx_clkout[4]` to the `tx_coreclkin[7:0]` ports.
3. Connect `tx_clkout[4]` to the transmitter data and control logic for all eight channels.

Note: Resetting or powering down channel 4 causes a loss of the clock for all eight channels.

The common clock must have a 0 ppm difference for the read side of the transmitter phase compensation FIFO of all the identical channels. A frequency difference causes the FIFO to under run or overflow, depending on whether the common clock is slower or faster, respectively.

You can drive the 0 ppm common clock by one of the following sources:

- `tx_clkout` of any channel in non-bonded channel configurations
- `tx_clkout[0]` in bonded channel configurations
- Dedicated refclk pins

Note: The Quartus II software does not allow gated clocks or clocks that are generated in the FPGA logic to drive the `tx_coreclkin` ports.

You must ensure a 0 ppm difference. The Quartus II software is unable to ensure a 0 ppm difference because it allows you to use external pins, such as dedicated `refclk` pins.

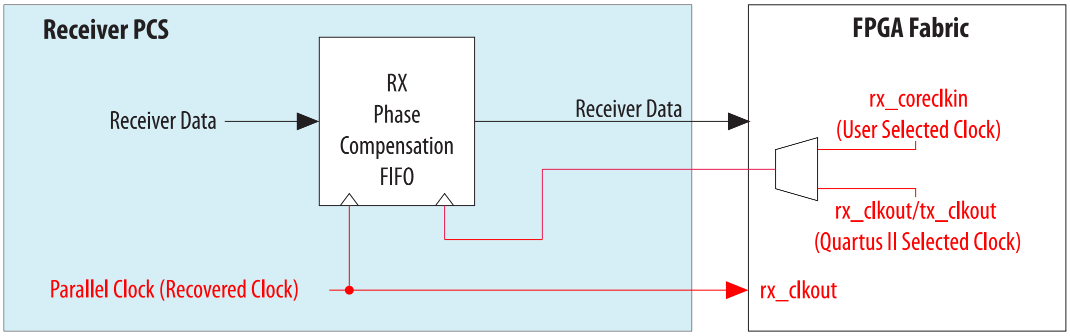
Receiver Datapath Interface Clock

The 6-Gbps receiver datapath interface is comprised of the read side RX phase compensation FIFO.

The read side of the RX phase compensation FIFO makes up the 6-Gbps receiver datapath interface. The receiver datapath interface clock clocks this interface. The receiver PCS forwards the following clocks to the FPGA fabric:

- `rx_clkout`—for each receiver channel in a non-bonded configuration when you do not use a rate matcher
- `tx_clkout`—for each receiver channel in a non-bonded configuration when you use a rate matcher
- `single rx_clkout[0]`—for all receiver channels in a bonded configuration

Figure 2-39: 6-Gbps Receiver Datapath Interface Clocking

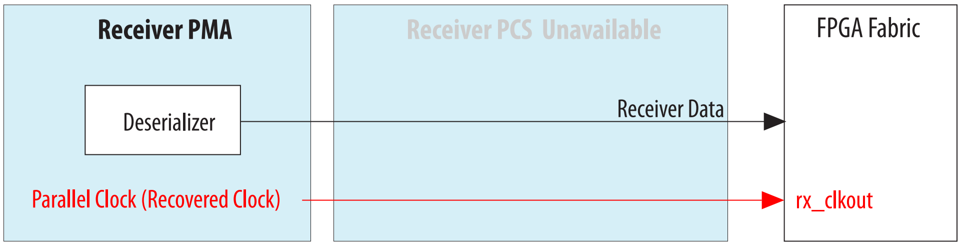


All configurations that use the PCS channel must have a 0 ppm difference between the receiver datapath interface clock and the read side clock of the RX phase compensation FIFO.

For the 10-Gbps transceivers in Arria V GT/ST devices, there are no PCS blocks. The only receiver datapath available is a direct connection from the receiver PMA deserializer to the FPGA fabric.

For each receiver channel in a non-bonded configuration, the receiver PMA forwards the `rx_clkout` clock to the FPGA fabric.

Figure 2-40: Arria V GT/ST 10-Gbps Receiver Datapath Interface Clocking



Note: For more information about interface clocking for each configuration, refer to the Transceiver Custom Configuration in Arria V Devices and Transceiver Protocol Configurations in Arria V Devices chapters.

You can clock the receiver datapath interface by one of the following options:

- The Quartus II-selected receiver datapath interface clock
- The user-selected receiver datapath interface clock

Note: To reduce GCLK, RCLK, and PCLK resource utilization in your design, you can select the user-selection option to share the transceiver datapath interface clocks.

Related Information

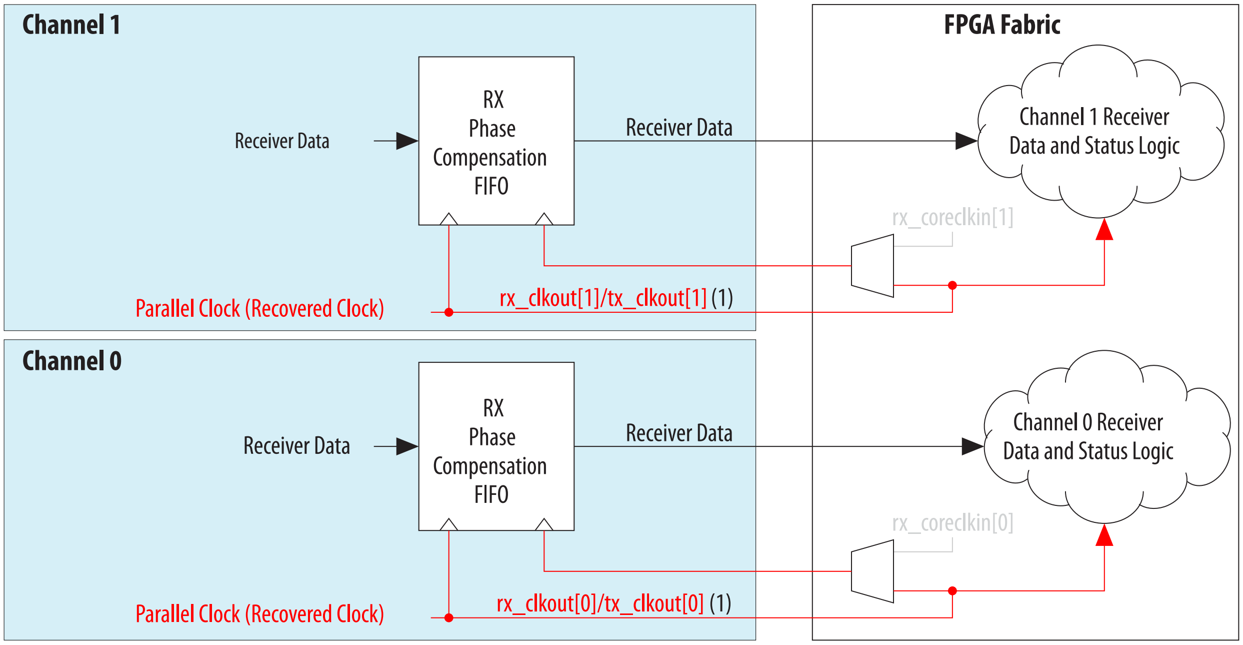
- **Transceiver Custom Configurations in Arria V Devices**
- **Transceiver Protocol Configurations in Arria V Devices**

Quartus II Software-Selected Receiver Datapath Interface Clock

Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the receiver datapath interface.

Figure 2-41: 6-Gbps Receiver Datapath Interface Clocking for Non-Bonded Channels

The figure shows receiver datapath interface of two 6-Gbps transceiver non-bonded channels clocked by their respective receiver PCS clocks, which are forwarded to the FPGA fabric.


Channel 1
Receiver Data
RX Phase Compensation FIFO
Receiver Data
Parallel Clock (Recovered Clock)
rx_clkout[1]/tx_clkout[1] (1)
FPGA Fabric
Channel 1 Receiver Data and Status Logic
rx_coreclkin[1]
Channel 0
Receiver Data
RX Phase Compensation FIFO
Receiver Data
Parallel Clock (Recovered Clock)
rx_clkout[0]/tx_clkout[0] (1)
Channel 0 Receiver Data and Status Logic
rx_coreclkin[0]

Note : (1) If you use a rate matcher, the tx_clkout clock is used.

Figure 2-42: Arria V GT/ST 10-Gbps Receiver Datapath Interface Clocking for Non-Bonded Channels

The figure shows Arria V GT/ST receiver datapath interface of two 10-Gbps transceiver non-bonded channels clocked by their respective receiver CDR recovered PMA clocks, which are forwarded to the FPGA fabric.

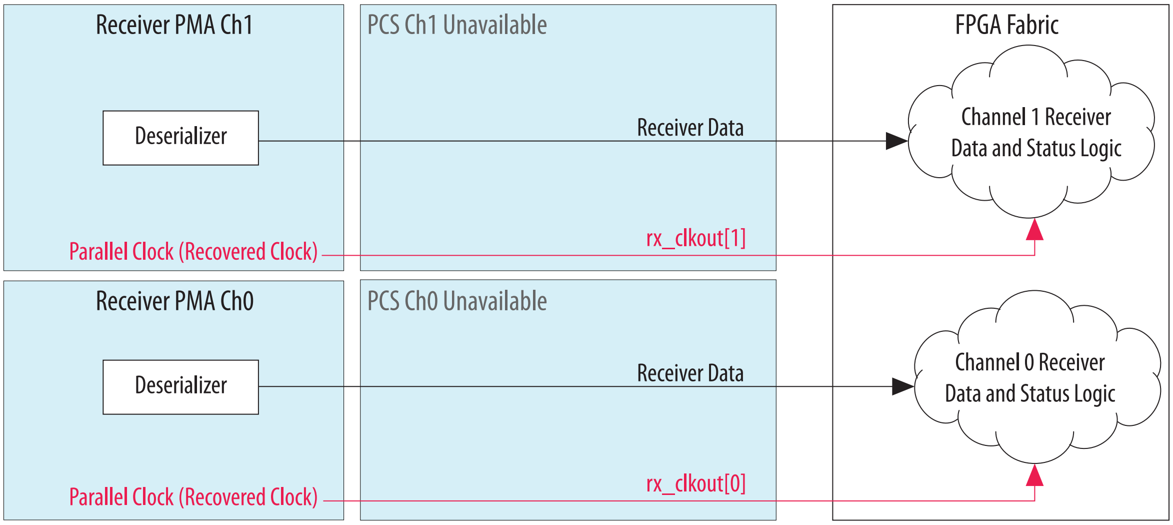
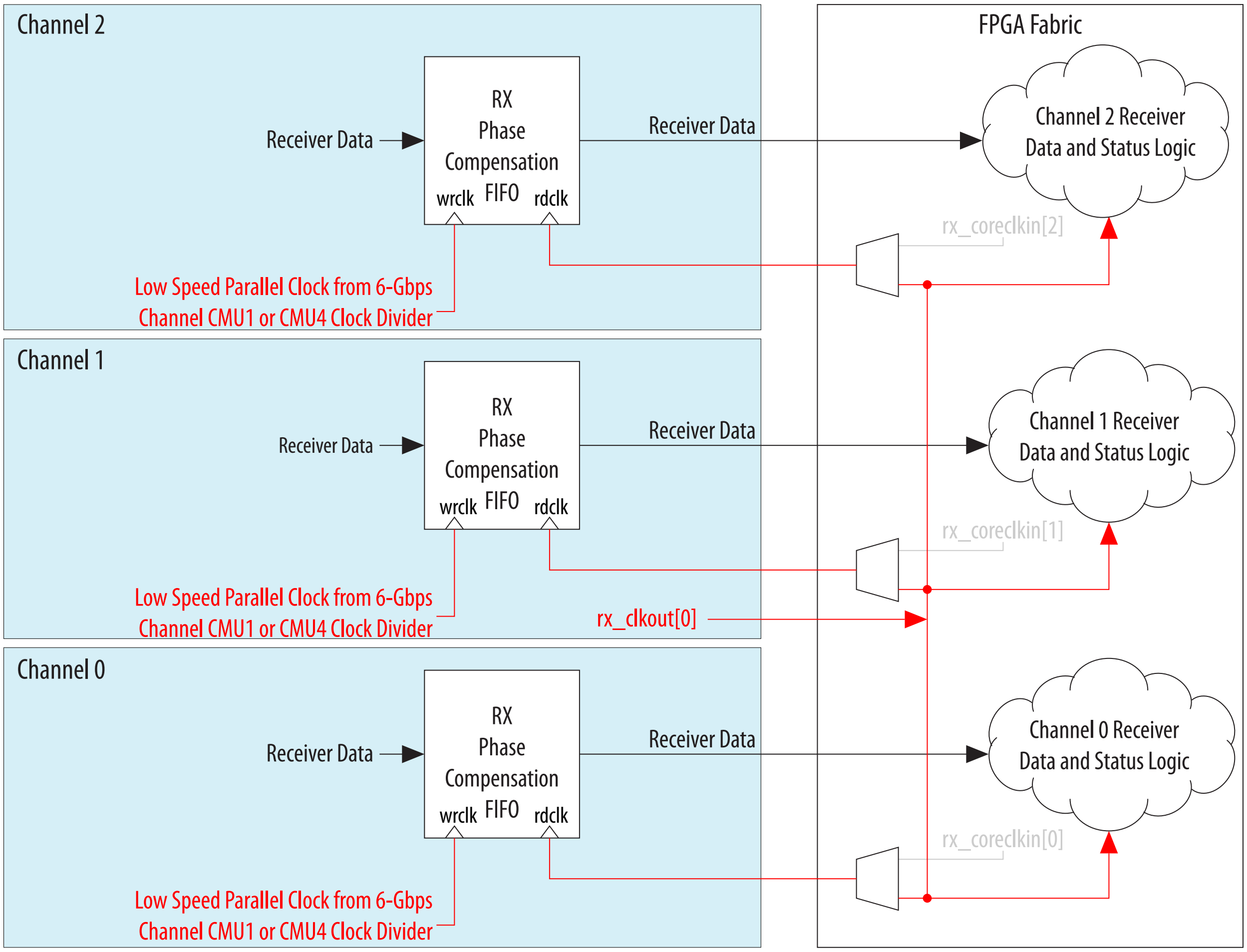


Figure 2-43: 6-Gbps Receiver Datapath Interface Clocking for Three Bonded Channels

The figure shows the 6-Gbps receiver datapath interface of three bonded channels clocked by the `rx_clkout[0]` clock. The `rx_clkout[0]` clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.



Selecting a Receiver Datapath Interface Clock

Multiple non-bonded receiver channels use a large portion of GCLK, RCLK, and PCLK resources. Selecting a common clock driver for the receiver datapath interface of all identical receiver channels saves clock resources.

Non-bonded multiple receiver channels lead to high utilization of GCLK, RCLK, and PCLK resources—one clock resource per channel. You can significantly reduce GCLK, RCLK, and PCLK resource use for the receiver datapath clocks if the receiver channels are identical.

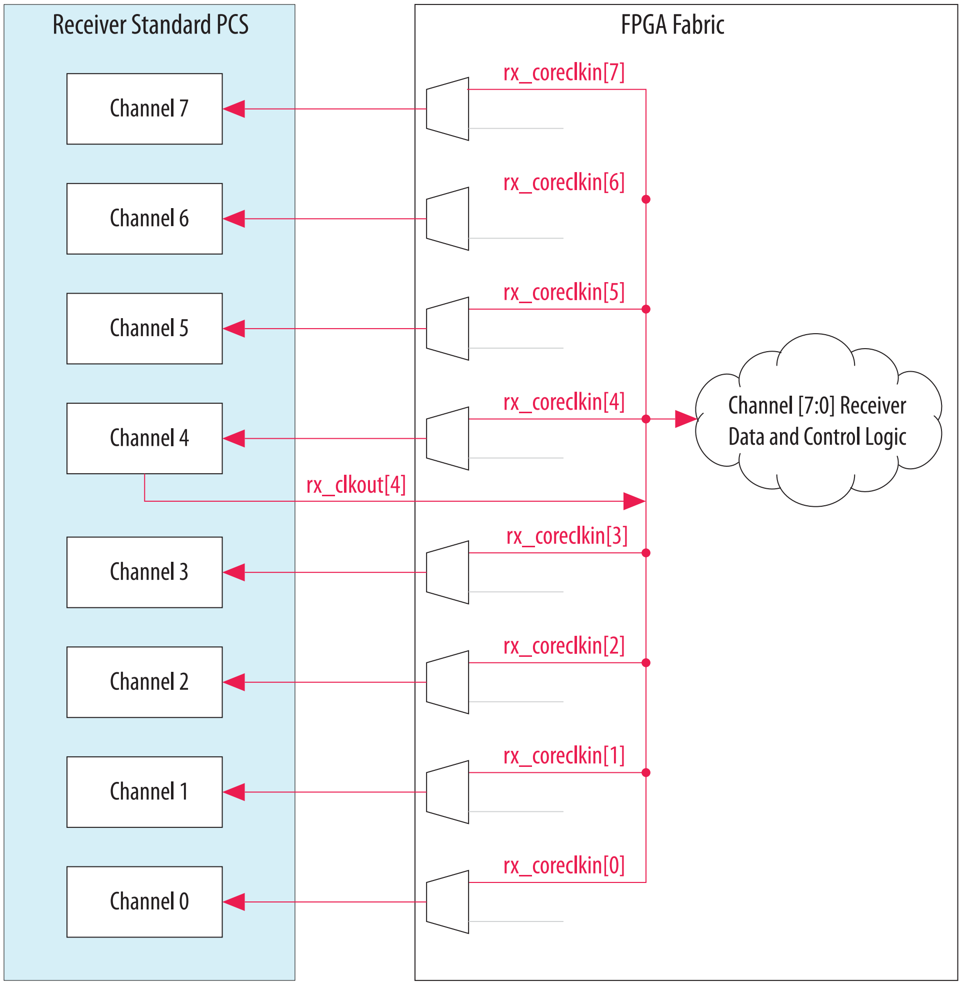
Note: Identical receiver channels are defined as channels that have the same input reference clock source for the CDR and the same receiver PMA and PCS configuration. Identical receiver channels need to be PPM aligned with regards to their remote transmitters. These channels may have different analog settings, such as receiver common mode voltage (V_{ICM}), equalization, or DC gain setting.

To achieve clock resource savings, select a common clock driver for the receiver datapath interface of all identical receiver channels. To select a common clock driver, perform these steps:

1. Instantiate the `rx_coreclkin` port for all the identical receiver channels.
2. Connect the common clock driver to their receiver datapath interface, and receiver data and control logic.

The following figure shows eight identical channels that are clocked by a single clock (`rx_clkout` of channel 4).

Figure 2-44: Eight Identical Channels with a Single User-Selected Receiver Interface Clock



To clock eight identical channels with a single clock, perform these steps:

- Instantiate the `rx_coreclkin` port for all the identical receiver channels (`rx_coreclkin[7:0]`).
- Connect `rx_clkout[4]` to the `rx_coreclkin[7:0]` ports.
- Connect `rx_clkout[4]` to the receiver data and control logic for all eight channels.

Note: Resetting or powering down channel 4 leads to a loss of the clock for all eight channels.

The common clock must have a 0 ppm difference for the write side of the RX phase compensation FIFO of all the identical channels. A frequency difference causes the FIFO to under run or overflow, depending on whether the common clock is faster or slower, respectively.

You can drive the 0 ppm common clock driver from one of the following sources:

- `tx_clkout` of any channel in non-bonded receiver channel configurations with the rate matcher
- `rx_clkout` of any channel in non-bonded receiver channel configurations without the rate matcher
- `tx_clkout[0]` in bonded receiver channel configurations
- Dedicated `refclk` pins

Note: The Quartus II software does not allow gated clocks or clocks generated in the FPGA logic to drive the `rx_coreclkin` ports.

Note: You must ensure a 0 ppm difference. The Quartus II software is unable to ensure a 0 ppm difference because it allows you to use external pins, such as dedicated `refclk` pins.

GXB 0 PPM Core Clock Assignment for GZ Devices

The common clock should have a 0 ppm difference with respect to the read side of the TX FIFO (in the 10G PCS channel) or TX phase compensation FIFO (in the Standard PCS channel) of all the identical channels. A frequency difference causes the FIFO to under-run or overflow, depending on whether the common clock is slower or faster, respectively.

The 0 ppm common clock driver can be driven by one of the following sources:

- `tx_clkout` in non-bonded channel configurations
- `tx_clkout[0]` in bonded channel configurations
- `rx_clkout` in non-bonded channel configurations
- `refclk` when there is 0 PPM difference between `refclk` and `tx_clkout`

Table 2-13: 0 PPM Core Clock Settings

The following table lists the 0 PPM core clock settings that you make in the Quartus II Assignment Editor.

Assignments[32]	Description
To	`tx_dataout/rx_datain pins` of all channels whose `tx/rx_coreclk` ports are connected together and driven by the 0 PPM clock driver.
Assignment Name	0 PPM coreclk setting
Value	ON

Note: For more information about QSF assignments and how 0 PPM is used with various transceiver PHYs, refer to the *Altera Transceiver PHY IP Core User Guide*.

Related Information

Altera Transceiver PHY IP Core User Guide

[32] You can find the full hierarchy name of the 0 PPM clock driver using the Node Finder feature in the Quartus II Assignment Editor.

Document Revision History

The table below lists the revision history for this chapter.

Table 2-14: Document Revision History

Date	Version	Changes
January 2016	2016.01.08	• Updated the Table "Input Reference Clock Sources" to indicate fPLL to ATX PLL input refclk cascading. • Added a note to the "Fractional PLL (fPLL)" section.
March 2015	2015.03.17	Updated the note in the Table "Characteristics of x1, x6, and xN Clock Lines".
September 2014	2014.09.30	• Updated the chapter to indicate that it is not recommended to use fractional PLL in fractional mode as a TX PLL or for PLL cascading. • Modified *Figure: Three Transceiver Bank Channels Bonded Using the PLL Feedback Compensation Path for GZ Devices* to indicate that fPLL does not support PLL feedback compensation when used as a TX PLL.
March 2014	2014.03.07	• Updated the Table "Input Reference Clock Sources". • Updated the Figure "Input Reference Clock Source for CMU PLL Driving Channels with Serial Data Rates Beyond 6.5536 Gbps". • Updated the Table "Characteristics of x1, x6, and xN Clock Lines". • Updated the Figure "Four Bonded Transmitter Channels Driven by fPLL using x6 Clock Line Within a Transceiver Bank" to indicate that when fPLL is used as a transmit PLL, all transceiver channels need to be placed in one transceiver bank. • Updated the tables "Clock Path for Non-Bonded Configurations" and "Clock Path for Bonded Configurations". • Corrected an error in the "Quartus II-Software Selected Transmitter Datapath Interface Clock" section.
October 2013	2013.10.18	• Updated "Input Reference Clocking" section.

Date	Version	Changes
May 2013	2013.05.06	• Updated for Quartus II software version 13.0 feature support. • Updated "Input Reference Clocking" section for Arria V GZ devices. • Updated "Internal Clocking" section for Arria V GZ devices. • Updated "FPGA Fabric Transceiver Interface Clocking" section for Arria V GZ devices. • Added link to the known document issues in the Knowledge Base.
March 2013	2013.03.15	• Updated Table 2-1: Input Reference Clock Sources • Updated Table 2-4: Characteristics of x1, x6, and xN Clock Lines • Updated Figure 2-8: x1 Clock Line Architecture (more than 6.5536 Gbps) • Updated "Transmitter Clock Network".
November 2012	2012.11.19	• Reorganized content and updated template. • Updated for the Quartus II software version 12.1.
June 2012	1.2	• Updated for the Quartus II software version 12.0. • Added basic clocking information from obsoleted “basics” chapter.
November 2011	1.1	Updated chapter for clarity and Quartus II software version 11.1.
August 2011	1.0	Initial release.

Transceiver Reset Control in Arria V Devices

3

2016.01.08

AV53003 Subscribe Send Feedback

Altera's recommended reset sequence ensures that both the physical coding sublayer (PCS) and physical medium attachment (PMA) in each transceiver channel are initialized and functioning correctly.

There are multiple reset options available to reset the analog and digital portions of the transmitter and receiver.

Table 3-1: Available Reset Options

Transceiver PHY IP Core	Embedded Reset Controller	User-Coded Reset Controller	Transceiver PHY Reset Controller IP	Avalon Memory-Mapped Reset Registers
XAUI	X			X
PCIe	X			X
10GBASE-R	X			X
Interlaken For Arria V GZ	X			X
Custom	X	X	X	X
Low Latency For Arria V GZ	X	X	X	X
Deterministic Latency	X	X	X	X
Native PHY		X	X	

Related Information

- **Arria V Device Handbook: Known Issues**
  Lists the planned updates to the *Arria V Device Handbook* chapters.
- **Altera Transceiver PHY IP Core User Guide**

PHY IP Embedded Reset Controller

The embedded reset controller in the PHY IP enables you to initialize the transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) blocks.

To simplify your transceiver-based design, the embedded reset controller provides an option that requires only one control input to implement an automatic reset sequence. Only one embedded reset controller is available for all the channels in a PHY IP instance.

The embedded reset controller automatically performs the entire transceiver reset sequence whenever the `phy_mgmt_clk_reset` signal is triggered. In case of loss-of-link or loss-of-data, the embedded reset controller asserts the appropriate reset signals. You must monitor `tx_ready` and `rx_ready`. A high on these status signals indicates the transceiver is out of reset and ready for data transmission and reception.

Note: Deassert the `mgmt_rst_reset` signal of the transceiver reconfiguration controller at the same time as `phy_mgmt_clk_reset` to start calibration.

Note: You must have a valid and stable ATX PLL reference clock before deasserting the `phy_mgmt_clk_reset` and `mgmt_rst_reset` signals for successful ATX PLL calibration.

ATX PLLs are available in Arria V GZ devices.

Note: The PHY IP embedded reset controller is enabled by default in all transceiver PHY IP cores except the Native PHY IP core.

Embedded Reset Controller Signals

The following figure shows the embedded reset controller and signals in the PHY IP instance. These signals reset your transceiver when you use the embedded reset controller.

Figure 3-1: Embedded Reset Controller

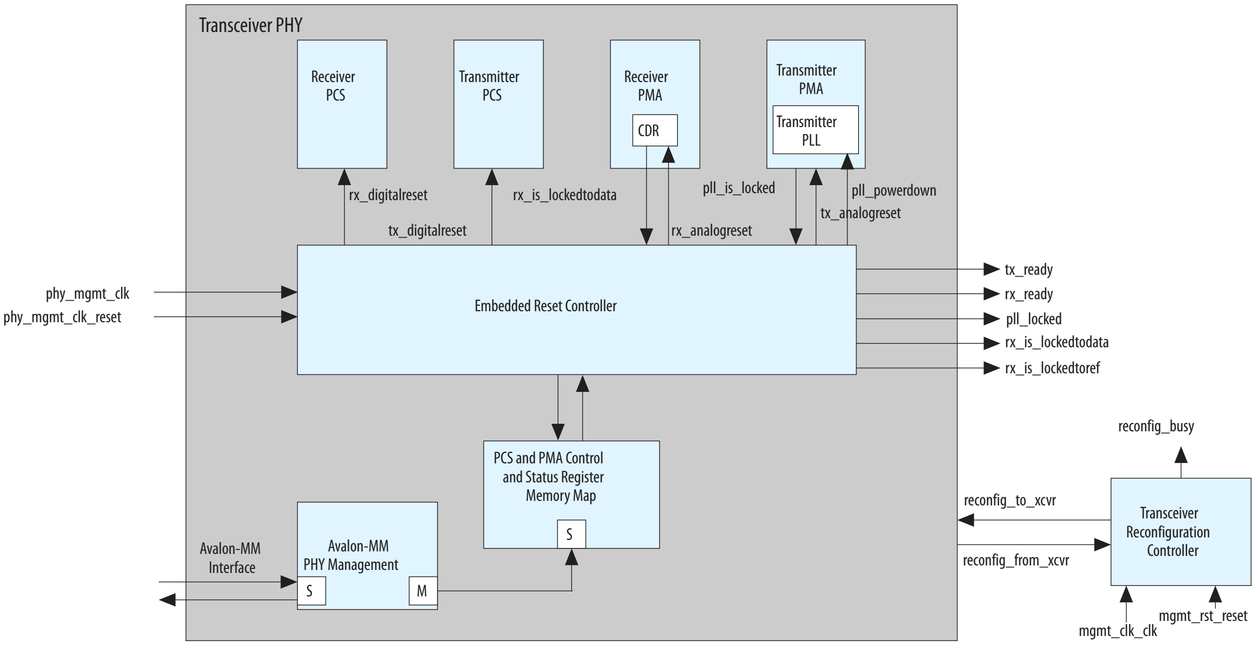


Table 3-2: Embedded Reset Controller Reset Control and Status Signals

Signal Name	Signal	Description
phy_mgmt_clk	Control Input	Clock for the embedded reset controller.
phy_mgmt_clk_reset	Control Input	A high-to-low transition of this asynchronous reset signal initiates the automatic reset sequence control. Hold this signal high to keep the reset signals asserted.
tx_ready	Status Output	A continuous high on this signal indicates that the transmitter (TX) channel is out of reset and is ready for data transmission. This signal is synchronous to phy_mgmt_clk.
rx_ready	Status Output	A continuous high on this signal indicates that the receiver (RX) channel is out of reset and is ready for data reception. This signal is synchronous to phy_mgmt_clk.
reconfig_busy	Status Output	An output from the Transceiver Reconfiguration Controller block indicates the status of the dynamic reconfiguration controller. At the first mgmt_clk_clk clock cycle after power-up, reconfig_busy remains low. This signal is asserted from the second mgmt_clk_clk clock cycle to indicate that the calibration process is in progress . When the calibration process is completed, the reconfig_busy signal is deasserted. This signal is also routed to the embedded reset controller by the Quartus® II software by embedding the signal in the reconfig_to_xcvr bus between the PHY IP and the Transceiver Reconfiguration Controller.
pll_locked	Status Output	This signal is asserted when the TX PLL achieves lock to the input reference clock. When this signal is asserted high, the embedded reset controller deasserts the tx_digitalreset signal.
rx_is_lockedtodata	Status Output	This signal is an optional output status port. When asserted, this signal indicates that the CDR is locked to the RX data and the CDR has changed from lock-to-reference (LTR) to lock-to-data (LTD) mode.
rx_is_lockedtoref	Status Output	This is an optional output status port. When asserted, this signal indicates that the CDR is locked to the reference clock.
mgmt_clk_clk	Clock	Clock for the Transceiver Reconfiguration Controller. This clock must be stable before releasing mgmt_rst_reset.

Signal Name	Signal	Description
mgmt_rst_reset	Reset	Reset for the Transceiver Reconfiguration Controller

Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Power-Up

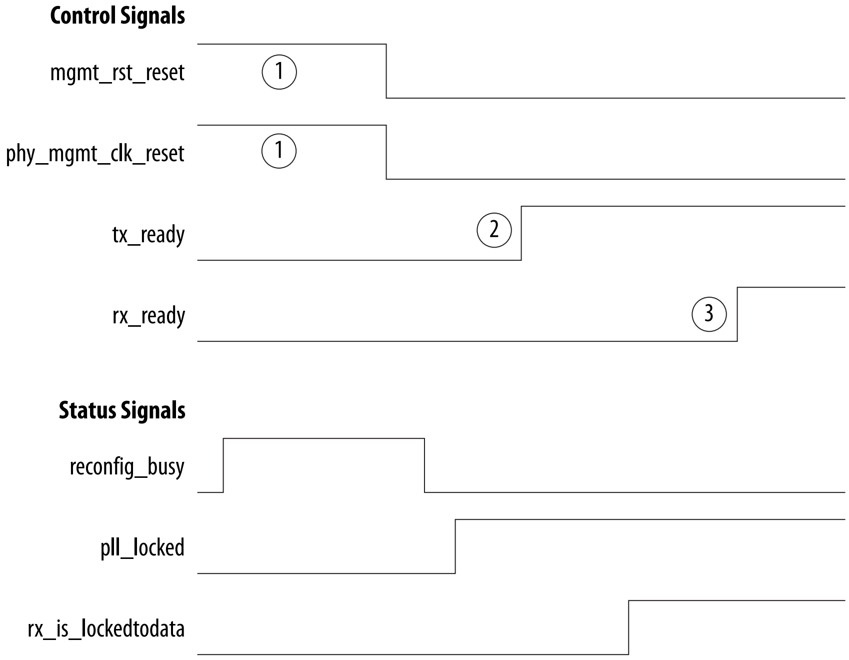
Follow this reset sequence to ensure a reliable link initialization after the initial power-up.

The numbers in the following figure correspond to the following numbered list, which guides you through the transceiver reset sequence during device power-up.

1. During device power-up, mgmt_rst_reset and phy_mgmt_clk_reset must be asserted to initialize the reset sequence. phy_mgmt_clk_reset holds the transceiver blocks in reset and mgmt_rst_reset is required to start the calibration IPs. Both these signals should be held asserted for a minimum of two phy_mgmt_clk clock cycles. If phy_mgmt_clk_reset and mgmt_rst_reset are driven by the same source, deassert them at the same time. If the two signals are not driven by the same source, phy_mgmt_clk_reset must be deasserted before mgmt_rst_reset.
2. After the transmitter calibration and reset sequence are complete, the tx_ready status signal is asserted and remains asserted to indicate that the transmitter is ready to transmit data.
3. After the receiver calibration and reset sequence are complete, the rx_ready status signal is asserted and remains asserted to indicate that the receiver is ready to receive data.

Note: If the tx_ready and rx_ready signals do not stay asserted, the reset sequence did not complete successfully and the link will be down.

Figure 3-2: Reset Sequence Timing Diagram Using Embedded Reset Controller during Device Power-Up



Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Operation

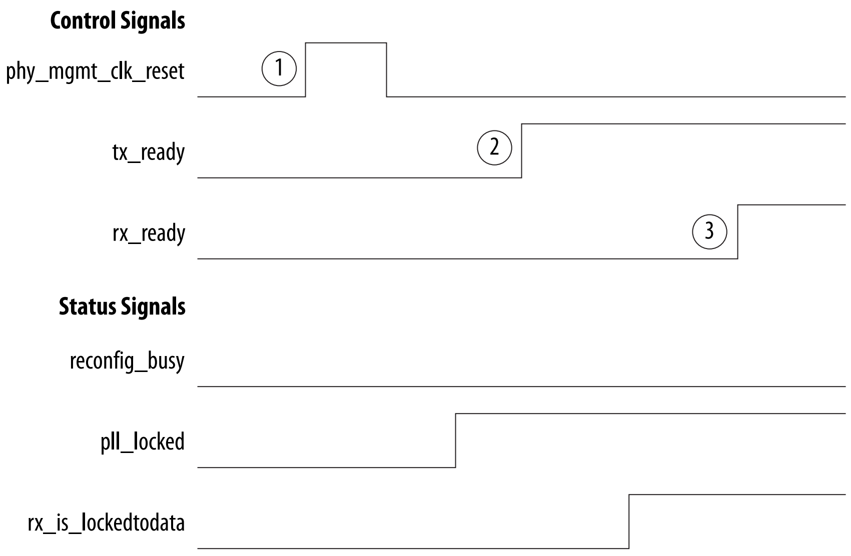
Follow this reset sequence to reset the entire transceiver at any point during the device operation, to re-establishing a link, or after certain dynamic reconfigurations.

The numbers in the following figure correspond to the numbered list, which guides you through the transceiver reset sequence during device operation.

1. Assert `phy_mgmt_clk_reset` for two `phy_mgmt_clk` clock cycles to re-start the entire transceiver reset sequence.
2. After the transmitter reset sequence is complete, the `tx_ready` status signal is asserted and remains asserted to indicate that the transmitter is ready to transmit data.
3. After the receiver reset sequence is complete, the `rx_ready` status signal is asserted and remains asserted to indicate that the receiver is ready to receive data.

Note: If the `tx_ready` and `rx_ready` signals do not stay asserted, the reset sequence did not complete successfully and the link will be down.

Figure 3-3: Reset Sequence Timing Diagram Using Embedded Reset Controller during Device Operation



Note: To reset the transmitter and receiver analog and digital blocks separately without repeating the entire reset sequence, use the Avalon Memory Map registers.

User-Coded Reset Controller

You must implement external reset controller logic (user-coded reset controller) if you disable the embedded reset controller to initialize the transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) blocks.

You can implement a user-coded reset controller with one of the following:

- Using your own Verilog/VHDL code to implement the reset sequence
- Using the Quartus II IP Catalog, which provides a ready-made reset controller IP to place your own Verilog/VHDL code

When using manual mode, you must create a user-coded reset controller to manage the input signals.

Note: You must disable the embedded reset controller before using the user-coded reset controller.

Note: The embedded reset controller can only be disabled for non-protocol transceiver PHY IPs, such as 10GBASE-R PHY, custom PHY, low latency PHY and deterministic latency PHY. Native PHY IP does not have an embedded reset controller, so you must implement your own reset logic.

If you implement your own reset controller, consider the following:

- The user-coded reset controller must be level sensitive (active high)
- The user-coded reset controller does not depend on `phy_mgmt_clk_reset`
- You must provide a clock and reset to the reset controller logic
- The internal signals of the PHY IP embedded reset controller are configured as ports
- You can hold the transceiver channels in reset by asserting the appropriate reset control signals

Note: You must have a valid and stable ATX PLL reference clock before deasserting the `pll_powerdown` and `mgmt_rst_reset` signals for successful ATX PLL calibration.

ATX PLLs are available in Arria V GZ devices.

Related Information

"Transceiver PHY Reset Controller IP Core" chapter of the Altera Transceiver PHY IP Core User Guide.

For information about the transceiver PHY reset controller.

User-Coded Reset Controller Signals

Use the signals in the following figure and table with a user-coded reset controller.

Figure 3-4: Interaction Between the Transceiver PHY Instance, Transceiver Reconfiguration Controller, and the User-Coded Reset Controller

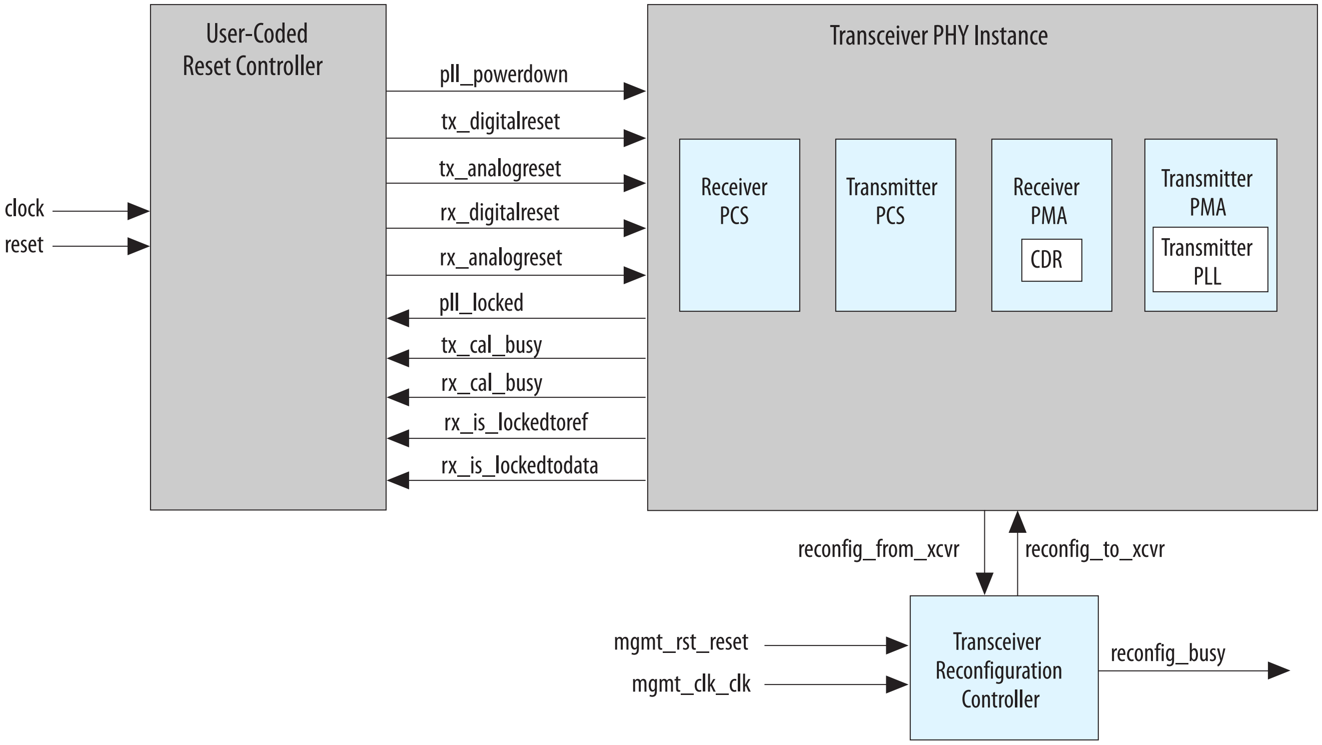


Table 3-3: Signals Used by the Transceiver PHY instance, Transceiver Reconfiguration Controller, and User-Coded Reset Controller

Signal Name	Signal Type	Description
`mgmt_clk_clk`	Clock	Clock for the Transceiver Reconfiguration Controller. This clock must be stable before releasing `mgmt_rst_reset`.
`mgmt_rst_reset`	Reset	Reset for the Transceiver Reconfiguration Controller
`pll_powerdown`	Control	Resets the TX PLL when asserted high

Signal Name	Signal Type	Description
`tx_analogreset`	Control	Resets the TX PMA when asserted high
`tx_digitalreset`	Control	Resets the TX PCS when asserted high
`rx_analogreset`	Control	Resets the RX PMA when asserted high
`rx_digitalreset`	Control	Resets the RX PCS when asserted high
`reconfig_busy`	Status	A high on this signal indicates that reconfiguration is active
`tx_cal_busy`	Status	A high on this signal indicates that TX calibration is active
`rx_cal_busy`	Status	A high on this signal indicates that RX calibration is active
`pll_locked`	Status	A high on this signal indicates that the TX PLL is locked
`rx_is_lockedtoref`	Status	A high on this signal indicates that the RX CDR is in the lock to reference (LTR) mode
`rx_is_lockedtodata`	Status	A high on this signal indicates that the RX CDR is in the lock to data (LTD) mode

Resetting the Transmitter with the User-Coded Reset Controller During Device Power-Up

Follow this reset sequence when designing your User-Coded Reset Controller to ensure a reliable transmitter initialization after the initial power-up.

The numbers in the figure correspond to the following numbered list, which guides you through the transmitter reset sequence during device power-up.

1. To reset the transmitter, begin with:
    - Assert `mgmt_rst_reset` at power-up to start the calibration IPs. Hold `mgmt_rst_reset` active for a minimum of two reset controller clock cycles.
    - Assert and hold `pll_powerdown`, `tx_analogreset`, and `tx_digitalreset` at power-up to reset the transmitter. You can deassert `tx_analogreset` at the same time as `pll_powerdown`.
    - Assert `pll_powerdown` for a minimum duration of 1 µs ($t_{pll\_powerdown}$). If you use ATX PLL calibration (available in Arria V GZ devices), deassert `pll_powerdown` before `mgmt_rst_reset` so that the ATX PLL is not powered down during calibration. Otherwise, `pll_powerdown` can be deasserted anytime after `mgmt_rst_reset` is deasserted.
    - Make sure there is a stable reference clock to the PLL before deasserting `pll_powerdown` and `mgmt_rst_reset`.
2. After the transmitter PLL locks, the `pll_locked` status gets asserted after t_{pll_lock}.
3. After the transmitter calibration completes, the `tx_cal_busy` status is deasserted. Depending on the transmitter calibrations, this could happen before or after the `pll_locked` is asserted.
4. Deassert `tx_digitalreset` after the gating conditions occur for a minimum duration of $t_{tx\_digitalreset}$. The gating conditions are:

- `pll_powerdown` is deasserted
- `pll_locked` is asserted
- `tx_cal_busy` is deasserted

The transmitter is out of reset and ready for operation.

Note: During calibration, `pll_locked` might assert and deassert as the calibration IP runs.

Figure 3-5: Reset Sequence Timing Diagram for Transmitter using the User-Coded Reset Controller during Device Power-Up

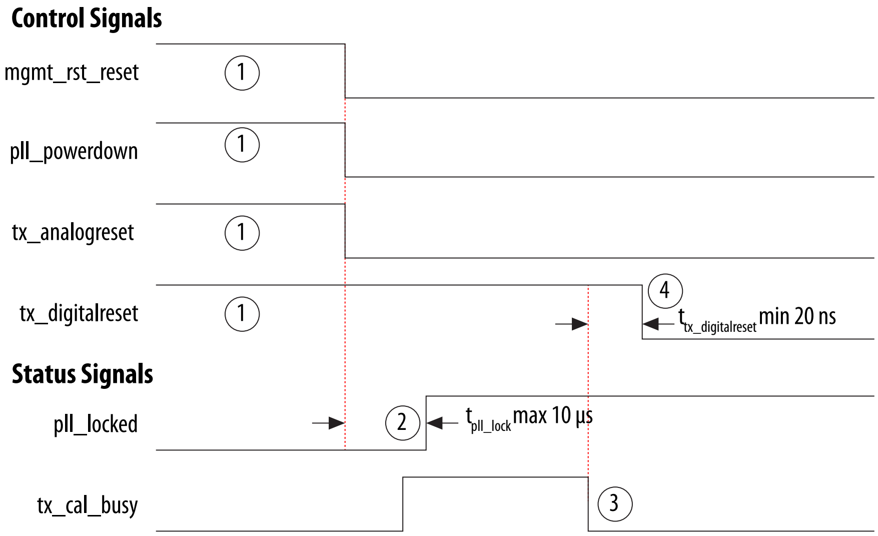


Table 3-4: Guidelines for Resetting the PLL, TX PMA, and TX PCS

To Reset	You Must Reset

PLL	`pll_powerdown`
	`tx_analogreset`
	`tx_digitalreset`

TX PMA	`tx_analogreset` `tx_digitalreset`
TX PCS	`tx_digitalreset`

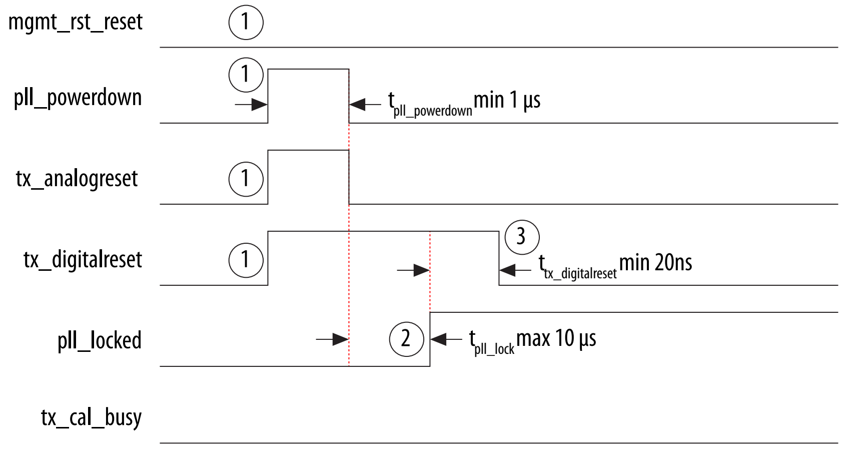
Resetting the Transmitter with the User-Coded Reset Controller During Device Operation

Follow this reset sequence if you want to reset the PLL, or analog or digital blocks of the transmitter at any point during device operation. This might be necessary for re-establishing a link or after certain dynamic reconfigurations.

The numbers in the following figure correspond to the following numbered list, which guides you through the transmitter reset sequence during device operation.

1. To reset the transmitter:
   - Assert `pll_powerdown`, `tx_analogreset` and `tx_digitalreset`. `tx_digitalreset` must be asserted every time `pll_powerdown` and `tx_analogreset` are asserted to reset the PCS blocks.
   - Hold `pll_powerdown` asserted for a minimum duration of $t_{pll\_powerdown}$.
   - Deassert `tx_analogreset` at the same time or after `pll_powerdown` is deasserted.
2. After the transmitter PLL locks, the `pll_locked` status is asserted after t_{pll_lock}. While the TX PLL locks, the `pll_locked` status signal may toggle. It is asserted after t_{pll_lock}.
3. Deassert `tx_digitalreset` after a minimum duration of $t_{tx\_digitalreset}$, and after all the gating conditions are removed:
   - `pll_powerdown` is deasserted
   - `pll_locked` is deasserted
   - `tx_cal_busy` is deasserted

Figure 3-6: Reset Sequence Timing Diagram for Transmitter using the User-Coded Reset Controller during Device Operation


mgmt_rst_reset
1
pll_powerdown
1
$t_{pll\_powerdown}$ min 1 µs
tx_analogreset
1
tx_digitalreset
1
3
$t_{tx\_digitalreset}$ min 20ns
pll_locked
2
t_{pll_lock} max 10 µs
tx_cal_busy

Resetting the Receiver with the User-Coded Reset Controller During Device Power-Up Configuration

Follow this reset sequence to ensure a reliable receiver initialization after the initial power-up.

The numbers in the following figure correspond to the following numbered list, which guides you through the receiver reset sequence during device power-up.

1. Assert `mgmt_rst_reset` at power-up to start the calibration IPs. Hold `mgmt_rst_reset` active for a minimum of two `mgmt_clk_clock` cycles. Hold `rx_analogreset` and `rx_digitalreset` active at

power-up to hold the receiver in reset. You can deassert them after all the gating conditions are removed.

2. After the receiver calibration completes, the `rx_cal_busy` status is deasserted.
3. Deassert `rx_analogreset` after a minimum duration of $t_{rx\_analogreset}$ after `rx_cal_busy` is deasserted.
4. `rx_is_lockedtodata` is a status signal from the receiver CDR indicating that the CDR is in the lock to data (LTD) mode. Ensure `rx_is_lockedtodata` is asserted and stays asserted for a minimum duration of t_{LTD} before deasserting `rx_digitalreset`. If `rx_is_lockedtodata` is asserted and toggles, you must wait another additional t_{LTD} duration before deasserting `rx_digitalreset`.
5. Deassert `rx_digitalreset` after a minimum duration of t_{LTD} after `rx_is_lockedtodata` stays asserted. Ensure `rx_analogreset` and `rx_cal_busy` are deasserted before deasserting `rx_digitalreset`.

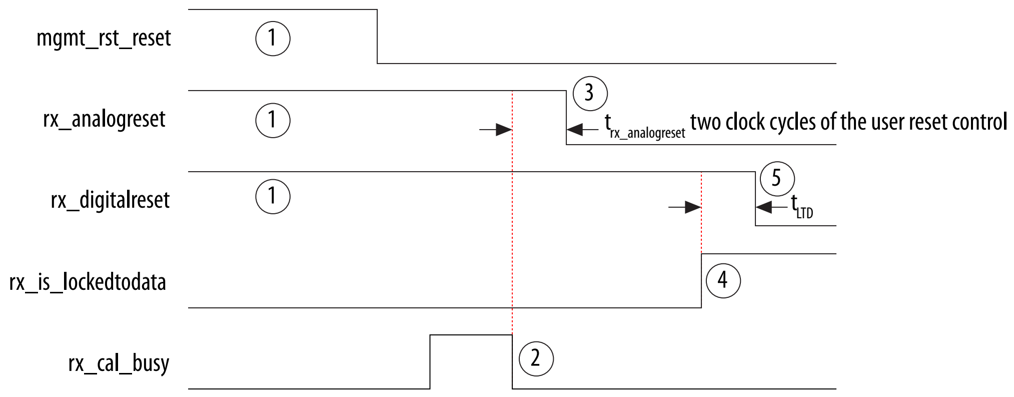
The receiver is now out of reset and ready for operation.

Note: `rx_is_lockedtodata` might toggle when there is no data at the receiver input.

Note: `rx_is_lockedtoref` is a don't care when `rx_is_lockedtodata` is asserted.

Note: `rx_analogreset` must always be followed by `rx_digitalreset`.

Figure 3-7: Reset Sequence Timing Diagram for Receiver using the User-Coded Reset Controller during Device Power-Up



Related Information

[Transceiver Architecture in Arria V Devices](#)

For information about CDR lock modes.

Resetting the Receiver with the User-Coded Reset Controller During Device Operation

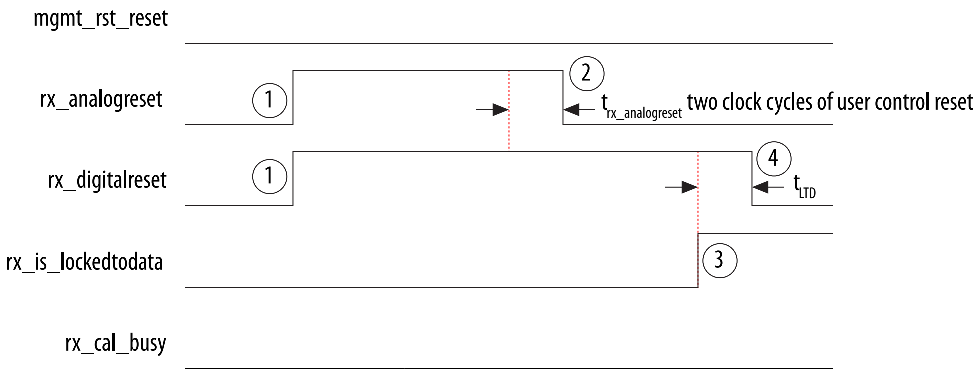
Follow this reset sequence to reset the analog or digital blocks of the receiver at any point during the device operation. This might be necessary for re-establishing a link or after certain dynamic reconfigurations.

The numbers in the following figure correspond to the following numbered list, which guides you through the receiver reset sequence during device operation.

1. Assert `rx_analogreset` and `rx_digitalreset` at any point independently. However, you must assert `rx_digitalreset` every time `rx_analogreset` is asserted to reset the PCS blocks.
2. Deassert `rx_analogreset` after a minimum duration of 40 ns ($t_{rx\_analogreset}$).
3. `rx_is_lockedtodata` is a status signal from the receiver CDR that indicates that the CDR is in the lock to data (LTD) mode. Ensure `rx_is_lockedtodata` is asserted and stays asserted before deasserting `rx_digitalreset`.
4. Deassert `rx_digitalreset` after a minimum duration of t_{LTD} after `rx_is_lockedtodata` stays asserted. Ensure `rx_analogreset` is deasserted.

Note: `rx_is_lockedtodata` might toggle when there is no data at the receiver input. `rx_is_lockedtoref` is a don't care when `rx_is_lockedtodata` is asserted.

Figure 3-8: Reset Sequence Timing Diagram for Receiver using the User-Coded Reset Controller during Device Operation



Related Information

Transceiver Architecture in Arria V Devices

For information about CDR lock modes.

Transceiver Reset Using Avalon Memory Map Registers

You can use Memory Map registers within the PHY IP instance to control the reset signals through the Avalon Memory Map interface.

This gives the flexibility of resetting the PLL, and transmitter and receiver analog and digital blocks separately without repeating the entire reset sequence.

Transceiver Reset Control Signals Using Avalon Memory Map Registers

The following table lists the memory map registers for CDR lock mode and channel reset. These signals help you reset your transceiver when you use Memory Map registers within the PHY IP.

Table 3-5: Transceiver Reset Control Using Memory Map Registers

Register Name	Description
`pma_rx_set_locktodata`	This register is for CDR manual lock mode only. When you set the register to high, the RX CDR PLL is in the lock to data (LTD) mode. The default is low when both registers have the CDR in auto lock mode.
`pma_rx_set_locktoref`	This register is for CDR manual lock mode only. When you set the register to high, the RX CDR PLL is in the lock to reference (LTR) mode if `pma_rx_set_lockedtodata` is not asserted. The default is low when both registers have the CDR in auto lock mode.
`reset_tx_digital`	When you set this register to high, the `tx_digitalreset` signal is asserted in every channel that is enabled for reset control through the `reset_ch_bitmask` register. To deassert the `tx_digitalreset` signal, set the `reset_tx_digital` register to 0.
`reset_rx_analog`	When you set this register to high, the `rx_analogreset` signal is asserted in every channel that is enabled for reset control through the `reset_ch_bitmask` register. To deassert the `rx_analogreset` signal, set the `reset_rx_analog` register to 0.
`reset_rx_digital`	When you set this register to high, the `rx_digitalreset` signal is asserted in every channel that is enabled for reset control through the `reset_ch_bitmask` register. To deassert the `rx_digitalreset` signal, set the `reset_rx_digital` register to 0.
`reset_ch_bitmask`	The registers provide an option to enable or disable certain channels in a PHY IP instance for reset control. By default, all channels in a PHY IP instance are enabled for reset control.
`pll_powerdown`	When asserted, the TX phase-locked loop (PLL) is turned off.

Related Information

Altera Transceiver PHY IP Core User Guide
For information about register addresses.

Clock Data Recovery in Manual Lock Mode

Use the clock data recovery (CDR) manual lock mode to override the default CDR automatic lock mode depending on your design requirements.

Related Information

Transceiver PHY Reset Controller IP Core chapter of the V-Series Transceiver PHY IP Core User Guide.

Refer to the description of the `rx_digitalreset` signal in the "Top-Level Signals" table for information about using the manual lock mode.

Control Settings for CDR Manual Lock Mode

Use the following control settings to set the CDR lock mode:

Table 3-6: Control Settings for the CDR in Manual Lock Mode

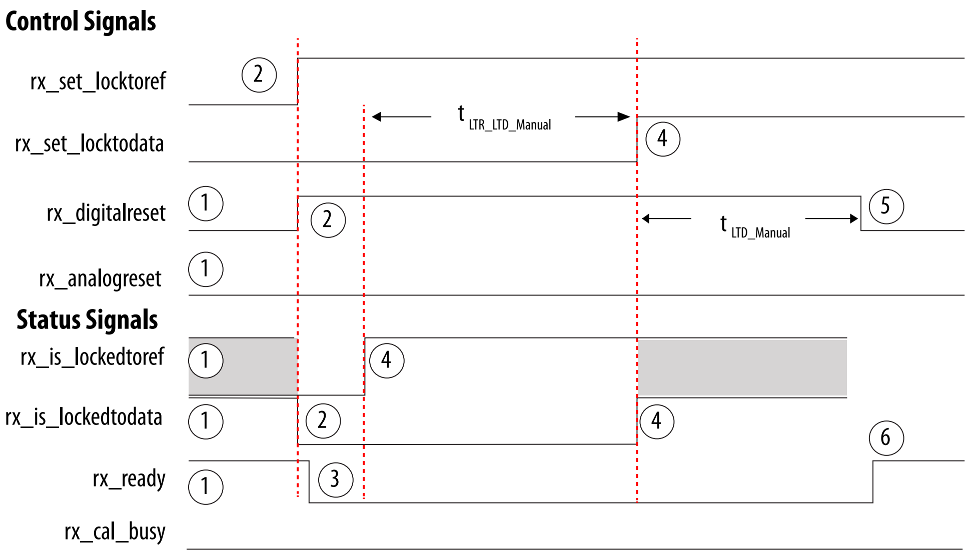
rx_set_locktoref	rx_set_locktodata	CDR Lock Mode
0	0	Automatic
1	0	Manual-RX CDR LTR
X	1	Manual-RX CDR LTD

Resetting the Transceiver in CDR Manual Lock Mode

The numbers in this list correspond to the numbers in the following figure, which guides you through the steps to put the CDR in manual lock mode.

1. Make sure that the calibration is complete (`rx_cal_busy` is low) and the transceiver goes through the initial reset sequence. The `rx_digitalreset` and `rx_analogreset` signals should be low. The `rx_is_lockedtoref` is a don't care and can be either high or low. The `rx_is_lockedtodata` and `rx_ready` signals should be high, indicating that the transceiver is out of reset. Alternatively, you can start directly with the CDR in manual lock mode after the calibration is complete.
2. Assert the `rx_set_locktoref` signal high to switch the CDR to the lock-to-reference mode. The `rx_is_lockedtodata` status signal is deasserted. Assert the `rx_digitalreset` signal high at the same time or after `rx_set_lockedtoref` is asserted if you use the user-coded reset. When the Transceiver PHY reset controller is used, the `rx_digitalreset` is automatically asserted.
3. After the `rx_digitalreset` signal gets asserted, the `rx_ready` status signal is deasserted.
4. Assert the `rx_set_locktodata` signal high, $t_{LTR\_LTD\_Manual}$ (minimum 15 µs) after the CDR is locked to reference. `rx_is_locktoref` should be high and stable for a minimum $t_{LTR\_LTD\_Manual}$ (15 µs), before asserting `rx_set_locktodata`. This is required to filter spurious glitches on `rx_is_lockedtoref`. The `rx_is_lockedtodata` status signal gets asserted, which indicates that the CDR is now set to LTD mode.
   The `rx_is_lockedtoref` status signal can be a high or low and can be ignored after asserting `rx_set_locktodata` high after the CDR is locked to reference.
5. Deassert the `rx_digitalreset` signal after a minimum of t_{LTD_Manual} (4 µs).
6. If you are using the Transceiver PHY Reset Controller, the `rx_ready` status signal gets asserted after the `rx_digitalreset` signal is deasserted. This indicates that the receiver is now ready to receive data with the CDR in manual mode.

Figure 3-9: Reset Sequence Timing Diagram for Receiver when CDR is in Manual Lock Mode



Resetting the Transceiver During Dynamic Reconfiguration

Reset is required for transceiver during dynamic reconfiguration except in the PMA Analog Control Reconfiguration mode.

In general, follow these guidelines when dynamically reconfiguring the transceiver:

1. Hold the targeted channel and PLL in the reset state before dynamic reconfiguration starts.
2. Repeat the sequence as needed after dynamic reconfiguration is complete, which is indicated by deassertion of the `reconfig_busy`, `tx_cal_busy`, `rx_cal_busy` signals.

Guidelines for Dynamic Reconfiguration if Transmitter Duty Cycle Distortion Calibration is Required During Device Operation

If transmitter duty cycle distortion calibration is required during device operation, ensure the general guidelines for transceiver dynamic reconfiguration are followed. Additionally, use the following recommendations:

1. Do not connect `tx_cal_busy` to the transceiver Reset Controller IP.
2. Disable the embedded reset controller and use an external reset controller.

   Note: If channel reconfiguration is required before TX DCD calibration, ensure the following:

- The TX PLL, TX channel, and Transceiver Reconfiguration Controller blocks must not be in the reset state during TX DCD calibration. Ensure the following signals are not asserted during TX DCD calibration:
  - `pll_powerdown`
  - `tx_digitalreset`
  - `tx_analogreset`
  - `mgmt_rst_reset`

Repeat the reset sequence when TX DCD calibration is complete.

Arria V GZ devices do not require channel reconfiguration before TX dynamic reconfiguration calibration.

Note: Reset signals for the PMA are required only in PMA-direct mode.

Transceiver Blocks Affected by the Reset and Powerdown Signals

The following table lists blocks that are affected by specific reset and powerdown signals.

Table 3-7: Transceiver Blocks Affected

Transceiver Block	pll_ powerdown	rx_digital-reset	rx_analogr-eset	tx_digitalreset	tx_analogreset
PLL					
CMU PLL	Yes	—	—	—	—
ATX PLL for Arria V GZ	Yes	—	—	—	—
Receiver Standard PCS					
Receiver Word Aligner	—	Yes	—	—	—
Receiver Deskew FIFO	—	Yes	—	—	—
Receiver Rate Match FIFO	—	Yes	—	—	—
Receiver 8B/10B Decoder	—	Yes	—	—	—
Receiver Byte Deserializer	—	Yes	—	—	—
Receiver Byte Ordering	—	Yes	—	—	—
Receiver Phase Compensation FIFO	—	Yes	—	—	—
Receiver 10G PCS in Arria V GZ Devices					
Receiver Gear Box	—	Yes	—	—	—
Receiver Block Synchronizer	—	Yes	—	—	—
Receiver Disparity Checker	—	Yes	—	—	—
Receiver Descrambler	—	Yes	—	—	—

Transceiver Block	pll_ powerdown	rx_digital-reset	rx_analogr-eset	tx_digitalreset	tx_analogreset
Receiver Frame Sync	—	Yes	—	—	—
Receiver 64B/66B Decoder	—	Yes	—	—	—
Receiver CRC32 Checker	—	Yes	—	—	—
Receiver FIFO	—	Yes	—	—	—
Receiver PMA					
Receiver Buffer	—	—	Yes	—	—
Receiver CDR	—	—	Yes	—	—
Receiver Deserializer	—	—	Yes	—	—
Transmitter Standard PCS					
Transmitter Phase Compensation FIFO	—	—	—	Yes	—
Byte Serializer	—	—	—	Yes	—
8B/10B Encoder	—	—	—	Yes	—
Transmitter Bit-Slip	—	—	—	Yes	—
Transmitter 10G PCS in Arria V GZ Devices					
Transmitter FIFO	—	—	—	Yes	—
Transmitter Frame Generator	—	—	—	Yes	—
Transmitter CRC32 Generator	—	—	—	Yes	—
Transmitter 64B/66B Encoder	—	—	—	Yes	—
Transmitter Scrambler	—	—	—	Yes	—
Transmitter Disparity Generator	—	—	—	Yes	—
Transmitter Gear Box	—	—	—	Yes	—
Transmitter PMA					
Transmitter Central/Local Clock Divider	—	—	—	—	Yes
Serializer	—	—	—	—	Yes
Transmitter Buffer	—	—	—	—	Yes

Transceiver Power-Down

To maximize power savings, enable PMA hard power-down across all channels on a side of the device where you do not use the transceivers.

The hard power-down granularity control of the transceiver PMA is per side (Arria V GX & Arria V GT) or per transceiver bank (Arria V GZ). To enable PMA hard power-down on the left or right side of the device, ground the transceiver power supply of the respective side.

VCCE_GXBL and VCCL_GXBL must be connected either to the required supply or to GND. The VCCH_GXBL pin must always be powered.

Related Information

- **Arria V Device Datasheet**
  For information about the transceiver power supply operating conditions of Arria V devices.
- **Arria V Device Family Pin Connection Guidelines**

Document Revision History

Date	Version	Changes
January 2016	2016.01.08	Added information about VCCE_GXBL and VCCL_GXBL to the "Transceiver Power-Down" section.
September 2014	2014.09.30	• Added information about using signals to the "Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Power-Up" section. • Added statement about using manual mode to the "User-Coded Reset Controller" section. • Added a link to the Related Links in the "Clock Data Recovery in Manual Lock Mode" section.
March 2014	2014.03.07	• Changed "User-Controlled Reset Controller" term to "User-Coded Reset Controller". • Updated the "Resetting the Transceiver in CDR Manual Lock Mode" section.
May 2013	2013.05.06	• Updated the guidelines for Dynamic Reconfiguration if TX DCD Calibration is required during device operation • Added link to the known document issues in the Knowledge Base.

Date	Version	Changes
November 2012	2012.11.19	• Rewritten and reorganized content, and updated template • Updated reset sequence procedures • Included sequences for resetting transceiver during device operation • Included information for Arria V GZ transceiver reset
June 2012	1.2	• Added "User-Controlled Reset Controller" section. • Updated Figure 3–1 and Table 3–1.
November 2011	1.1	• Updated all figures and tables. • Reorganized and updated the "Transceiver Reset Sequence" section.

Transceiver Protocol Configurations in Arria V Devices

4

2016.01.08

AV53004 Subscribe Send Feedback

The dedicated transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) circuitry supports the following communication protocols.

Table 4-1: Transceiver PCS Features for Arria V Devices

PCS Support	Data Rates (Gbps)	Transmitter Datapath	Receiver Datapath
PCI Express® (PCIe®) Gen1 (x1, x2, x4, and x8) and Gen2 (x1, x2, and x4)	2.5 (Gen1), 5 (Gen2)	The same as custom single- and double-width modes, plus the PHY interface for PCI Express (PIPE) 2.0 interface to the core logic	The same as custom single- and double-width modes, plus the rate match FIFO and PIPE 2.0 interface to the core logic
Gbps Ethernet (GbE)	1.25, 3.125	The same as custom single- and double-width modes	The same as custom single- and double-width modes, plus the rate match FIFO
Serial Digital Interface (SDI)	0.27[33], 1.485, and 2.97	Phase compensation FIFO and byte serializer	Phase compensation FIFO and byte deserializer
SATA	1.5, 3.0, and 6.0	Phase compensation FIFO, byte serializer, and 8B/10B encoder	Phase compensation FIFO, byte deserializer, word aligner, and 8B/10B decoder
Common Public Radio Interface (CPRI)	0.6144, 1.2288, 2.4576, 3.072, 4.9152, 6.144, 9.8304[34]	The same as custom single- and double-width modes, plus the transmitter (TX) deterministic latency	The same as custom single- and double-width modes, plus the receiver (RX) deterministic latency

[33] The 0.27 gigabits per second (Gbps) data rate is supported using oversampling user logic that must be implemented by the user in the FPGA core.

PCS Support	Data Rates (Gbps)	Transmitter Datapath	Receiver Datapath
OBSAI	0.768, 1.536, 3.072, 6.144	The same as custom single- and double-width modes, plus the TX deterministic latency	The same as custom single- and double-width modes, plus the RX deterministic latency
Serial RapidIO® (SRIO)	1.25, 2.5, 3.125	The same as custom single- and double-width modes	The same as custom single- and double-width modes
XAUI	3.125	Implemented using soft PCS	Implemented using soft PCS
10GBASE-R	10.3125	Implemented using soft PCS	Implemented using soft PCS

Related Information

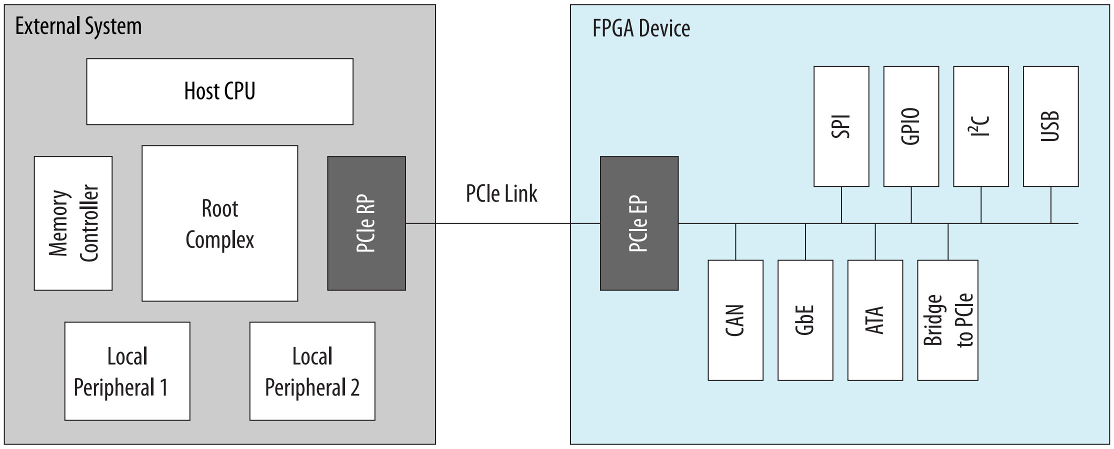
- **Arria V Device Handbook: Known Issues**
  Lists the planned updates to the *Arria V Device Handbook* chapters.
- **Use this chapter along with the Altera Transceiver PHY IP Core User Guide.**
- **Upcoming Arria V Device Features**

PCI Express

The Arria V devices have PCIe Hard IP that is designed for performance, ease-of-use, and increased functionality. The Hard IP consists of the media access control (MAC) lane, data link, and transaction layers. The PCIe Hard IP supports the PCIe Gen1 end point and root port up to x8 lane configurations. The PCIe endpoint support includes multifunction support for up to eight functions and Gen2 x4 lane configurations.

[34] The 9.8304 Gbps CPRI implementation (supported with 10-Gbps channels only) is implemented using PMA Direct mode. The PMA interfaces with the FPGA fabric directly, so you must implement the required PCS functionality in user logic (soft PCS).

Figure 4-1: PCIe Multifunction for Arria V Devices

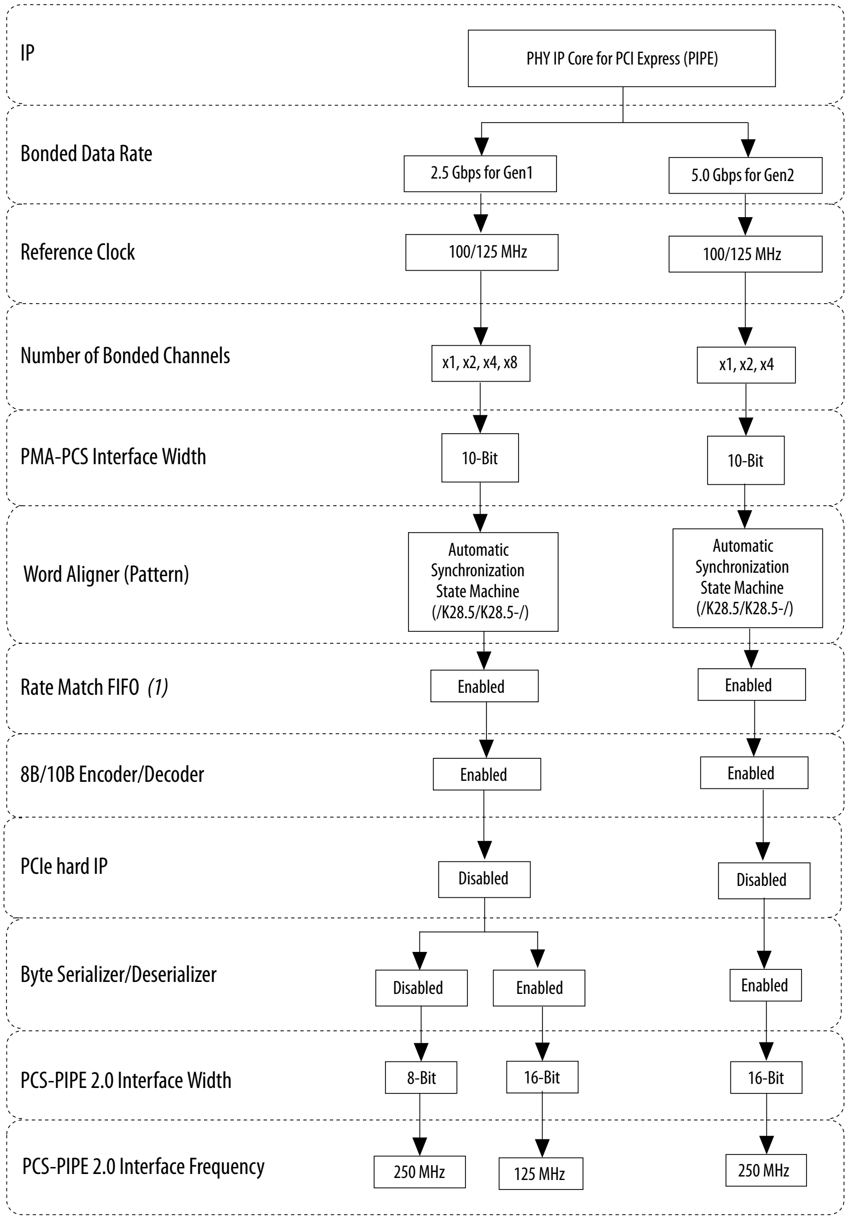


The Arria V PCIe Hard IP operates independently from the core logic, which allows the PCIe link to wake up and complete link training in less than 100 ms while the Arria V device completes loading the programming file for the rest of the device.

In addition, the Arria V device PCIe Hard IP has improved end-to-end datapath protection using error correction code (ECC).

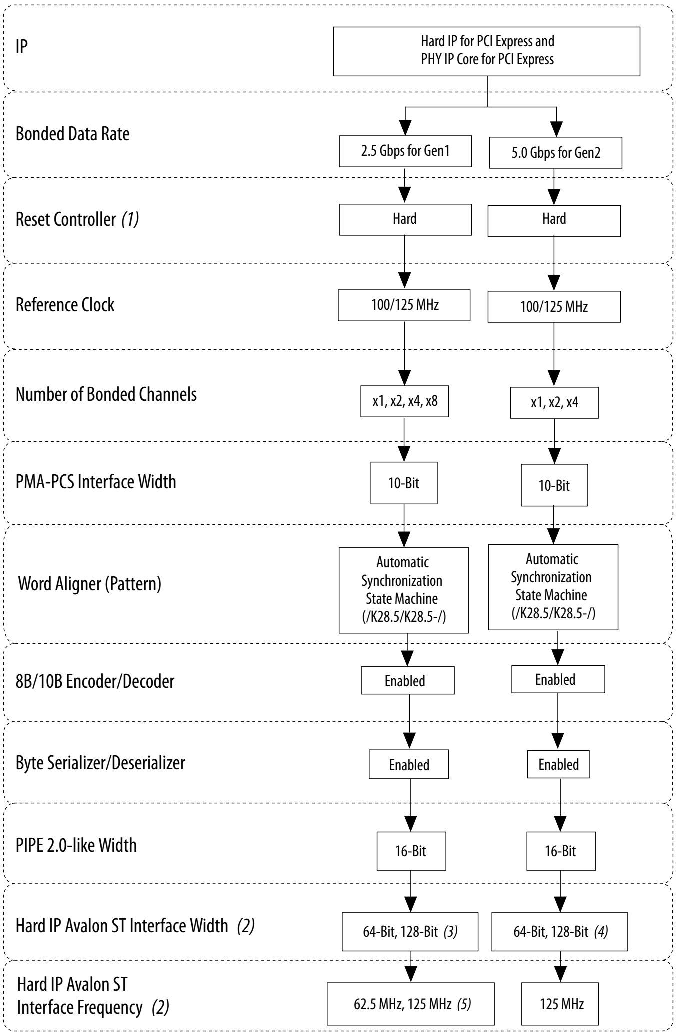
PCIe Transceiver Datapath

Figure 4-2: PCIe Gen1 and Gen2 PIPE Datapath Configuration



(1) When the PIPE low latency synchronous mode is enabled, the Rate Match FIFO operates in Low Latency mode.

Figure 4-3: PCIe Gen1 and Gen2 Hard IP and PHY IP Core for PCI Express Datapath Configuration



Notes:

(1) The PHY IP Core for PCI Express (PIPE configuration) employs the Embedded Reset Controller IP. It does not use the Hard or Soft Reset Controller employed in the Hard IP for PCI Express (HIP configuration).

(2) Does not apply to PHY IP Core for PCI Express configuration. Applies only to Hard IP for PCI Express configuration.

(3) 64-bit is for x1, x2, and x4 only. 128-bit is for x8 only.

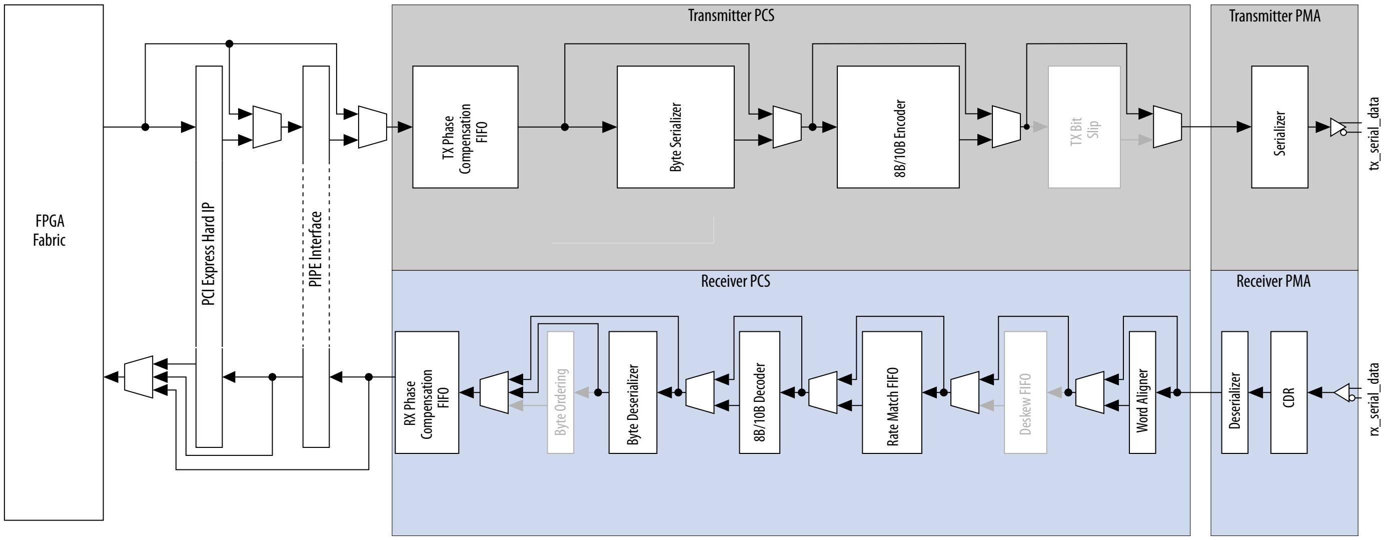
(4) 64-bit is for x1 and x2 only. 128-bit is for x4 only.

(5) 62.5 MHz interface frequency is for x1 with 64-bit Hard IP Avalon ST Interface Width only.

The transceiver datapath clocking varies between non-bonded (x1) and bonded (x2, x4, and x8) configurations.

Transceiver Channel Datapath

Figure 4-4: Transceiver Channel Datapath in a PIPE Configuration



Related Information

Transceiver Architecture in Arria V Devices

PCIe Supported Features

The PIPE configuration for the 2.5 Gbps (Gen1) and 5 Gbps (Gen2) data rates supports these features:

- PCIe-compliant synchronization state machine
- ±300 parts per million (ppm)—total 600 ppm—clock rate compensation
- 8-bit FPGA fabric–transceiver interface
- 16-bit FPGA fabric–transceiver interface
- Transmitter buffer electrical idle
- Receiver detection
- 8B/10B encoder disparity control when transmitting compliance pattern
- Power state management (Electrical Idle only)
- Receiver status encoding

PIPE Interface

In a PIPE configuration, each channel has a PIPE interface block that transfers data, control, and status signals between the PHY-MAC layer and the transceiver channel PCS and PMA blocks.

The PIPE interface block complies with version 2.0 of the PIPE specification. If you use the PIPE hard IP block, the PHY-MAC layer is implemented in the hard IP block. Otherwise, you can implement the PHY-MAC layer using soft IP in the FPGA fabric.

If you use the PIPE hard IP block, the PHY-MAC layer is implemented in the hard IP block. Otherwise, you can implement the PHY-MAC layer using soft IP in the FPGA fabric, which will be supported in future versions of the Quartus II software.

Note: The PIPE interface block is used in a PIPE configuration and cannot be bypassed.

In addition to transferring data, control, and status signals between the PHY-MAC layer and the transceiver, the PIPE interface block implements the following functions that are required in a PCIe-compliant physical layer device:

- Forces the transmitter buffer into an electrical idle state
- Initiates the receiver detect sequence
- Controls the 8B/10B encoder disparity when transmitting a compliance pattern
- Manages the PCIe power states (Electrical Idle only)
- Indicates the completion of various PHY functions, such as receiver detection and power state transitions on the `pipe_phystatus` signal
- Encodes the receiver status and error conditions on the `pipe_rxstatus[2:0]` signal, as specified in the PCIe specification

Transmitter Electrical Idle Generation

The PIPE interface block places the channel transmitter buffer in an electrical idle state when the electrical idle input signal is asserted.

During electrical idle, the transmitter buffer differential and common configuration output voltage levels are compliant to the PCIe Base Specification 2.1 for the PCIe Gen2 data rate.

The PCIe specification requires that the transmitter buffer be placed in electrical idle in certain power states.

Power State Management

The PCIe specification defines four power states: P0, P0s, P1, and P2.

The physical layer device must support these power states to minimize power consumption:

- P0 is the normal operating state during which packet data is transferred on the PCIe link.
- P0s, P1, and P2 are low-power states into which the physical layer must transition as directed by the PHY-MAC layer to minimize power consumption.

The PIPE interface in the transceivers provides an input port for each transceiver channel configured in a PIPE configuration.

Note: When transitioning from the P0 power state to lower power states (P0s, P1, and P2), the PCIe specification requires that the physical layer device implements power saving measures. The transceivers do not implement these power saving measures except to place the transmitter buffer in electrical idle in the lower power states.

8B/10B Encoder Usage for Compliance Pattern Transmission Support

The PCIe transmitter transmits a compliance pattern when the Link Training and Status State Machine (LTSSM) enters a polling compliance substate. The polling compliance substate assesses if the transmitter is electrically compliant with the PCIe voltage and timing specifications.

Receiver Status

The PCIe specification requires that the PHY encode the receiver status on a 3-bit status signal (`pipe_rxstatus[2:0]`).

This status signal is used by the PHY-MAC layer for its operation. The PIPE interface block receives the status signals from the transceiver channel PCS and PMA blocks, and encodes the status on the pipe_rxstatus[2:0] signal to the FPGA fabric. The encoding of the status signals on the pipe_rxstatus[2:0] signal is compliant with the PCIe specification.

Receiver Detection

The PIPE interface block in Arria V transceivers provides an input signal (pipe_txdetectrx_loopback) for the receiver detect operation that is required by the PCIe protocol during the detect substate of the LTSSM.

When the pipe_txdetectrx_loopback signal is asserted in the P1 power state, the PCIe interface block sends a command signal to the transmitter buffer in that channel to initiate a receiver detect sequence. In the P1 power state, the transmitter buffer must always be in the electrical idle state.

After receiving this command signal, the receiver detect circuitry creates a step voltage at the output of the transmitter buffer. If an active receiver that complies with the PCIe input impedance requirements is present at the far end, the time constant of the step voltage on the trace is higher than if the receiver is not present. The receiver detect circuitry monitors the time constant of the step signal that is seen on the trace to determine if a receiver was detected. The receiver detect circuitry monitor requires a 125-MHz clock for operation that you must drive on the fixedclk port.

Note: For the receiver detect circuitry to function reliably, the AC-coupling capacitor on the serial link and the receiver termination values used in your system must be compliant with the PCIe Base Specification 2.1.

The PCI Express PHY (PIPE) IP core provides a 1-bit PHY status (pipe_phystatus) and a 3-bit receiver status signal (pipe_rxstatus[2:0]) to indicate whether a receiver was detected or not, in accordance to the PIPE 2.0 specifications.

Clock Rate Compensation Up to ±300 ppm

In compliance with the PCIe protocol, the receiver channels are equipped with a rate match FIFO to compensate for the small clock frequency differences of up to ± 300 ppm between the upstream transmitter and local receiver clocks.

Related Information

Transceiver Architecture in Arria V Devices

PCIe Reverse Parallel Loopback

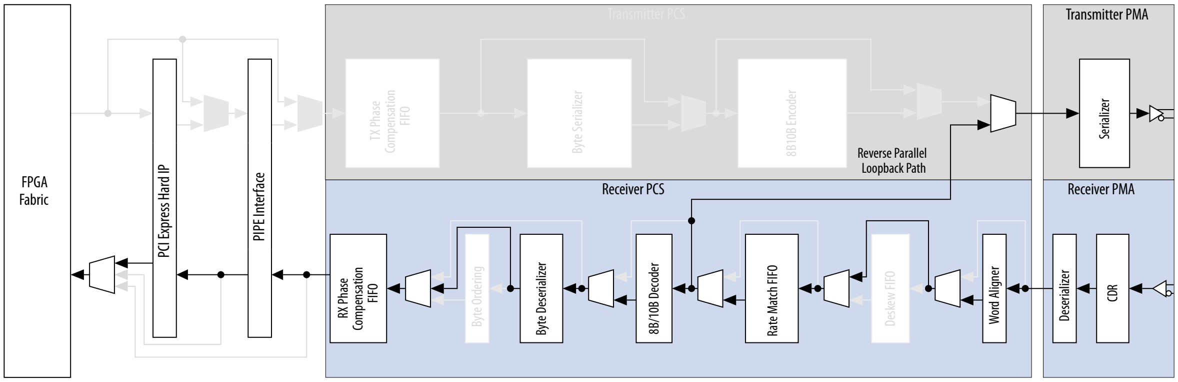
The PCIe reverse parallel loopback is only available in the PCIe functional configuration for the Gen1 data rate. The received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. It is then looped back to the transmitter serializer and transmitted out through the transmitter buffer. The received data is also available to the FPGA fabric through the port.

This loopback mode is compliant with the PCIe specification 2.1.

Arria V devices provide the pipe_txdetectrx_loopback input signal to enable this loopback mode. If the pipe_txdetectrx_loopback signal is asserted in the P1 power state, receiver detection is performed. If the signal is asserted in the P0 power state, reverse parallel loopback is performed.

Note: The PCIe reverse parallel loopback is the only loopback option that is supported in PIPE configurations.

Figure 4-5: PIPE Reverse Parallel Loopback Mode Datapath



PIPE Transceiver Channel Placement Guidelines

Table 4-2: PIPE Channel Placement for PCIe Gen1

Placement by the Quartus II software may vary with design, resulting in higher channel utilization.

Configuration	Data Channel Placement	Minimum Channel Utilization	Default Logical Data Channel Number for Master
x1	Any channel	2 (1 data channel, 1 clock channel)	Data_channel[0]
x2	2 contiguous channels	3 (2 data channels, 1 clock channel)	Data_channel[1]
x4	4 contiguous channels	5 (4 data channels, 1 clock channel)	Data_channel[1]
x8	8 contiguous channels	9 (8 data channels, 1 clock channel)	Data_channel[0]

To override the default data channel number for the Master channel, do following:

1. Assign the Master channel to the same bank of CMU PLL.
2. Apply the following Quartus II QSF assignment:

```
set_parameter -name master_ch_number <logical_data_channel_number>
-to <"test:pcie_i|altera_xcvr_pipe:test_inst|av_xcvr_pipe_nr:pipe_nr_inst|
av_xcvr_pipe_native:transceiver_core">
```

To support PIPE placement identical to PCIe HIP x8, use following two Quartus II QSF assignments:

```
set_parameter -name master_ch_number 4 -to <"test:pcie_i|altera_xcvr_pipe:test_inst|
av_xcvr_pipe_nr:pipe_nr_inst|
av_xcvr_pipe_native:transceiver_core”>


set_parameter -name dummy_ch_required 1 -to <"test:pcie_i|
altera_xcvr_pipe:test_inst|av_xcvr_pipe_nr:pipe_nr_inst|
av_xcvr_pipe_native:transceiver_core">
```

Note: For more information about the hard IP implementation of PCIe and restrictions, refer to the "Transceiver Banks" section of the *Transceiver Architecture in Arria V Devices* chapter.

The following four figures show examples of channel placement for PIPE x1, x2, x4, and x8 configurations.

Figure 4-6: Example of PIPE x1 Channel Placement

Channels shaded in blue provide the high-speed serial clock. Channels shaded in gray are data channels. You can place the PIPE data channels in any available channel in the transceiver bank.

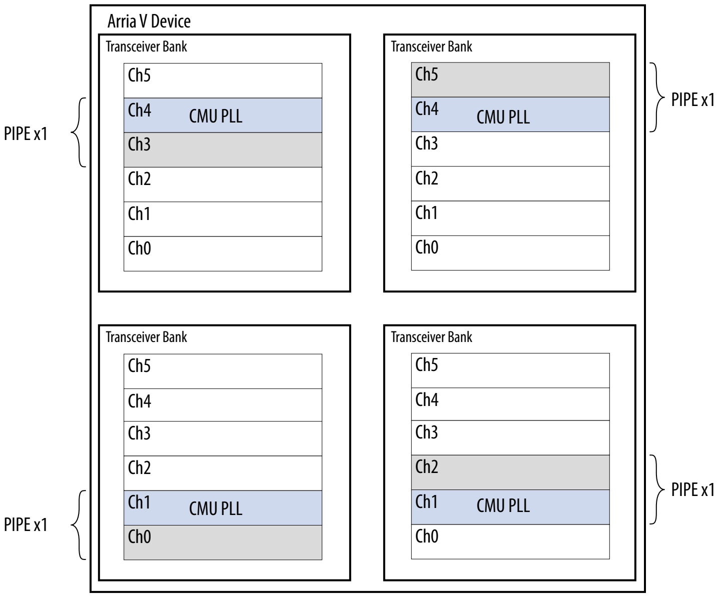


Figure 4-7: Example of PIPE x2 Channel Placement

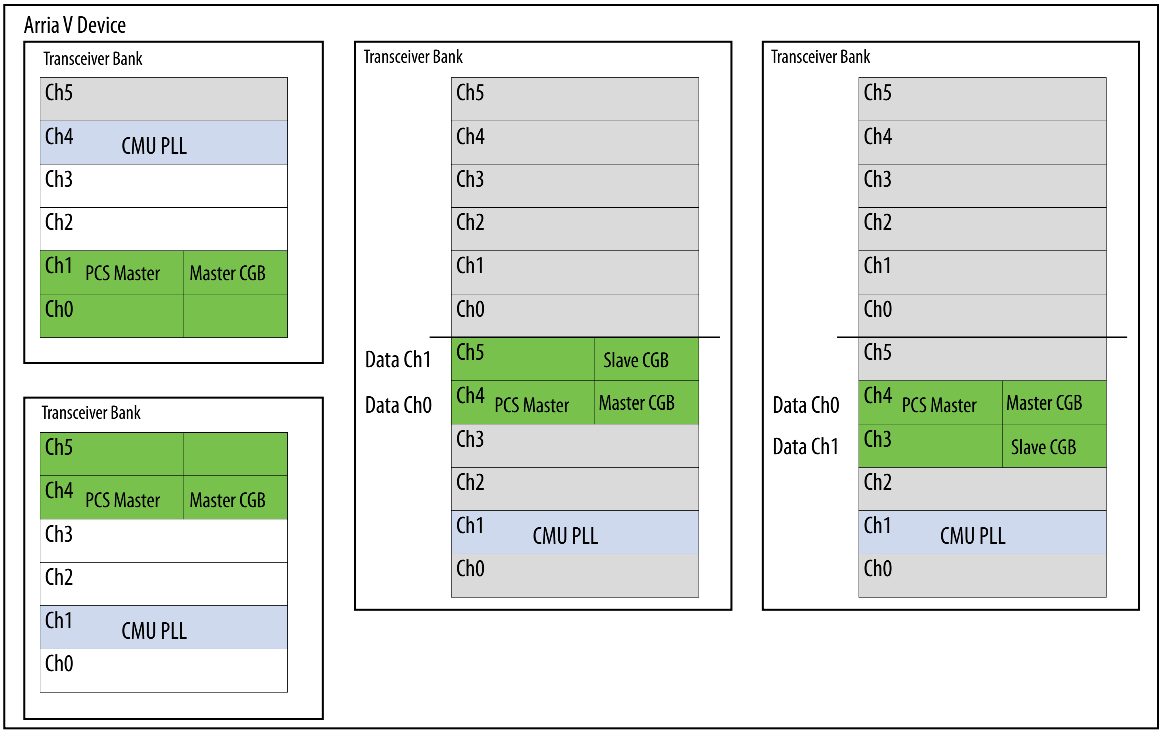


Figure 4-8: Example of PIPE x4 Channel Placement

Channels shaded in blue provide the high-speed serial clock. Channels shaded in gray are data channels.

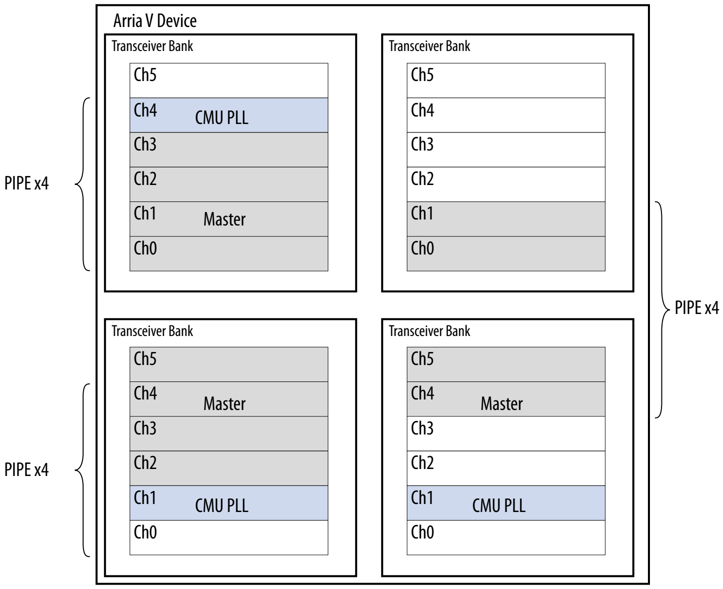
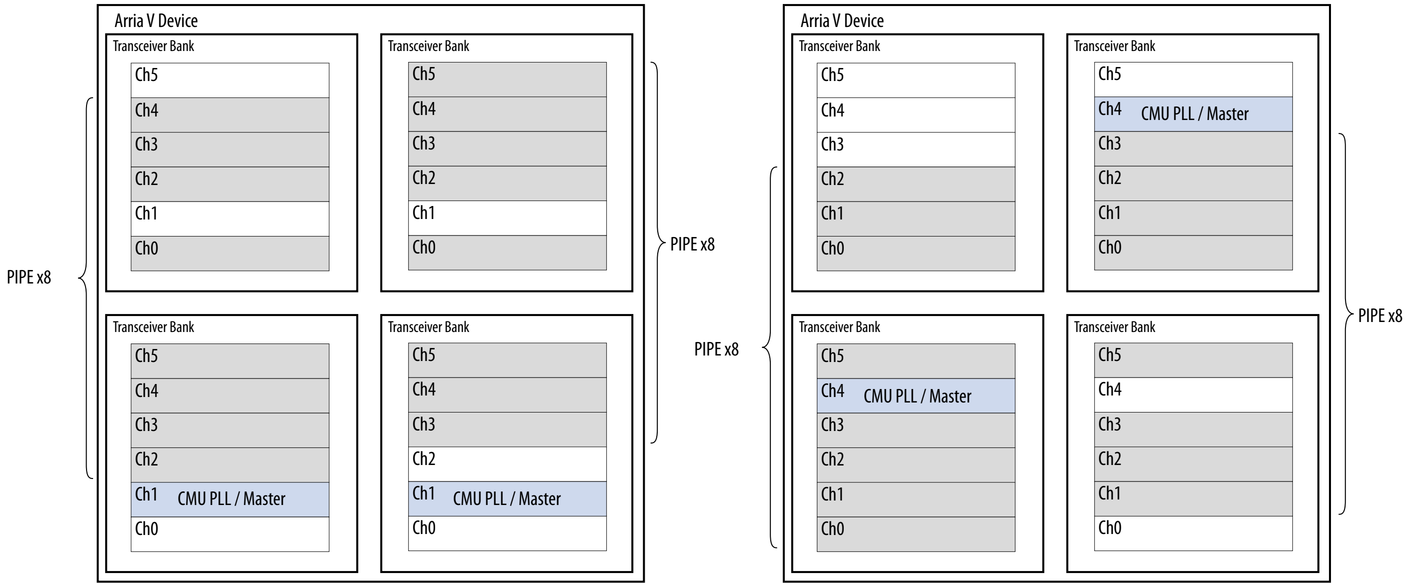


Figure 4-9: Example of PIPE x8 Channel Placement

Channels shaded in blue provide the high-speed serial clock. Channels shaded in gray are data channels.



Related Information

Transceiver Architecture in Arria V Devices

PCIe Supported Configurations and Placement Guidelines

Placement by the Quartus II software may vary with design and device. The following figures show examples of transceiver channel and PCIe Hard IP block locations, supported x1, x2, x4, and x8 bonding configurations, and channel placement guidelines. The Quartus II software automatically places the CMU PLL in a channel different from that of the data channels.

Note: This section shows the supported PCIe channel placement if you use both the top and bottom PCIe Hard IP blocks in the device separately.

The following guidelines apply to all channel placements:

- The CMU PLL requires its own channel and must be placed on channel 1 or channel 4
- The PCIe channels must be contiguous within the transceiver bank
- Lane 0 of the PCIe must be placed on channel 0 or channel 5

In the following figures, channels shaded in blue provide the high-speed serial clock. Channels shaded in gray are data channels.

Figure 4-10: PCIe HIP Supported x1 Guidelines

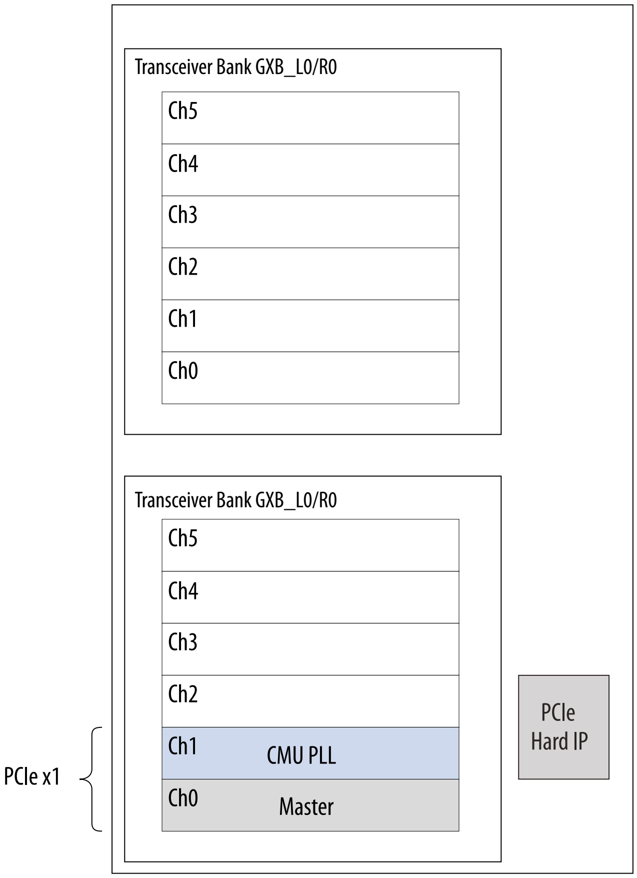


Figure 4-11: PCIe HIP Supported x2 and x4 Guidelines

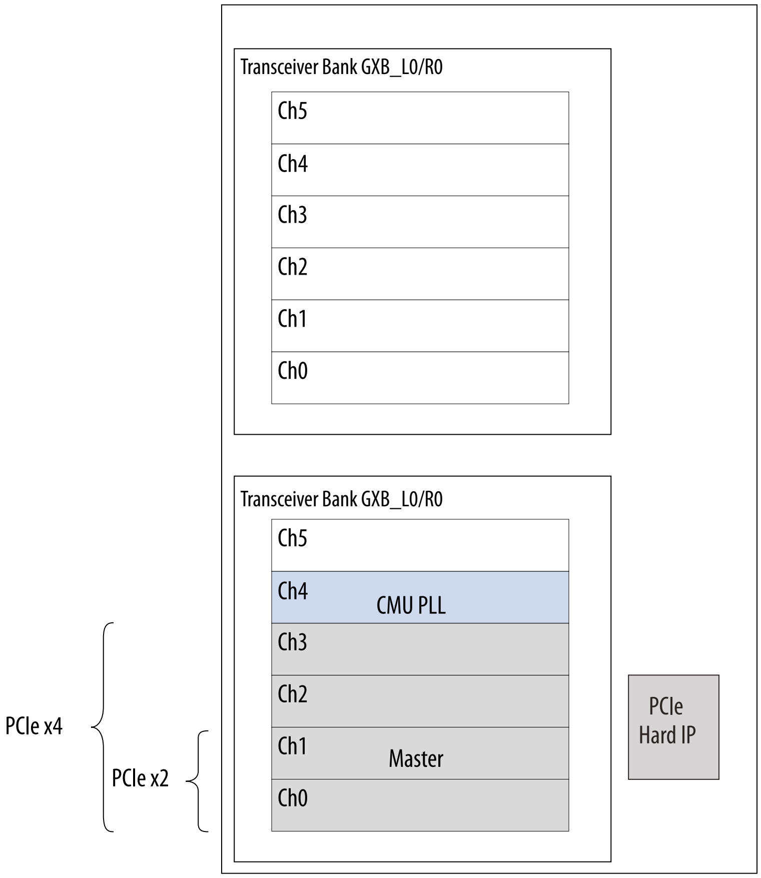
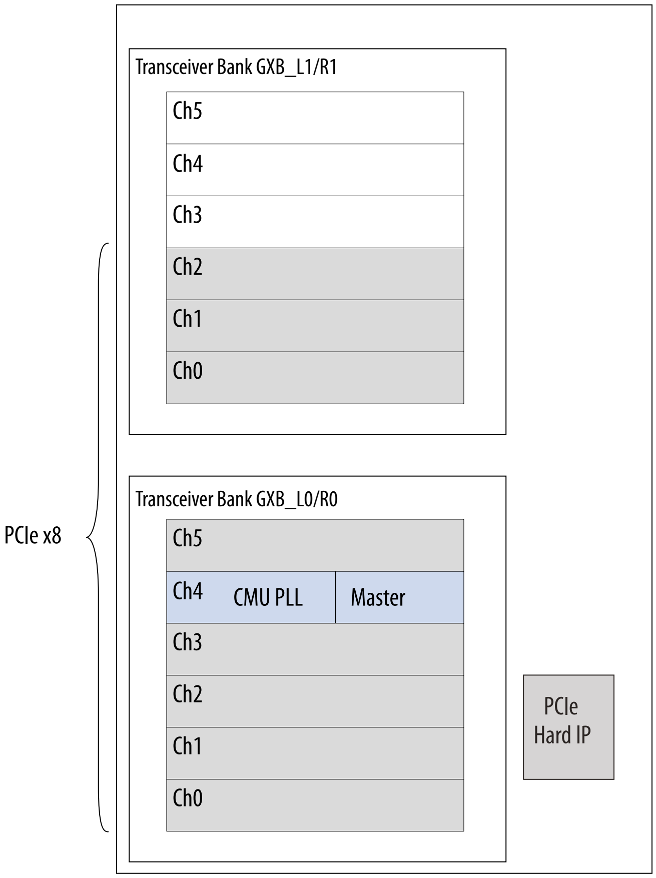


Figure 4-12: PCIe HIP Supported x8 Guidelines



For PCIe Gen1 and Gen2, there are restrictions on the achievable x1 and x4 bonding configurations if you intend to use both top and bottom Hard IP blocks in the device.

Related Information

Transceiver Architecture in Arria V Devices

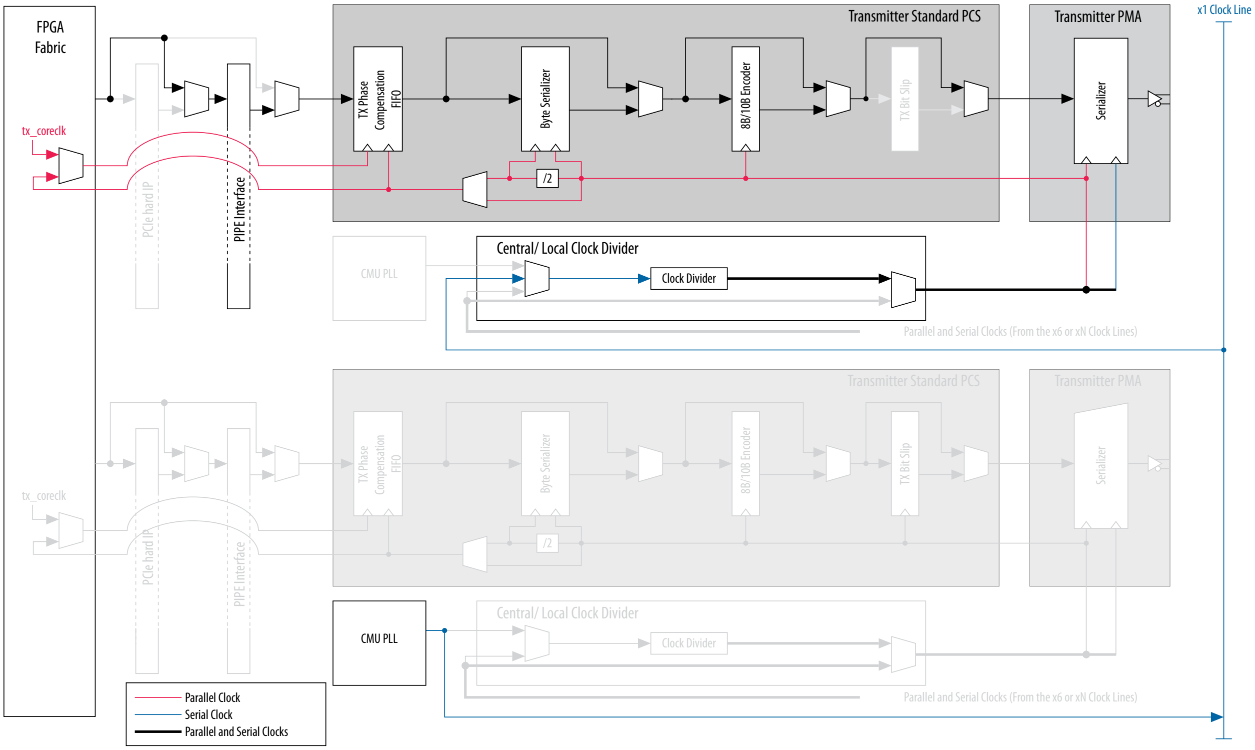
PIPE Transceiver Clocking

This section describes transceiver clocking for PIPE configurations.

PIPE x1 Configuration

The serial clock in the transceiver clocking configuration is provided by the CMU PLL in a channel different from that of the data channel. The local clock divider block in the data channel generates a parallel clock from this high-speed clock and distributes both clocks to the PMA and PCS of the data channel.

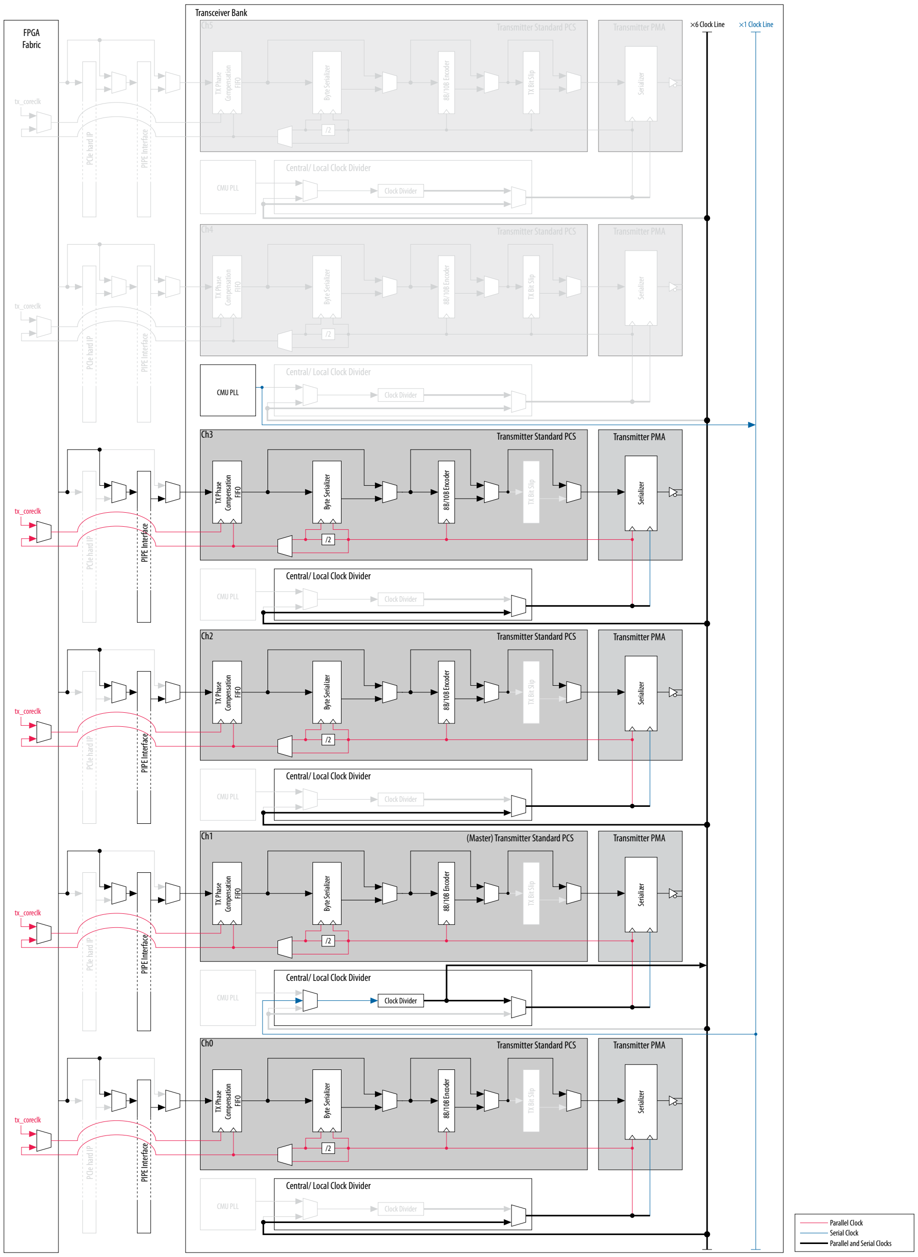
Figure 4-13: Transceiver Clocking Configuration in a PIPE x1 Configuration



PIPE x4 Configuration

In a PIPE x4 bonded configuration, clocking is independent for the receiver channels. The clocking and control signals are bonded only for the transmitter channels.

Figure 4-14: Transceiver Clocking Configuration in a PIPE x4 Configuration

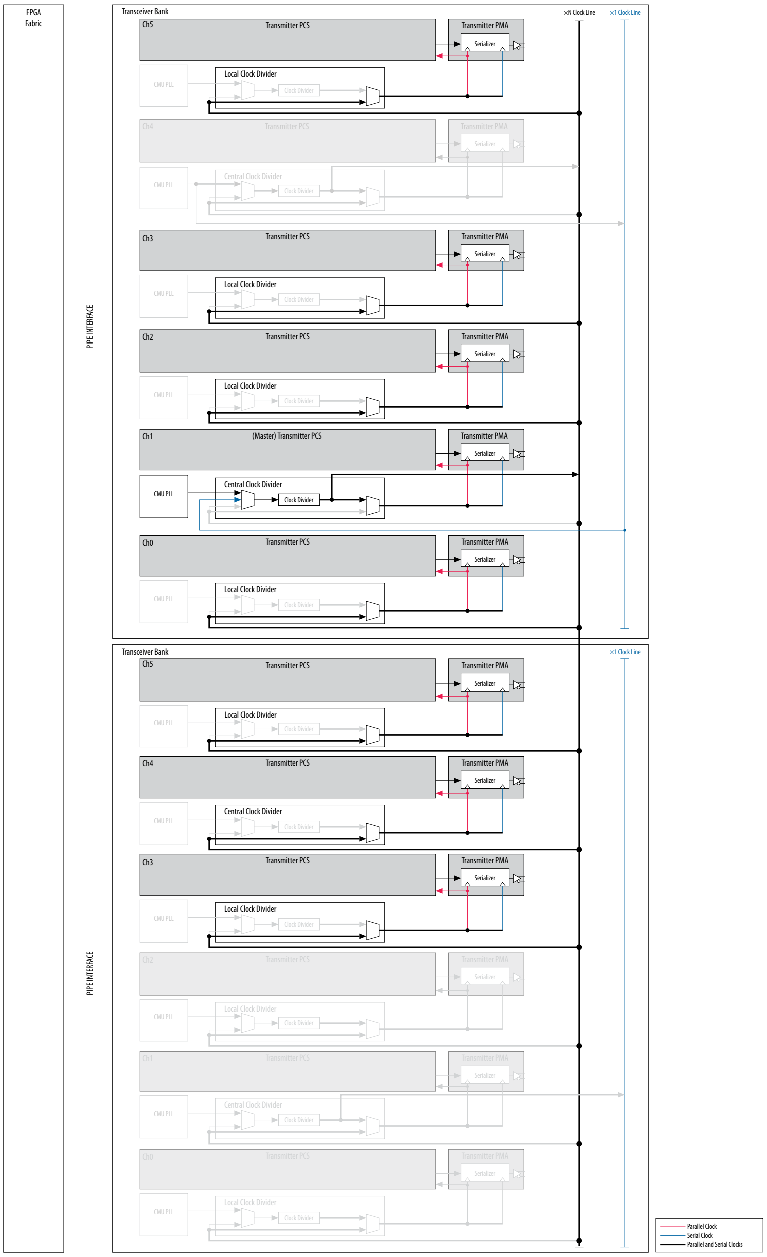


PIPE x8 Configuration

In a PIPE x8 bonded configuration, the clocking for the PMA and PCS blocks is independent for the receiver channels. The clocking and control signals are bonded only for the transmitter channels.

For more information about clocking in Arria V devices, refer to the *Transceiver Clocking in Arria V Devices* chapter.

Figure 4-15: Transceiver Clocking Configuration in a PIPE x8 Configuration



Related Information

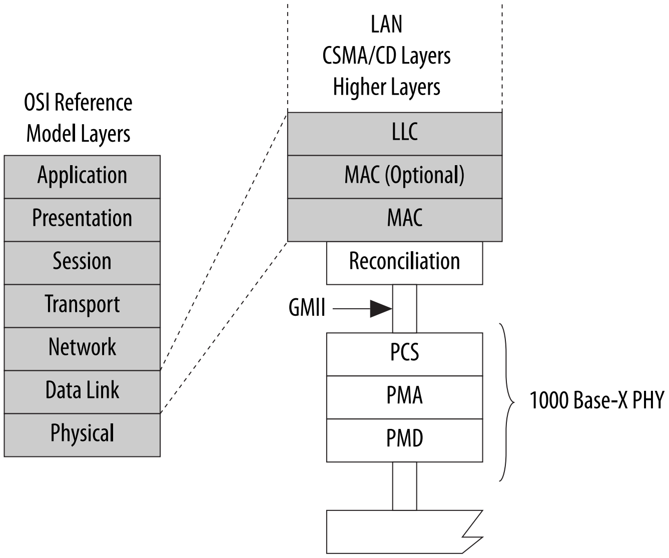
- "PCI Express PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide
- Transceiver Clocking in Arria V Devices

Gigabit Ethernet

The IEEE 802.3 specification defines the 1000BASE-X PHY as an intermediate, or transition layer that interfaces various physical media with the MAC in a gigabit ethernet (GbE) system, shielding the MAC layer from the specific nature of the underlying medium. The 1000BASE-X PHY is divided into the PCS, PMA, and PMD sublayers.

The PCS sublayer interfaces with the MAC through the gigabit media independent interface (GMII). The 1000BASE-X PHY defines a physical interface data rate of 1 Gbps and 2.5 Gbps.

Figure 4-16: 1000BASE-X PHY in a GbE OSI Reference Model



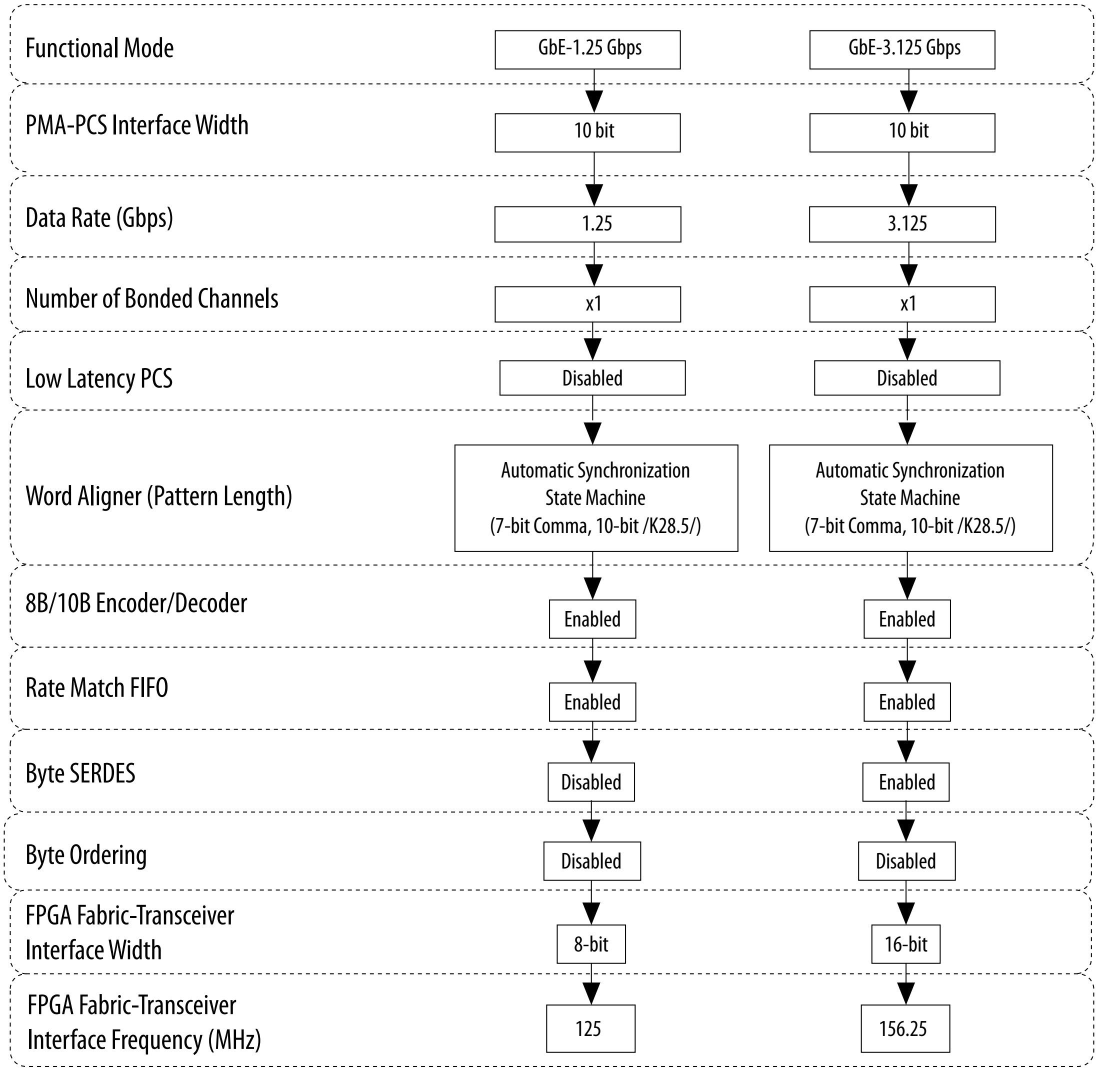
The transceivers, when configured in GbE functional mode, have built-in circuitry to support the following PCS and PMA functions, as defined in the IEEE 802.3 specification:

- 8B/10B encoding and decoding
- Synchronization
- Clock recovery from the encoded data forwarded by the receiver PMD
- Serialization and deserialization

Note: If you enabled the autonegotiation state machine in the FPGA core with the rate match FIFO, refer to the "Rate Match FIFO" section in the "Gigabit Ethernet Transceiver Datapath" section.

Note: The transceivers do not have built-in support for other PCS functions, such as the autonegotiation state machine, collision-detect, and carrier-sense functions. If you require these functions, implement them in the FPGA fabric or in external circuits.

Figure 4-17: Transceiver Blocks in a GbE Configuration



Functional Mode	GbE-1.25 Gbps	GbE-3.125 Gbps
PMA-PCS Interface Width	10 bit	10 bit
Data Rate (Gbps)	1.25	3.125
Number of Bonded Channels	x1	x1
Low Latency PCS	Disabled	Disabled
Word Aligner (Pattern Length)	Automatic Synchronization State Machine (7-bit Comma, 10-bit /K28.5/)	Automatic Synchronization State Machine (7-bit Comma, 10-bit /K28.5/)
8B/10B Encoder/Decoder	Enabled	Enabled
Rate Match FIFO	Enabled	Enabled
Byte SERDES	Disabled	Enabled
Byte Ordering	Disabled	Disabled
FPGA Fabric-Transceiver Interface Width	8-bit	16-bit
FPGA Fabric-Transceiver Interface Frequency (MHz)	125	156.25

Related Information

Gigabit Ethernet Transceiver Datapath on page 4-21

Refer to the "Rate Match FIFO" section.

Gigabit Ethernet Transceiver Datapath

Figure 4-18: Transceiver Datapath in GbE-1.25 Gbps Configuration

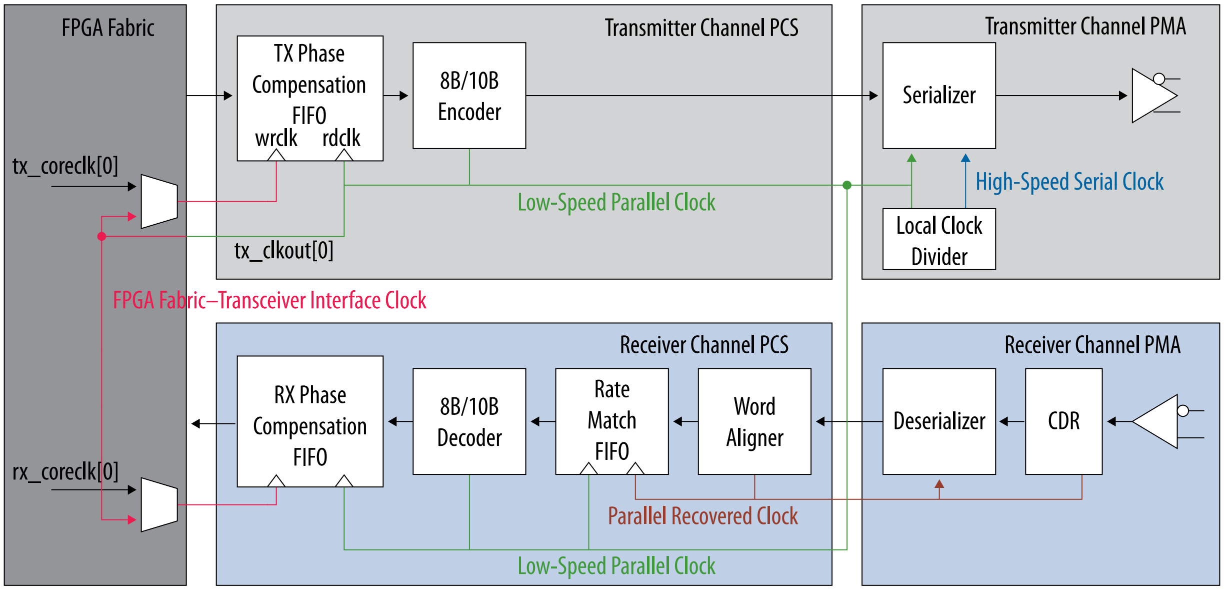


Figure 4-19: Transceiver Datapath in GbE-3.125 Gbps Configuration

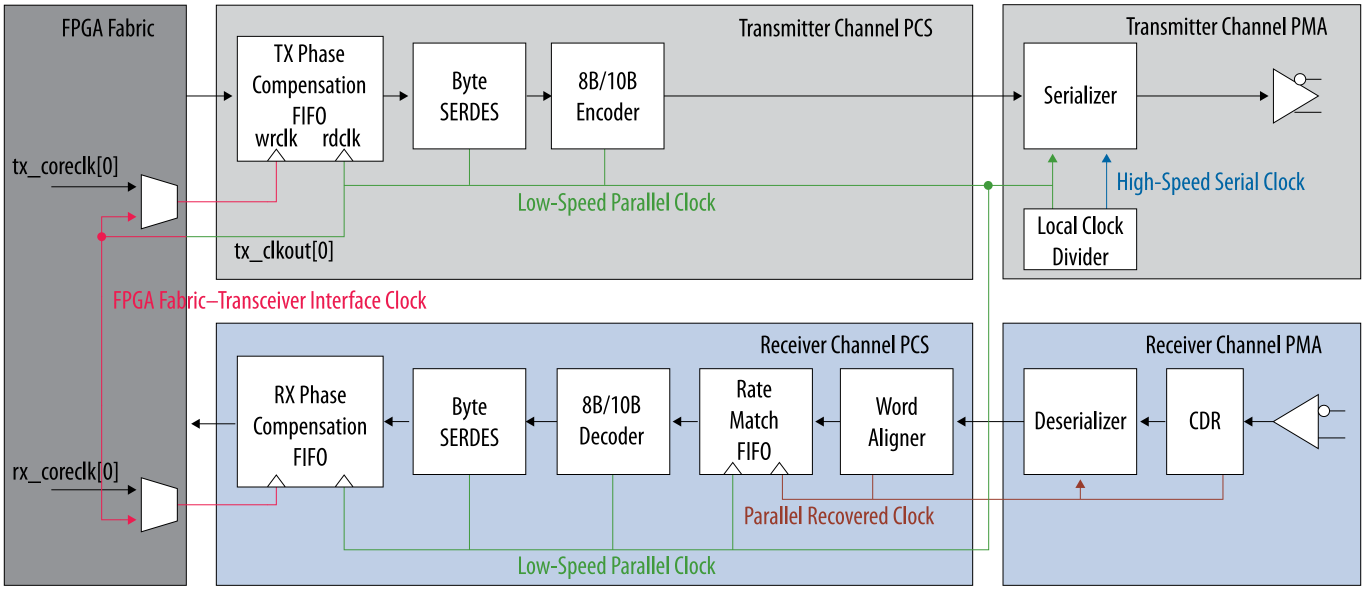


Table 4-3: Transceiver Datapath Clock Frequencies in GbE Configuration

Functional Mode	Data Rate	High-Speed Serial Clock Frequency	Parallel Recovered Clock and Low-Speed Parallel Clock Frequency	FPGA Fabric-Transceiver Interface Clock Frequency
GbE-1.25 Gbps	1.25 Gbps	625 MHz	125 MHz	125 MHz
GbE-3.125 Gbps	3.125 Gbps	1562.5 MHz	312.5 MHz	156.25 MHz

8B/10B Encoder

In GbE configuration, the 8B/10B encoder clocks in 8-bit data and 1-bit control identifiers from the transmitter phase compensation FIFO and generates 10-bit encoded data. The 10-bit encoded data is fed to the serializer.

For more information about the 8B/10B encoder functionality, refer to the **Transceiver Architecture for Arria V Devices** chapter.

Rate Match FIFO

In GbE configuration, the rate match FIFO is capable of compensating for up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The GbE protocol requires that the transmitter send idle ordered sets /I1/ (/K28.5/D5.6/) and /I2/ (/K28.5/D16.2/) during interpacket gaps, adhering to the rules listed in the IEEE 802.3 specification.

The rate match operation begins after the synchronization state machine in the word aligner indicates that the synchronization is acquired-by driving the `rx_syncstatus` signal high. The rate matcher always deletes or inserts both symbols (/K28.5/ and /D16.2/) of the /I2/ ordered sets, even if only one symbol needs to be deleted to prevent the rate match FIFO from overflowing or underrunning. The rate matcher can insert or delete as many /I2/ ordered sets as necessary to perform the rate match operation.

Two flags are forwarded to the FPGA fabric:

- `rx_rmfifodatadeleted`—Asserted for two clock cycles for each deleted /I2/ ordered set to indicate the rate match FIFO deletion event
- `rx_rmfifodatainserted`—Asserted for two clock cycles for each inserted /I2/ ordered set to indicate the rate match FIFO insertion event

Note: If you have the autonegotiation state machine in the FPGA, note that the rate match FIFO is capable of inserting or deleting the first two bytes (/K28.5//D2.2/) of /C2/ ordered sets during autonegotiation. However, the insertion or deletion of the first two bytes of /C2/ ordered sets can cause the autonegotiation link to fail. For more information, refer to the **Altera Knowledge Base Support Solution**.

For more information about the rate match FIFO, refer to the **Transceiver Architecture for Arria V Devices** chapter.

GbE Protocol-Ordered Sets and Special Code Groups

Table 4-4: GIGE Ordered Sets

The following ordered sets and special code groups are specified in the IEEE 802.3-2008 specification.

Code	Ordered Set	Number of Code Groups	Encoding
/C/	**Configuration**	—	Alternating /C1/ and /C2/
/C1/	Configuration 1	4	/K28.5/D21.5/ `Config_Reg` [35]
/C2/	Configuration 2	4	/K28.5/D2.2/ `Config_Reg` [35]
/I/	**IDLE**	—	Correcting /I1/, Preserving / I2/
/I1/	IDLE 1	2	/K28.5/D5.>6/
/I2/	IDLE 2	2	/K28.5/D16.2/
-	**Encapsulation**	—	—
/R/	`Carrier_Extend`	1	/K23.7/
/S/	`Start_of_Packet`	1	/K27.7/
/T/	`End_of_Packet`	1	/K29.7/
/V/	`Error_Propagation`	1	/K30.7/

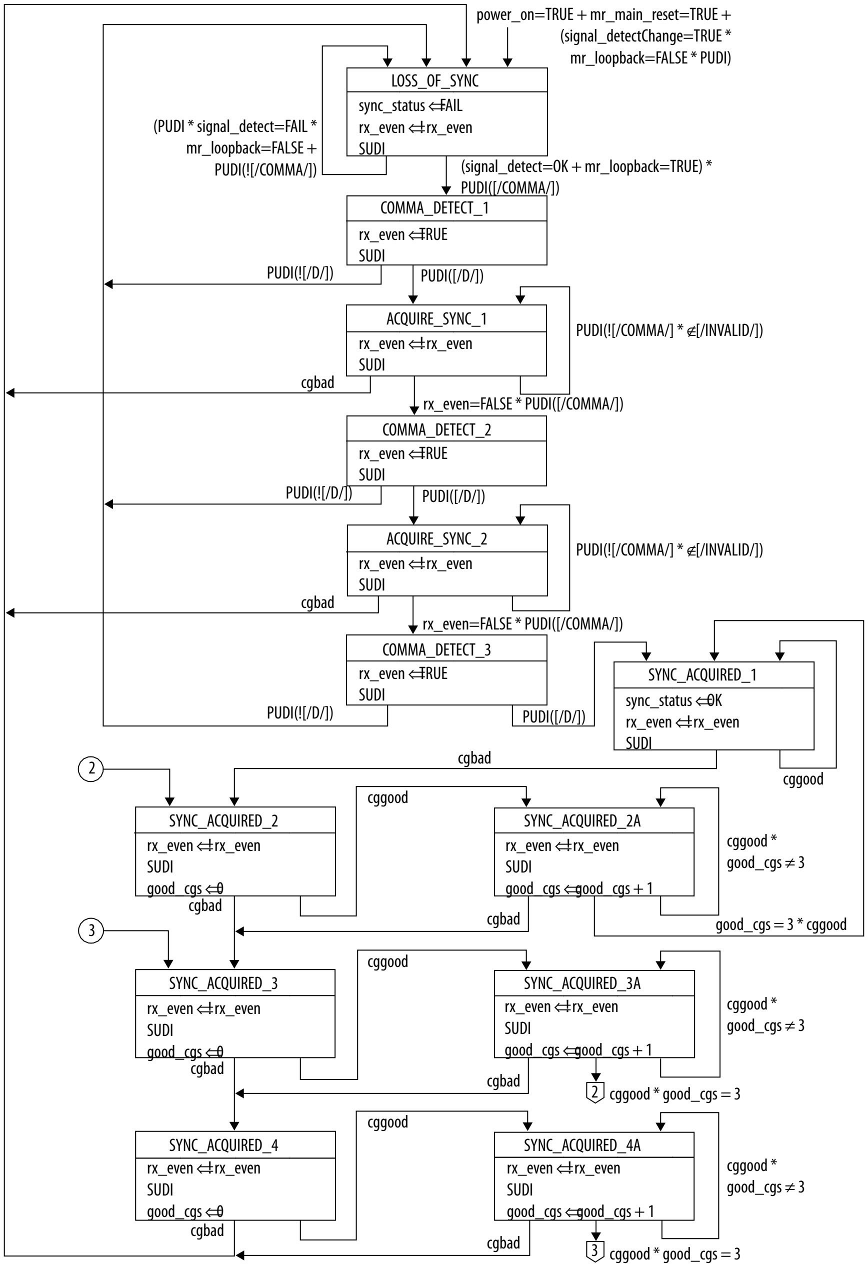
Table 4-5: Synchronization State Machine Parameters in GbE Mode

Synchronization State Machine Parameters	Setting
Number of valid {/K28.5/, /Dx,y/} ordered sets received to achieve synchronization	3
Number of errors received to lose synchronization	4
Number of continuous good code groups received to reduce the error count by 1	4

[35] Two data code groups represent the `Config_Reg` value.

Figure 4-20: Synchronization State Machine in GbE Mode

This figure is from “Figure 36–9” in the IEEE 802.3-2008 specification. For more details about the 1000BASE-X implementation, refer to Clause 36 of the IEEE 802.3-2008 specification.



Related Information

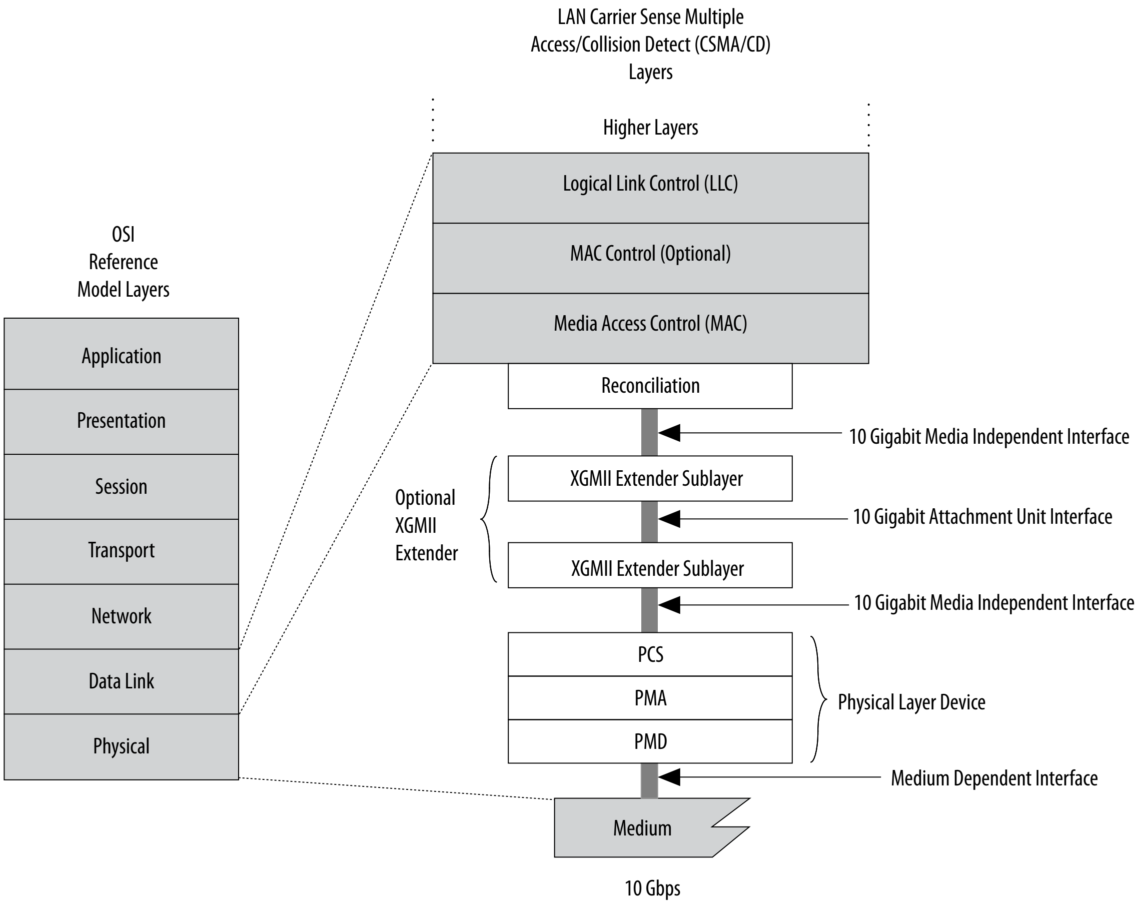
Refer to the "Custom PHY IP Core" and "Native PHY IP Core" chapters in the Altera Transceiver PHY IP Core User Guide

XAUI

In a XAUI configuration, the transceiver channel data path is configured using soft PCS. It provides the transceiver channel datapath description, clocking, and channel placement guidelines. To implement a XAUI link, instantiate the XAUI PHY IP core in the IP Catalog, which is under Ethernet in the Interfaces menu. The XAUI PHY IP core implements the XAUI PCS in soft logic.

XAUI is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in the IEEE 802.3ae-2002 specification. The XAUI PHY uses the XGMII interface to connect to the IEEE802.3 MAC and Reconciliation Sublayer (RS). The IEEE 802.3ae-2002 specification requires the XAUI PHY link to support a 10 Gbps data rate at the XGMII interface and four lanes each at 3.125 Gbps at the PMD interface.

Figure 4-21: XAUI and XGMII Layers


LAN Carrier Sense Multiple Access/Collision Detect (CSMA/CD) Layers
Higher Layers
Logical Link Control (LLC)
MAC Control (Optional)
Media Access Control (MAC)
Reconciliation
10 Gigabit Media Independent Interface
XGMII Extender Sublayer
10 Gigabit Attachment Unit Interface
XGMII Extender Sublayer
10 Gigabit Media Independent Interface
Optional XGMII Extender
PCS
PMA
PMD
Physical Layer Device
Medium Dependent Interface
Medium
10 Gbps
OSI Reference Model Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical

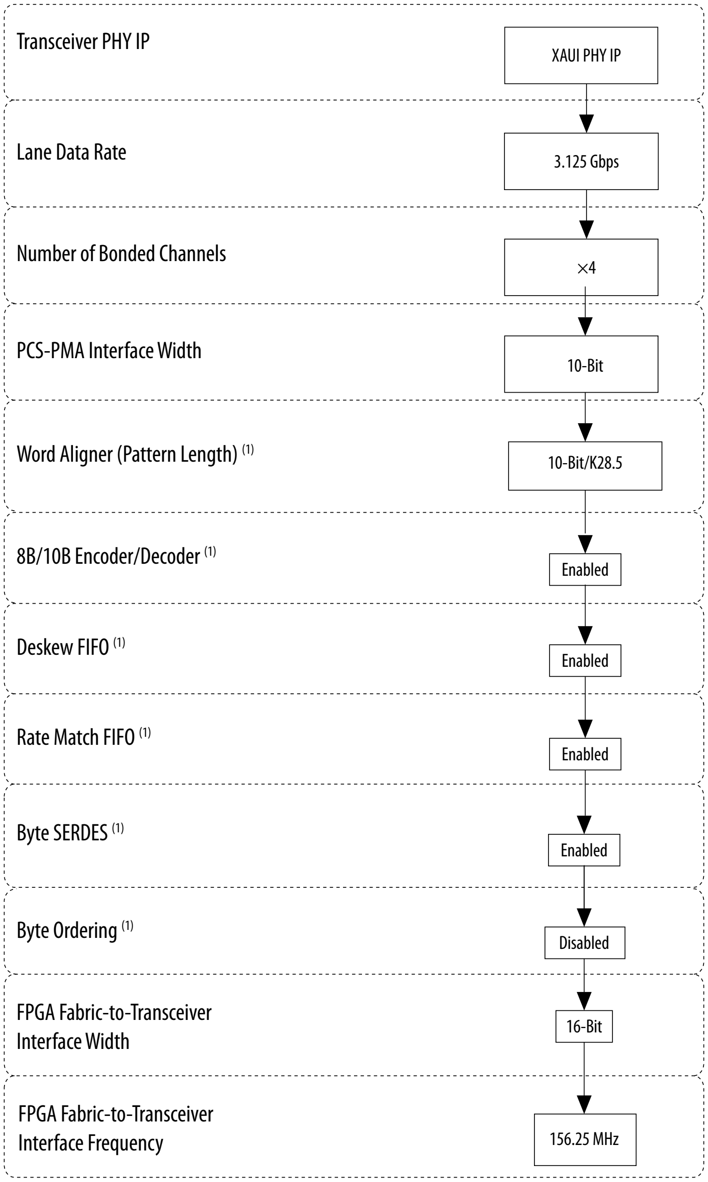
Related Information

Refer to the "XAUI PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide.

Transceiver Datapath in a XAUI Configuration

The XAUI PCS is implemented in soft logic inside the FPGA core when using the XAUI PHY IP core. You must ensure that your channel placement is compatible with the soft PCS implementation.

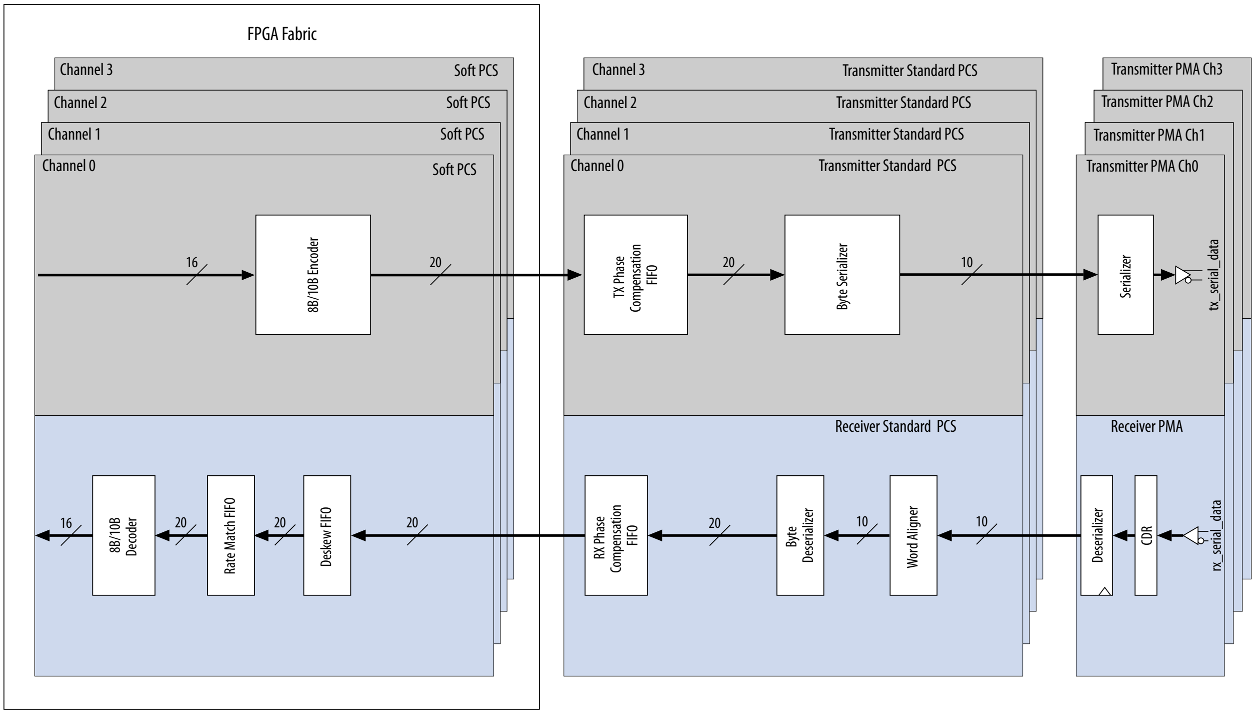
Figure 4-22: XAUI Configuration Datapath



Note:
1. Implemented in soft logic.

Figure 4-23: Transceiver Channel Datapath for XAUI Configuration

Standard PCS in a low latency configuration is used in this configuration. Additionally, a portion of the PCS is implemented in soft logic.



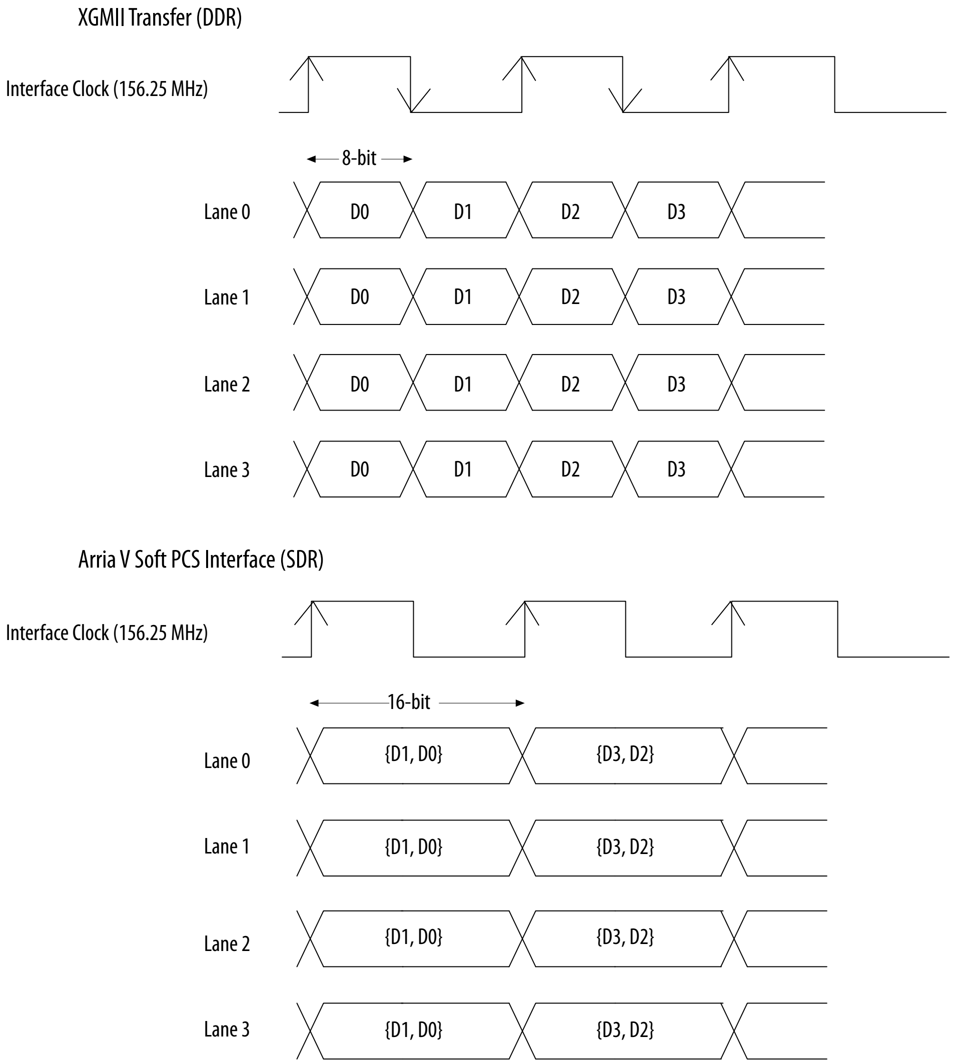
XAUI Supported Features

64-Bit SDR Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the XAUI PCS and the Ethernet MAC/RS. The specification requires each of the four XAUI lanes to transfer 8-bit data and 1-bit wide control code at both the positive and negative edge (DDR) of the 156.25 MHz interface clock.

Arria V transceivers and soft PCS solution in a XAUI configuration do not support the XGMII interface to the MAC/RS as defined in IEEE 802.3-2008 specification. Instead, they allow the transferring of 16-bit data and 2-bit control code on each of the four XAUI lanes, only at the positive edge (SDR) of the 156.25 MHz interface clock

Figure 4-24: Implementation of the XGMII Specification in Arria V Devices Configuration


XGMII Transfer (DDR)
Interface Clock (156.25 MHz)
8-bit
Lane 0
Lane 1
Lane 2
Lane 3
D0
D1
D2
D3
Arria V Soft PCS Interface (SDR)
Interface Clock (156.25 MHz)
16-bit
{D1, D0}
{D3, D2}

8B/10B Encoding/Decoding

Each of the four lanes in a XAUI configuration support an independent 8B/10B encoder/decoder as specified in Clause 48 of the IEEE802.3-2008 specification. 8B/10B encoding limits the maximum number of consecutive 1s and 0s in the serial data stream to five, thereby ensuring DC balance as well as enough transitions for the receiver CDR to maintain a lock to the incoming data.

The XAUI PHY IP core provides status signals to indicate running disparity as well as the 8B/10B code group error.

Transmitter and Receiver State Machines

In a XAUI configuration, the Arria V soft PCS implements the transmitter and receiver state diagrams shown in Figure 48-6 and Figure 48-9 of the IEEE802.3-2008 specification.

In addition to encoding the XGMII data to PCS code groups, in conformance with the 10GBASE-X PCS, the transmitter state diagram performs functions such as converting Idle ||I|| ordered sets into Sync ||K||, Align ||A||, and Skip ||R|| ordered sets.

In addition to decoding the PCS code groups to XGMII data, in conformance with the 10GBASE-X PCS, the receiver state diagram performs functions such as converting Sync ||K||, Align ||A||, and Skip ||R|| ordered sets to Idle ||I|| ordered sets.

Synchronization

The word aligner block in the receiver PCS of each of the four XAUI lanes implements the receiver synchronization state diagram shown in Figure 48-7 of the IEEE802.3-2008 specification.

The XAUI PHY IP core provides a status signal per lane to indicate if the word aligner is synchronized to a valid word boundary.

Deskew

The lane aligner block in the receiver PCS implements the receiver deskew state diagram shown in Figure 48-8 of the IEEE 802.3-2008 specification.

The lane aligner starts the deskew process only after the word aligner block in each of the four XAUI lanes indicates successful synchronization to a valid word boundary.

The XAUI PHY IP core provides a status signal to indicate successful lane deskew in the receiver PCS.

Clock Compensation

The rate match FIFO in the receiver PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. It does so by inserting and deleting Skip ||R|| columns, depending on the ppm difference.

The clock compensation operation begins after:

- The word aligner in all four XAUI lanes indicates successful synchronization to a valid word boundary.
- The lane aligner indicates a successful lane deskew.

The rate match FIFO provides status signals to indicate the insertion and deletion of the Skip ||R|| column for clock rate compensation.

Transceiver Clocking and Channel Placement Guidelines in XAUI Configuration

Transceiver Clocking

Figure 4-25: Transceiver Clocking for XAUI Configuration

One of the two channel PLLs configured as a CMU PLL in a transceiver bank generates the transmitter serial and parallel clocks for the four XAUI channels. The x6 clock line carries the transmitter serial and parallel clocks to the PMA and PCS of each of the four channels.

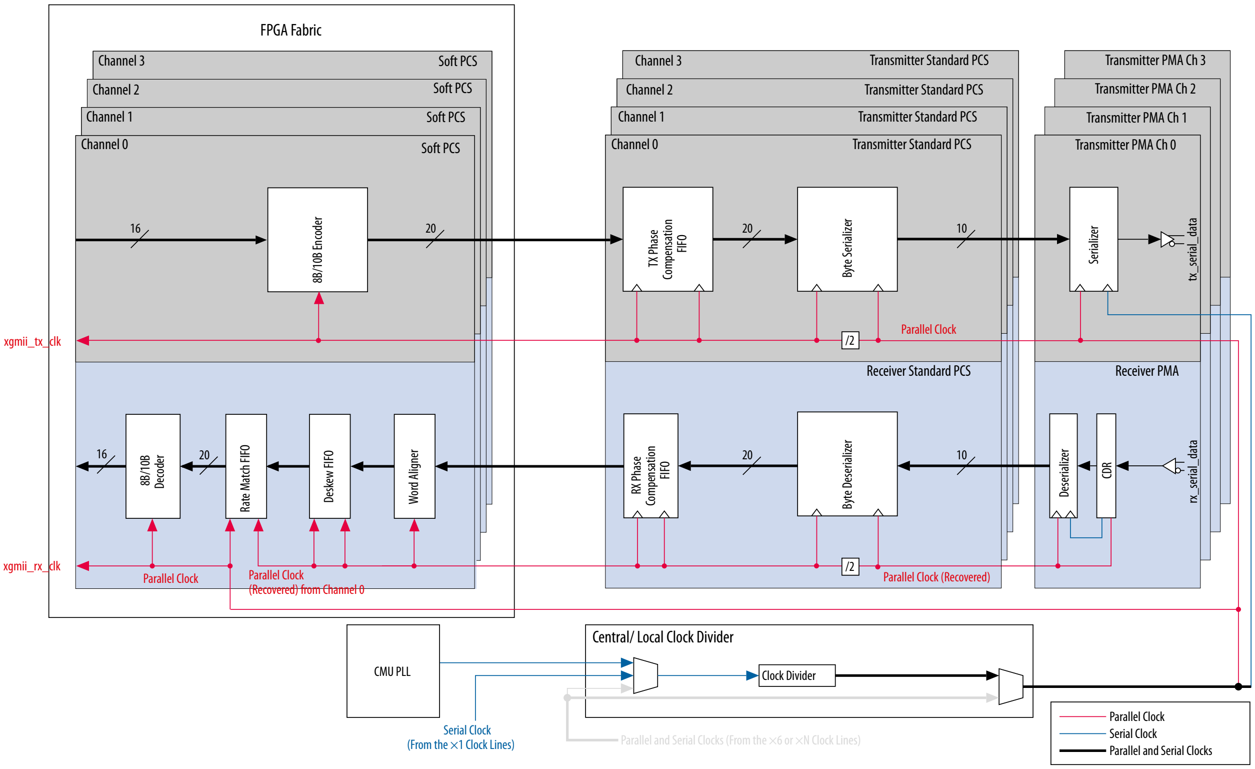


Table 4-6: Input Reference Clock Frequency and Interface Speed Specifications for XAUI Configurations

Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
156.25	16-bit data, 2-bit control	156.25

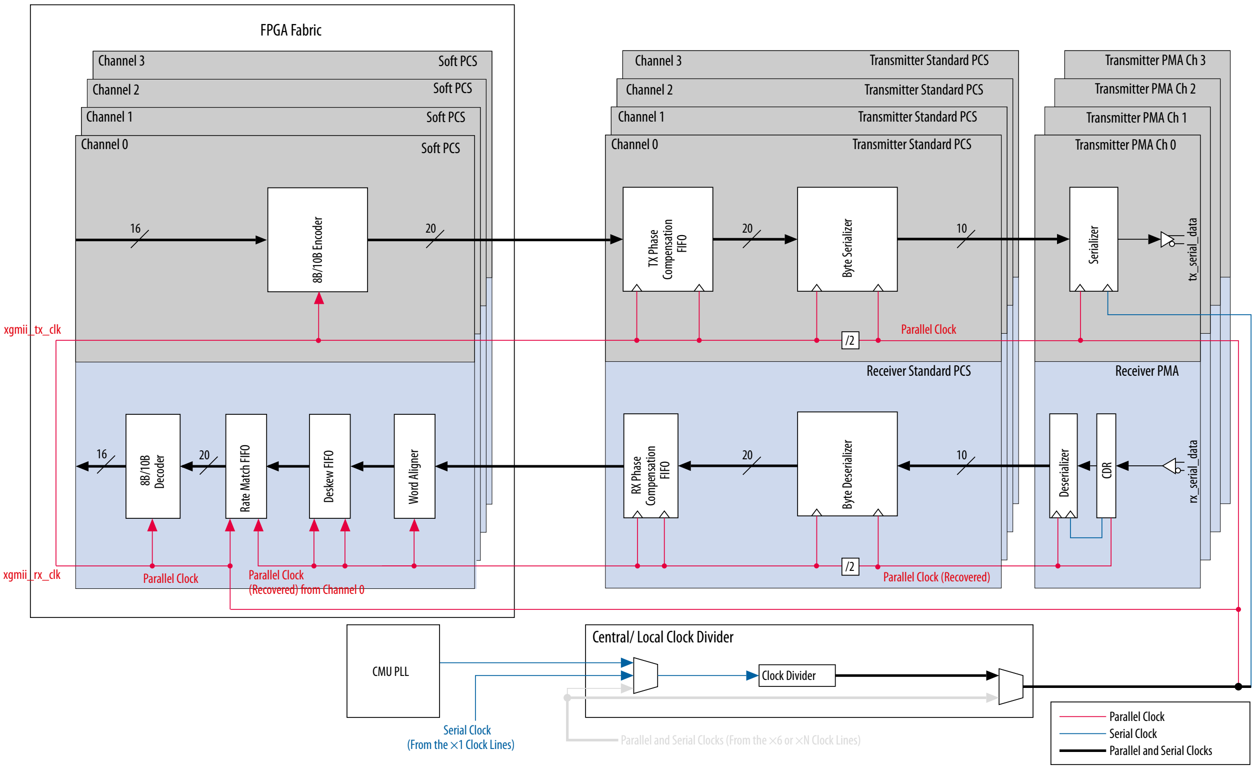
Transceiver Clocking Guidelines for Soft PCS Implementation

In the soft PCS implementation in the XAUI configuration, you must route `xgmii_rx_clk` to `xgmii_tx_clk` as shown in the following figure.

This method uses `xgmii_rx_clk` to compensate for the phase difference on the TX side.

Without this method, the `tx_digitalreset` signal may experience intermittent failure.

Figure 4-26: Transceiver Clocking for XAUI Soft PCS Implementation



Transceiver Channel Placement Guidelines

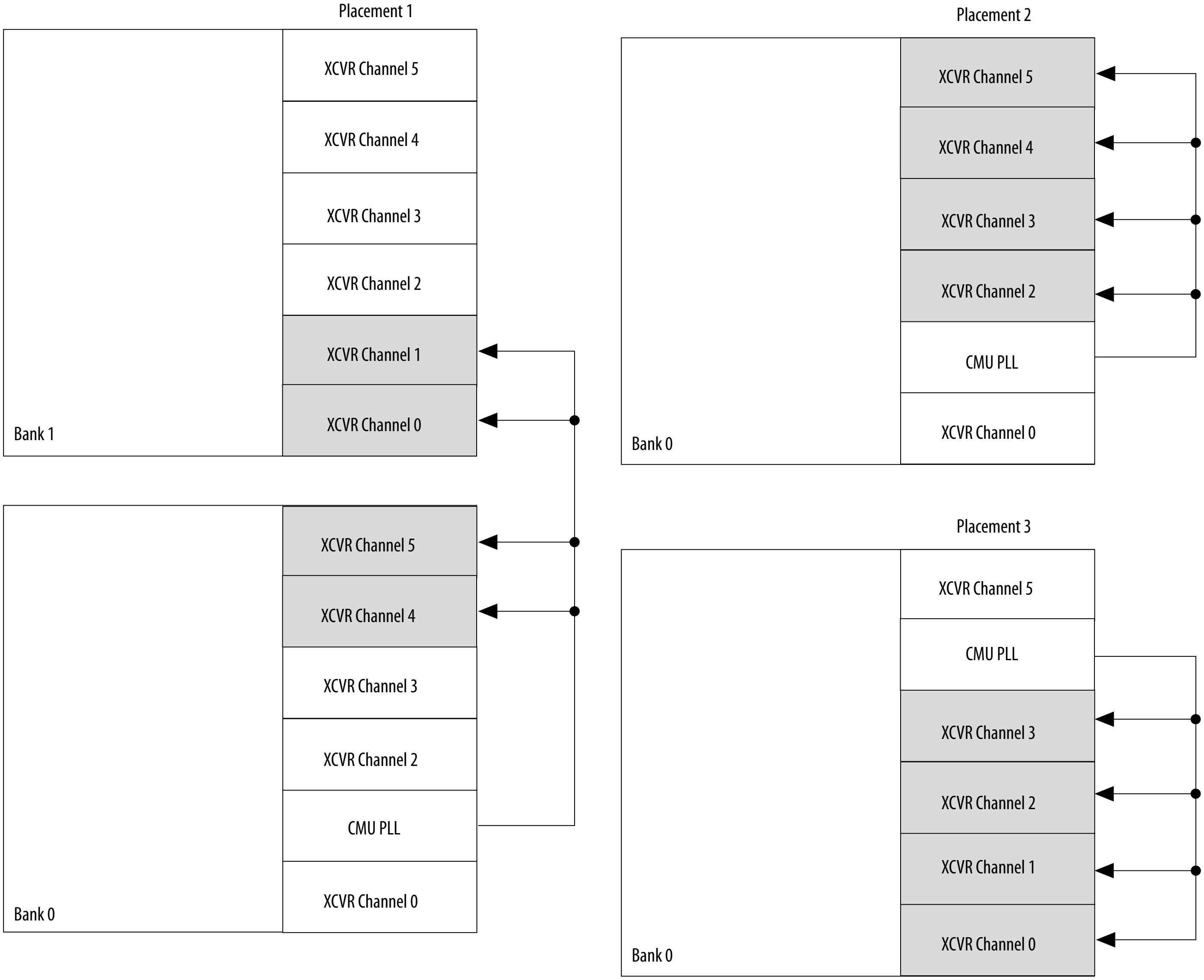
In the soft PCS implementation of the XAUI configuration, you can construct the four XAUI lanes at any channels within the two transceiver banks. However, Altera recommends you place the four channels contiguously to close timing more easily The channels may all be placed in one bank or they may span two banks. The following figure shows several possible channel placements when using the CMU PLL to drive the XAUI link.

The soft PCS implementation of the XAUI configuration has the following channel placement restrictions:

- The channels must be contiguous.
- Ch1 or Ch4 must be selected as logical channel 0.

Figure 4-27: Transceiver Channel Placement Guidelines in a XAUI Configuration

The Quartus II software implements the XAUI PCS in soft logic. Each XAUI link requires a dedicated CMU PLL. A single CMU PLL cannot be shared among different XAUI links.



Related Information

To implement the QSF assignment workaround using the Assignment Editor, refer to the "XAUI PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide.

10GBASE-R

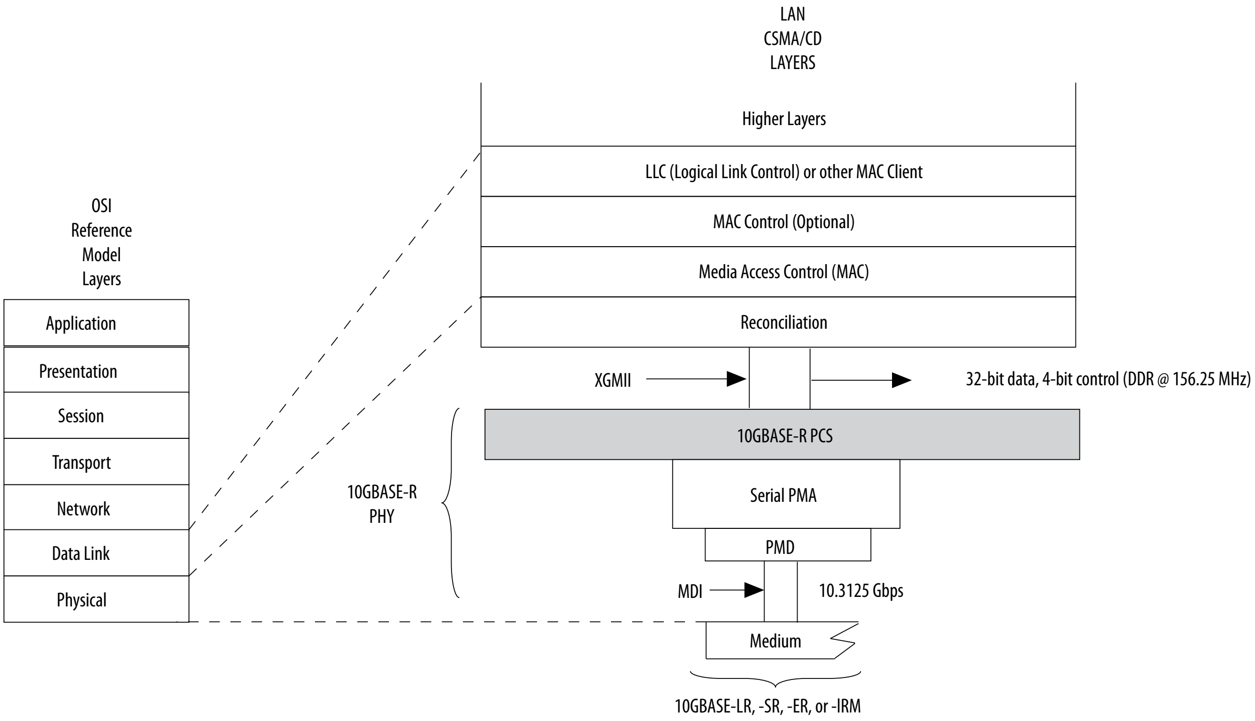
Arria V GT and ST devices support 10GBASE-R in PMA direct mode using soft PCS. 10GBASE-R is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in clause 49 of the IEEE 802.3-2008 specification. The 10GBASE-R PHY uses the XGMII interface to connect to the IEEE802.3

media access control (MAC) and reconciliation sublayer (RS). The IEEE 802.3-2008 specification requires each 10GBASE-R link to support a 10 Gbps data rate at the XGMII interface and a 10.3125 Gbps serial line rate with 64B/66B encoding.

The 10-Gbps transceivers transmitter in Arria V GT and ST devices are compliant with the 10GBASE-KR specification under the following conditions:

- A maximum of three full duplex channels within a bank are used. These three channels do not include a CMU PLL.
- The transmitted signal is 64B/66B encoded.

Figure 4-28: 10GBASE-R PHY Connection to IEEE802.3 MAC and RS



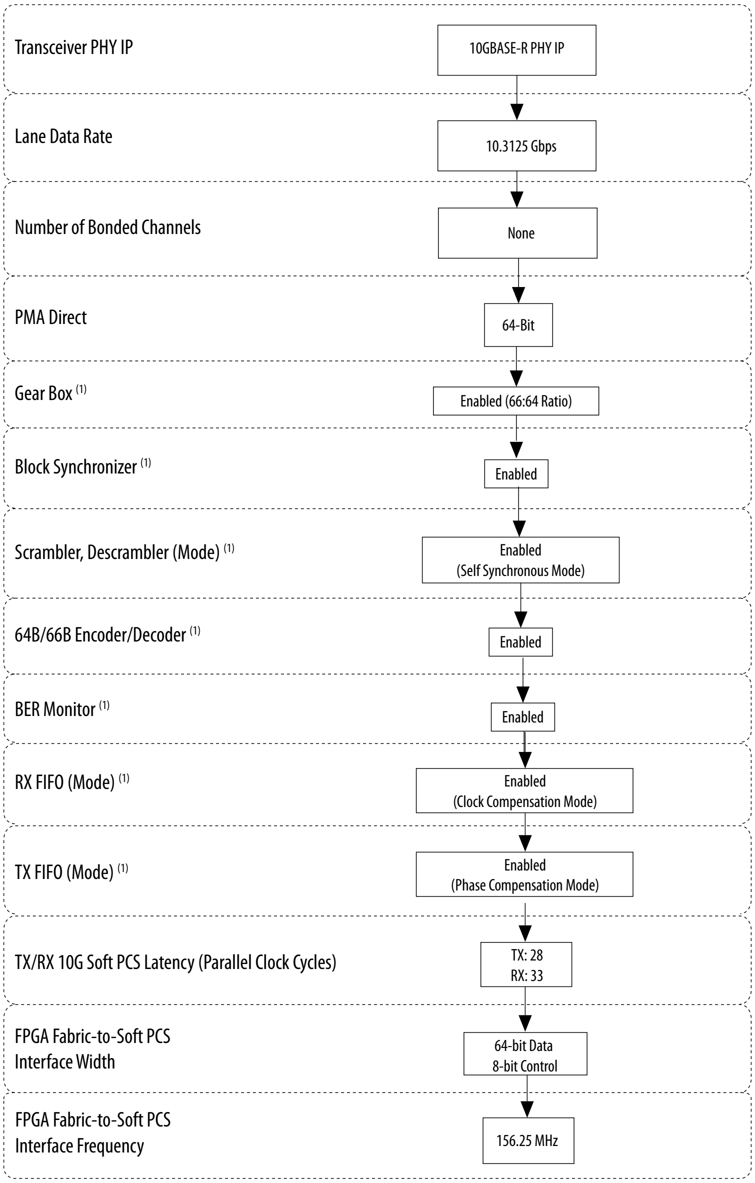
To implement a 10GBASE-R link, instantiate the **10GBASE-R PHY IP** core in the IP Catalog, under **Ethernet** in the Interfaces menu.

Related Information

Refer to the 10GBASE-R PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide.

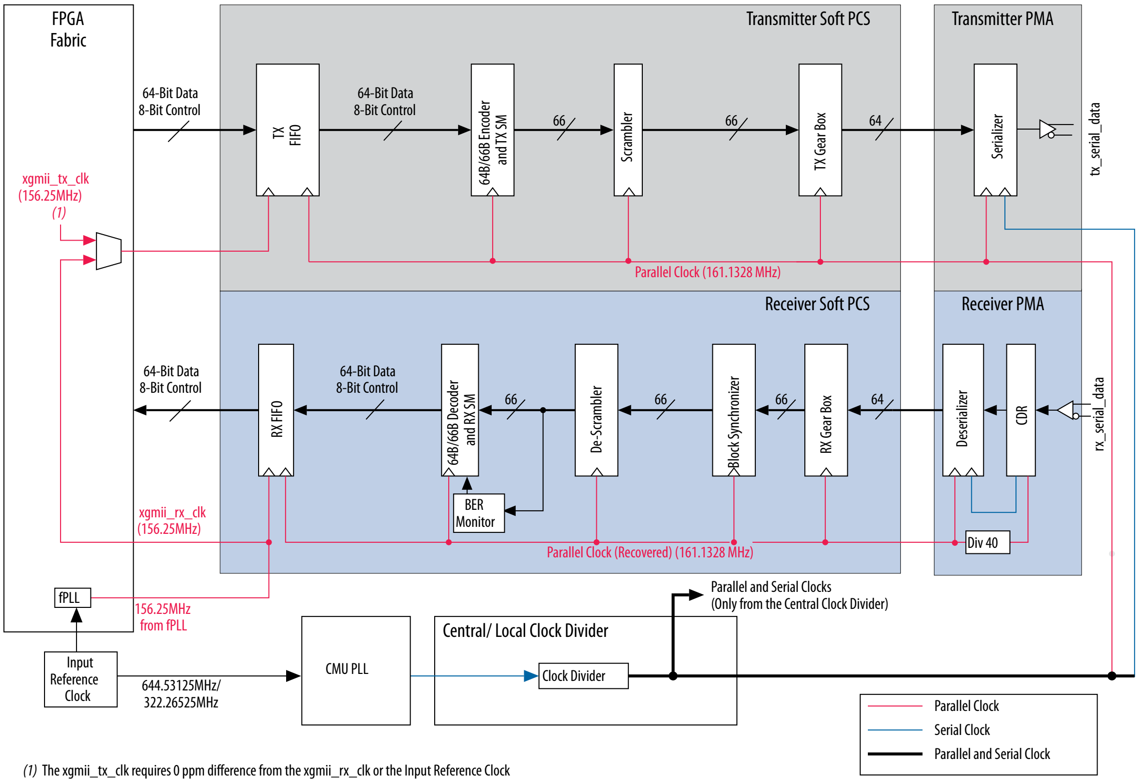
10GBASE-R Transceiver Datapath Configuration

Figure 4-29: 10GBASE-R Datapath Configuration for Arria V GT and ST Devices


Transceiver PHY IP
10GBASE-R PHY IP
Lane Data Rate
10.3125 Gbps
Number of Bonded Channels
None
PMA Direct
64-Bit
Gear Box (1)
Enabled (66:64 Ratio)
Block Synchronizer (1)
Enabled
Scrambler, Descrambler (Mode) (1)
Enabled (Self Synchronous Mode)
64B/66B Encoder/Decoder (1)
Enabled
BER Monitor (1)
Enabled
RX FIFO (Mode) (1)
Enabled (Clock Compensation Mode)
TX FIFO (Mode) (1)
Enabled (Phase Compensation Mode)
TX/RX 10G Soft PCS Latency (Parallel Clock Cycles)
TX: 28
RX: 33
FPGA Fabric-to-Soft PCS Interface Width
64-bit Data
8-bit Control
FPGA Fabric-to-Soft PCS Interface Frequency
156.25 MHz

Note:
1. Implemented in soft logic.

Figure 4-30: Transceiver Channel Datapath for a 10GBASE-R Configuration for Arria V GT and ST Devices



(1) The xgmii_tx_clk requires 0 ppm difference from the xgmii_rx_clk or the Input Reference Clock

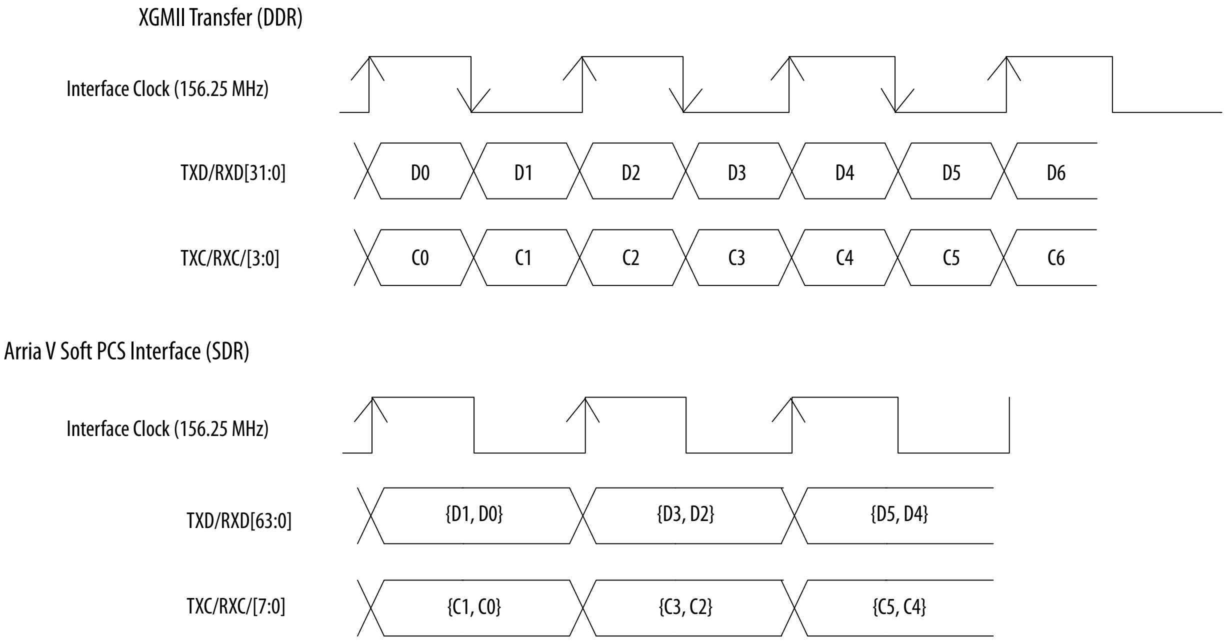
10GBASE-R Supported Features

64-Bit Single Data Rate (SDR) Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the 10GBASE-R soft PCS and the Ethernet MAC/RS. The XGMII interface defines the 32-bit data and 4-bit wide control character clocked between the MAC/RS and the soft PCS at both the positive and negative edge (double data rate – DDR) of the 156.25 MHz interface clock.

Arria V soft PCS does not support the XGMII interface to the MAC/RS as defined in the IEEE 802.3-2008 specification. Instead, they support a 64-bit data and 8-bit control SDR interface between the MAC/RS and the soft PCS.

Figure 4-31: XGMII Interface (DDR) versus Arria V Soft PCS Interface (SDR) for 10GBASE-R



64B/66B Encoding/Decoding

Arria V soft PCS in a 10GBASE-R configuration supports 64B/66B encoding and decoding as specified in Clause 49 of the IEEE802.3-2008 specification. The 64B/66B encoder receives 64-bit data and 8-bit control code from the transmitter FIFO and converts it into 66-bit encoded data. The 66-bit encoded data contains two overhead sync header bits that the receiver soft PCS uses for block synchronization and bit-error rate (BER) monitoring.

The 64B/66B encoding also ensures enough transitions on the serial data stream for the receiver clock data recovery (CDR) to maintain its lock on the incoming data.

Transmitter and Receiver State Machines

Arria V soft PCS in a 10GBASE-R configuration implement the transmitter and receiver state diagrams shown in Figure 49-14 and Figure 49-15 of the IEEE802.3-2008 specification.

Besides encoding the raw data specified in the 10GBASE-R soft PCS, the transmitter state diagram performs functions such as transmitting local faults (LBLOCK_T) under reset, as well as transmitting error codes (EBLOCK_T) when the 10GBASE-R soft PCS rules are violated.

Besides decoding the incoming data specified in the 10GBASE-R soft PCS, the receiver state diagram performs functions such as sending local faults (LBLOCK_R) to the MAC/RS under reset and substituting error codes (EBLOCK_R) when the 10GBASE-R soft PCS rules are violated.

Block Synchronizer

The block synchronizer in the receiver soft PCS determines when the receiver has obtained lock to the received data stream. It implements the lock state diagram shown in Figure 49-12 of the IEEE 802.3-2008 specification.

The block synchronizer provides a status signal to indicate whether it has achieved block synchronization or not.

Self-Synchronous Scrambling/Descrambling

The scrambler/descrambler blocks in the transmitter/receiver soft PCS implements the self-synchronizing scrambler/descrambler polynomial 1 + x39 + x58, as described in clause 49 of the IEEE 802.3-2008 specification. The scrambler/descrambler blocks are self-synchronizing and do not require an initialization seed. Barring the two sync header bits in each 66-bit data block, the entire payload is scrambled or descrambled.

BER Monitor

The BER monitor block in the receiver soft PCS implements the BER monitor state diagram shown in Figure 49-13 of the IEEE 802.3-2008 specification. The BER monitor provides a status signal to the MAC whenever the link BER threshold is violated.

The 10GBASE-R PHY IP core provides a status flag to indicate a high BER whenever 16 synchronization header errors are received within a 125 µs window.

Clock Compensation

The receiver FIFO in the receiver soft PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. The receiver FIFO does so by inserting Idles (/I/) and deleting Idles (/I/) or Ordered Sets (/O/), depending on the ppm difference.

Idle Insertion -- The receiver FIFO inserts eight /I/ codes following an /I/ or /O/ to compensate for clock rate disparity.

Idle (/I/) or Sequence Ordered Set (/O/) Deletion -- The receiver FIFO deletes either four /I/ codes or ordered sets (/O/) to compensate for the clock rate disparity. The receiver FIFO implements the following IEEE802.3-2008 deletion rules:

- Deletes the lower four /I/ codes of the current word when the upper four bytes of the current word do not contain a Terminate /T/ control character.
- Deletes the upper four /I/ codes of the current word when the previous word's lower four bytes do not contain a Terminate /T/ control character.
- Deletes one /O/ ordered set only when the receiver FIFO receives two consecutive /O/ ordered sets.

Related Information

Refer to the 10GBASE-R PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide

10GBASE-R Transceiver Clocking

The CMU PLL can be used as a TX PLL for Arria V GT and ST devices. Arria V GZ devices can use either the CMU PLL or the ATX PLL as a TX PLL.

Table 4-7: Input Reference Clock Frequency and Interface Speed Specifications for 10GBASE-R Configurations

Input Reference Clock Frequency (MHz)	FPGA Fabric-Soft PCS Interface Width	FPGA Fabric-Soft PCS Interface Frequency (MHz)
644.53125, 322.265625	64-bit data, 8-bit control	156.25

Serial Digital Interface

The Society of Motion Picture and Television Engineers (SMPTE) defines various Serial Digital Interface (SDI) standards for transmission of uncompressed video.

The following SMPTE standards are popular in video broadcasting applications:

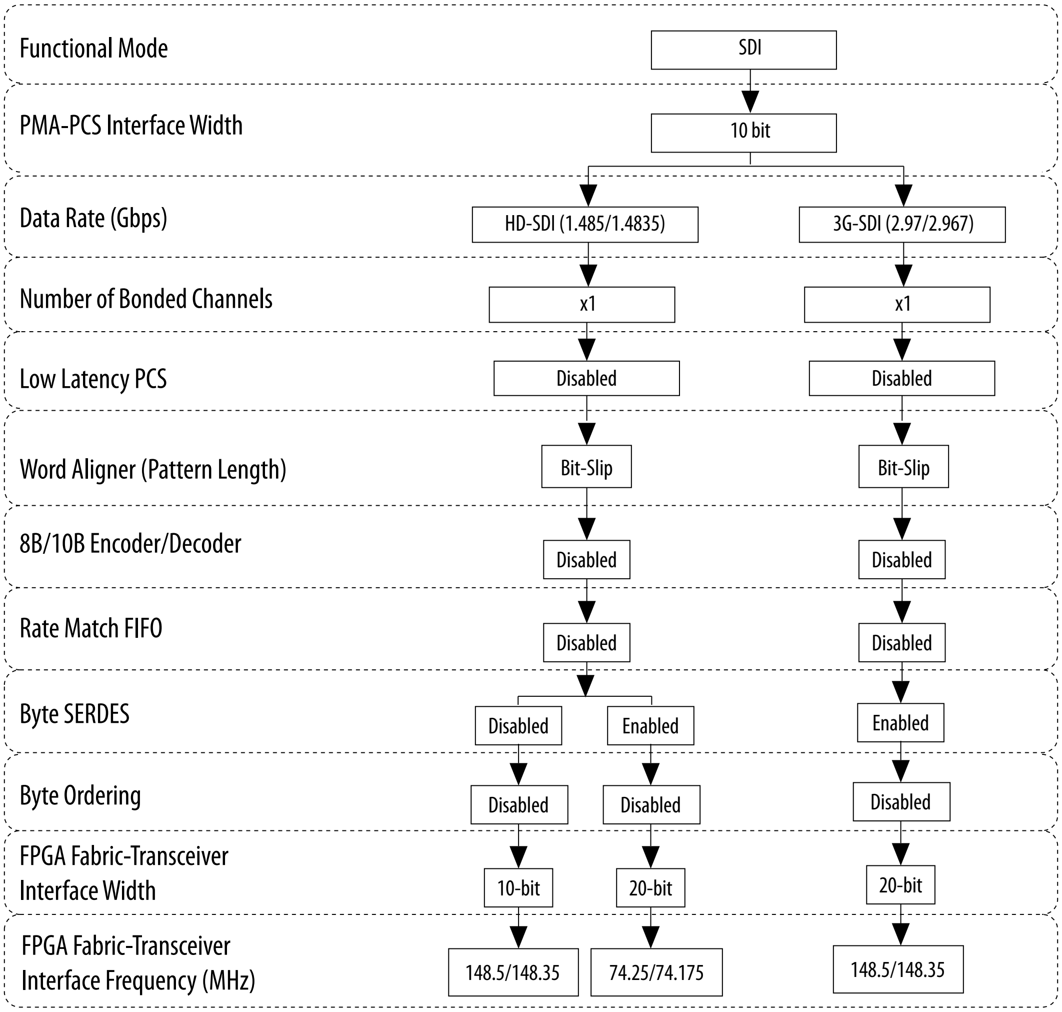
- SMPTE 259M standard - more popularly known as the standard-definition (SD) SDI; defined to carry video data at 270 Mbps
- SMPTE 292M standard - more popularly known as the high-definition (HD) SDI; defined to carry video data at either 1485 Mbps or 1483.5 Mbps
- SMPTE 424M standard - more popularly known as the third-generation (3G) SDI; defined to carry video data at either 2970 Mbps or 2967 Mbps

Configurations Supported in SDI Mode

Table 4-8: Configurations Supported in SDI Mode

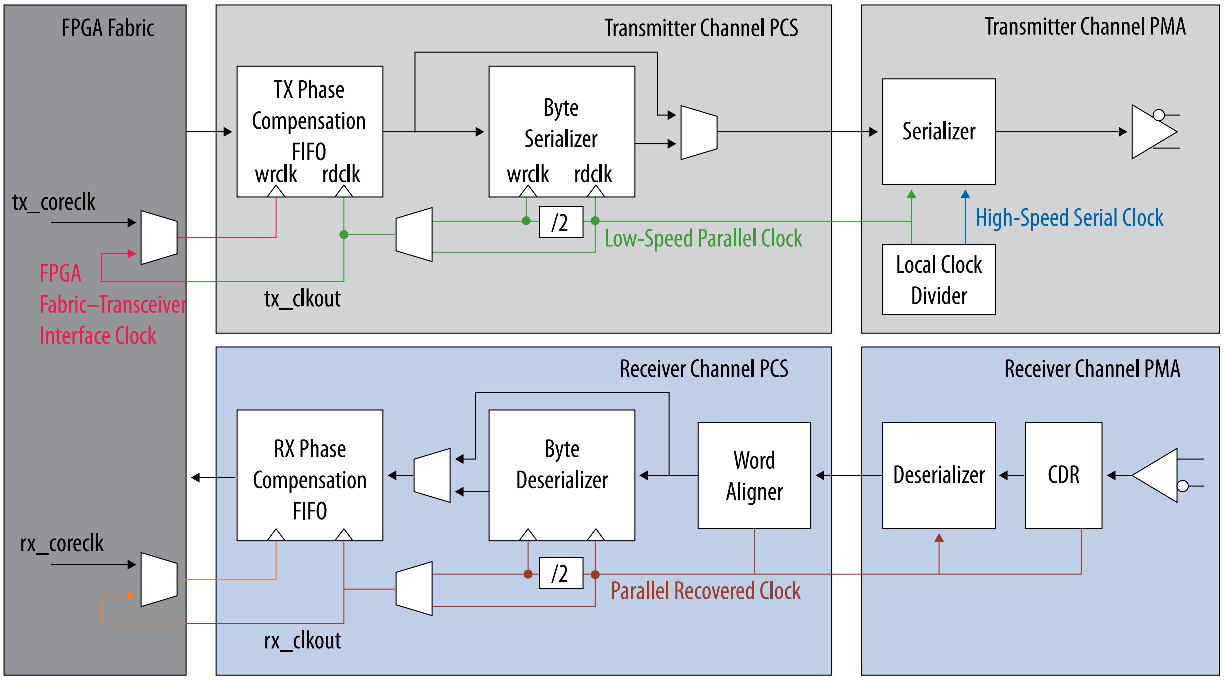
Configuration	Data Rate (Mbps)	REFCLK Frequencies (MHz)	FPGA Fabric-Transceiver Interface Width
HD	1,485	74.25, 148.5	10 bit and 20 bit
	1,483.5	74.175, 148.35	10 bit and 20 bit
3G	2,970	148.5, 297	Only 20-bit interfaces allowed in 3G
	2,967	148.35, 296.7	Only 20-bit interfaces allowed in 3G

Figure 4-32: SDI Mode



Serial Digital Interface Transceiver Datapath

Figure 4-33: SDI Mode Transceiver Datapath



Transmitter Datapath

The transmitter datapath in the HD-SDI configuration with a 10-bit wide FPGA fabric-transceiver interface consists of the transmitter phase compensation FIFO and the 10:1 serializer. In HD-SDI and 3G-SDI configurations with 20-bit wide FPGA fabric-transceiver interface, the transmitter datapath also includes the byte serializer.

Note: In SDI mode, the transmitter is purely a parallel-to-serial converter. You must implement the SDI transmitter functions, such as the scrambling and cyclic redundancy check (CRC) code generation, in the FPGA logic array.

Receiver Datapath

In the 10-bit channel width SDI configuration, the receiver datapath consists of the clock recovery unit (CRU), 1:10 deserializer, word aligner in bit-slip mode, and receiver phase compensation FIFO. In the 20-bit channel width SDI configuration, the receiver datapath also includes the byte deserializer.

Note: You must implement the SDI receiver functions, such as descrambling, framing, and CRC checker, in the FPGA logic array.

Receiver Word Alignment and Framing

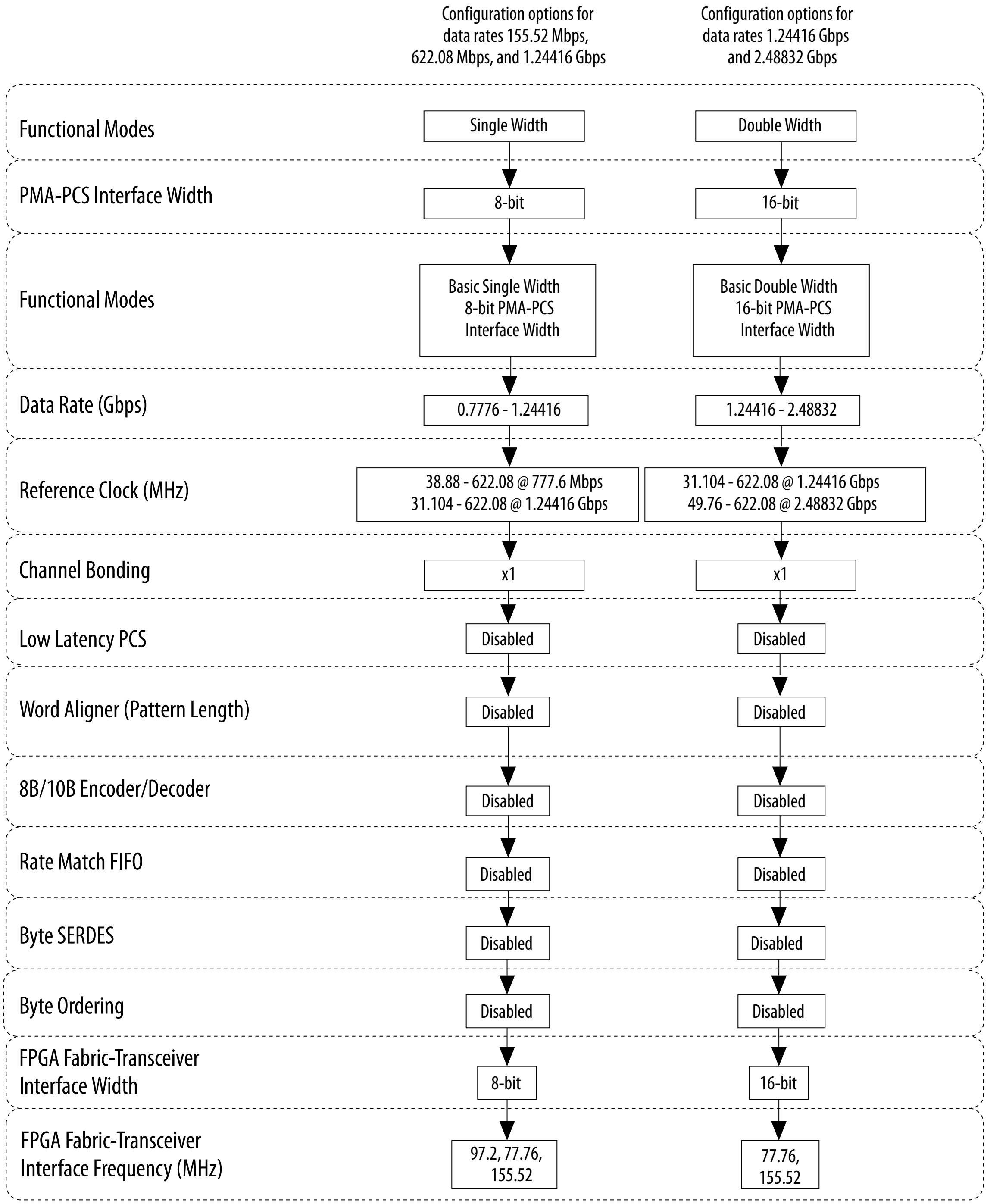
In SDI systems, the word aligner in the receiver datapath is not useful because the word alignment and framing happen after descrambling. Altera recommends that you drive the `rx_bitslip` signal of the PHY IP core low to avoid having the word aligner insert bits in the received data stream.

Gigabit-Capable Passive Optical Network (GPON)

The GPON protocol network provides optical fiber cabling and signals to the home and office using a point-to-multipoint scheme. GPON data rates of 155.52 Mbps, 622.08 Mbps, 1.24416 Gbps, and 2.48832 Gbps, with a reference clock of 155.52 MHz are supported. The minimum supported data rate is 600 Mbps, so a 5x oversampling factor is used for the GPON data rate of 155.52 Mbps, resulting in a data rate of 777.6 Mbps.

Note: You must build the oversampling at the PLD.

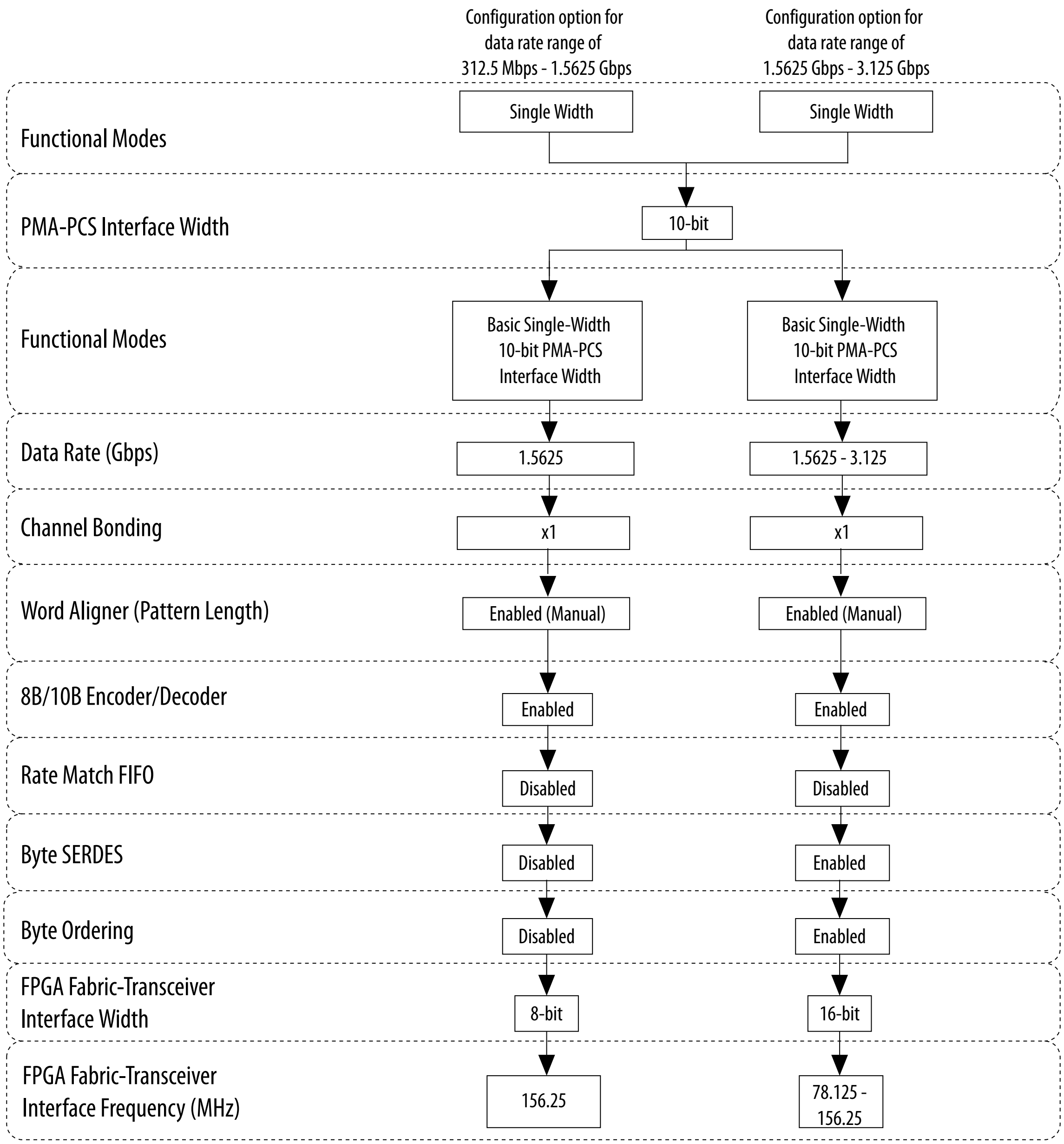
Figure 4-34: Configurations for the GPON Protocol


Configuration options for data rates 155.52 Mbps, 622.08 Mbps, and 1.24416 Gbps
Configuration options for data rates 1.24416 Gbps and 2.48832 Gbps
Functional Modes
Single Width
Double Width
PMA-PCS Interface Width
8-bit
16-bit
Functional Modes
Basic Single Width 8-bit PMA-PCS Interface Width
Basic Double Width 16-bit PMA-PCS Interface Width
Data Rate (Gbps)
0.7776 - 1.24416
1.24416 - 2.48832
Reference Clock (MHz)
38.88 - 622.08 @ 777.6 Mbps
31.104 - 622.08 @ 1.24416 Gbps
31.104 - 622.08 @ 1.24416 Gbps
49.76 - 622.08 @ 2.48832 Gbps
Channel Bonding
x1
x1
Low Latency PCS
Disabled
Disabled
Word Aligner (Pattern Length)
Disabled
Disabled
8B/10B Encoder/Decoder
Disabled
Disabled
Rate Match FIFO
Disabled
Disabled
Byte SERDES
Disabled
Disabled
Byte Ordering
Disabled
Disabled
FPGA Fabric-Transceiver Interface Width
8-bit
16-bit
FPGA Fabric-Transceiver Interface Frequency (MHz)
97.2, 77.76, 155.52
77.76, 155.52

Serial Data Converter (SDC) JESD204

The SDC (JESD204) protocol conforms to JESD204, a JEDEC standard that enables a high-speed serial connection between analog-to-digital converters and logic devices using only a two-wire high-speed serial interface. SDC (JESD204) data rate ranges of 312.5 Mbps to 3.125 Gbps are supported. The minimum supported data rate is 611 Mbps, so a 5x oversampling factor is used for the SDC (JESD204) data rate of 312.5 Mbps, resulting in a data rate of 1.5625 Gbps.

Figure 4-35: Configurations for the SDC (JESD204) Protocol


Configuration option for data rate range of 312.5 Mbps - 1.5625 Gbps
Configuration option for data rate range of 1.5625 Gbps - 3.125 Gbps
Functional Modes
Single Width
Single Width
PMA-PCS Interface Width
10-bit
Functional Modes
Basic Single-Width 10-bit PMA-PCS Interface Width
Basic Single-Width 10-bit PMA-PCS Interface Width
Data Rate (Gbps)
1.5625
1.5625 - 3.125
Channel Bonding
x1
x1
Word Aligner (Pattern Length)
Enabled (Manual)
Enabled (Manual)
8B/10B Encoder/Decoder
Enabled
Enabled
Rate Match FIFO
Disabled
Disabled
Byte SERDES
Disabled
Enabled
Byte Ordering
Disabled
Enabled
FPGA Fabric-Transceiver Interface Width
8-bit
16-bit
FPGA Fabric-Transceiver Interface Frequency (MHz)
156.25
78.125 - 156.25

SATA and SAS Protocols

Serial ATA (SATA) and Serial Attached SCSI (SAS) are data storage protocol standards that have the primary function of transferring data (directly or otherwise) between the host system and mass storage devices, such as hard disk drives, optical drives, and solid-state disks.

These serial storage protocols offer several advantages over older parallel storage protocol (ATA and SCSI) interfaces:

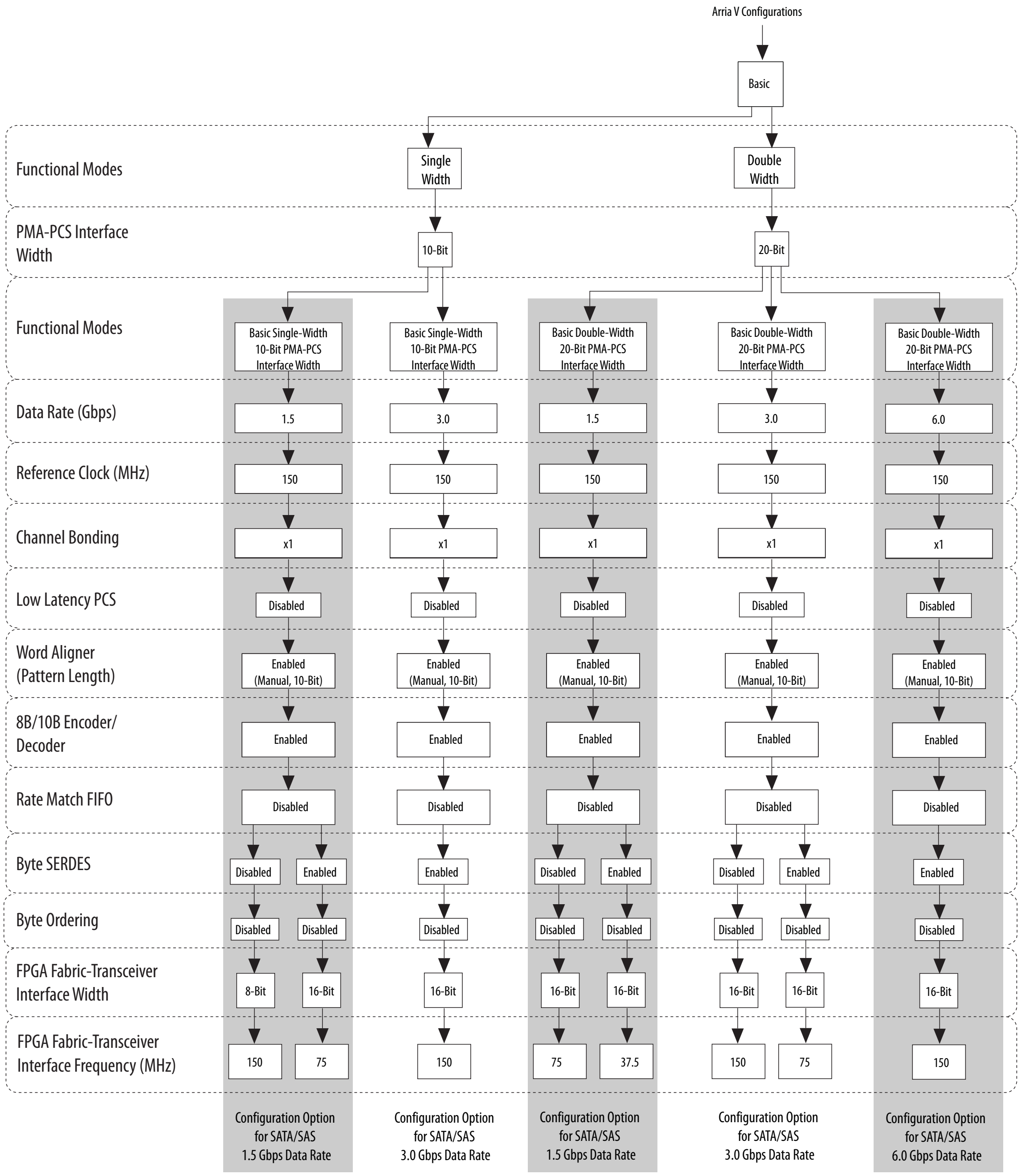
- Faster data transfer
- Hot swapping (when supported by the operating system)
- Thinner cables for more efficient air cooling
- Increased operation reliability

Table 4-9: Serial Data Rates for SATA and SAS Protocols

Protocol	SATA (Gbps)	SAS (Gbps)
Gen1	1.5	3.0
Gen2	3.0	6.0
Gen3	6.0	-

Figure 4-36: Configurations for the SATA and SAS Protocols

If you are configuring the SATA channel to support up to Gen3, set the base data rate to 6 Gbps and use the TX local clock divider to divide down to the Gen2 and Gen1 data rate.

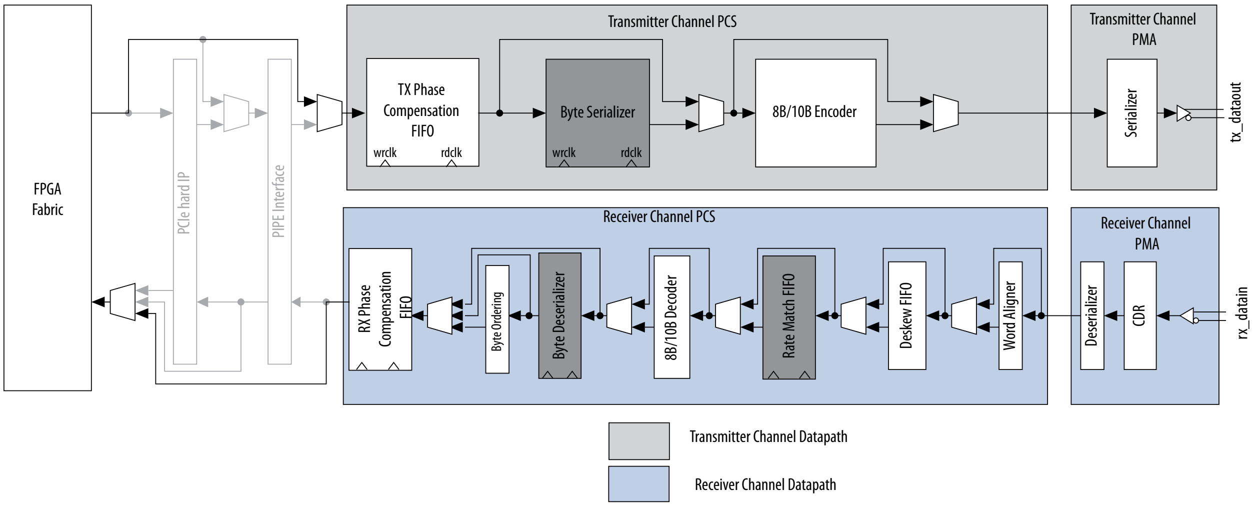


Deterministic Latency Protocols—CPRI and OBSAI

A deterministic latency option is available for use in high-speed serial interfaces such as the Common Public Radio Interface (CPRI) and OBSAI Reference Point 3 (OBSAI RP3). Both CPRI and OBSAI RP3 protocols place stringent requirements on the amount of latency variation that is permissible through a link that implements these protocols.

Arria V GT devices also support 9.8304 Gbps CPRI with PMA direct configuration; PCS is implemented in soft logic.

Figure 4-37: Transceiver Datapath in Deterministic Latency Mode



Related Information

Implementing 9.8G CPRI in Arria V GT and ST FPGAs
Describes a soft PCS implementation for 9.8G CPRI.

Latency Uncertainty Removal with the Phase Compensation FIFO in Register Mode

To remove the latency uncertainty through the receiver's phase compensation FIFO, the receiver and transmitter phase compensation FIFOs are always set to register mode. In register mode, the phase compensation FIFO acts as a register and thereby removes the uncertainty in latency. The latency through the transmitter and receiver phase compensation FIFO in register mode is one clock cycle.

The following options are available:

- Single-width mode with 8-bit channel width and 8B/10B encoder enabled or 10-bit channel width with 8B/10B disabled
- Double-width mode with 16-bit channel width and 8B/10B encoder enabled or 20-bit channel width with 8B/10B disabled

Channel PLL Feedback for Deterministic Relationship

To implement the deterministic latency functional mode, the phase relationship between the low-speed parallel clock and channel PLL input reference clock must be deterministic. A feedback path is enabled to ensure a deterministic relationship between the low-speed parallel clock and channel PLL input reference clock.

To achieve deterministic latency through the transceiver, the reference clock to the channel PLL must be the same as the low-speed parallel clock. For example, if you need to implement a data rate of 1.2288 Gbps for the CPRI protocol, which places stringent requirements on the amount of latency variation, you must choose a reference clock of 122.88 MHz to allow the usage of a feedback path from the channel PLL. This feedback path reduces the variations in latency.

When you select this option, provide an input reference clock to the channel PLL that has the same frequency as the low-speed parallel clock.

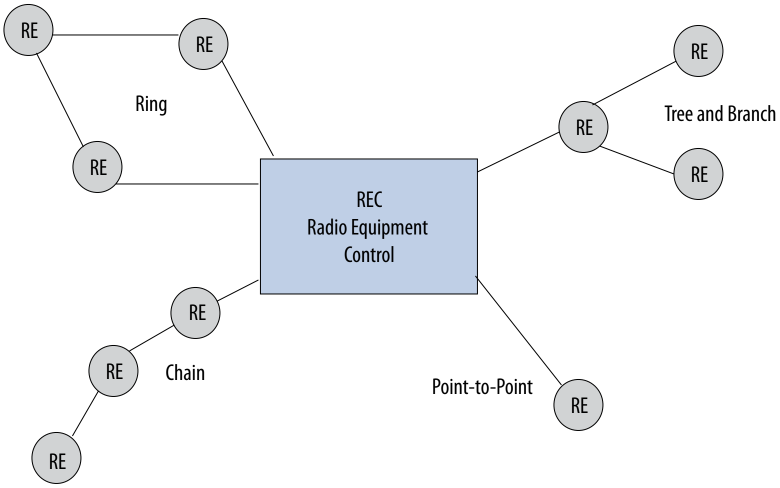
CPRI and OBSAI

Use the deterministic latency functional mode to implement protocols such as CPRI and OBSAI.

The CPRI interface defines a digital point-to-point interface between the Radio Equipment Control (REC) and the Radio Equipment (RE), allowing flexibility in either co-locating the REC and the RE, or a remote location of the RE.

Figure 4-38: CPRI Topologies

In most cases, CPRI links are between REC and RE modules or between two RE modules in a chain configuration.



If the destination for the high-speed serial data that leaves the REC is the first RE, it is a single-hop connection. If the serial data from the REC must traverse through multiple REs before reaching the destination RE, it is a multi-hop connection.

Remotely locating the RF transceiver from the main base station introduces a complexity with overall system delay. The CPRI specification requires that the accuracy of measurement of roundtrip delay on single-hop and multi-hop connections be within ±16.276 ns to properly estimate the cable delay.

For a single-hop system, this allows a variation in roundtrip delay of up to ±16.276 ns. However, for multi-hop systems, the allowed delay variation is divided among the number of hops in the connection—typically, equal to ±16.276 ns/(the number of hops) but not always equally divided among the hops.

Deterministic latency on a CPRI link also enables highly accurate triangulation of the location of the caller.

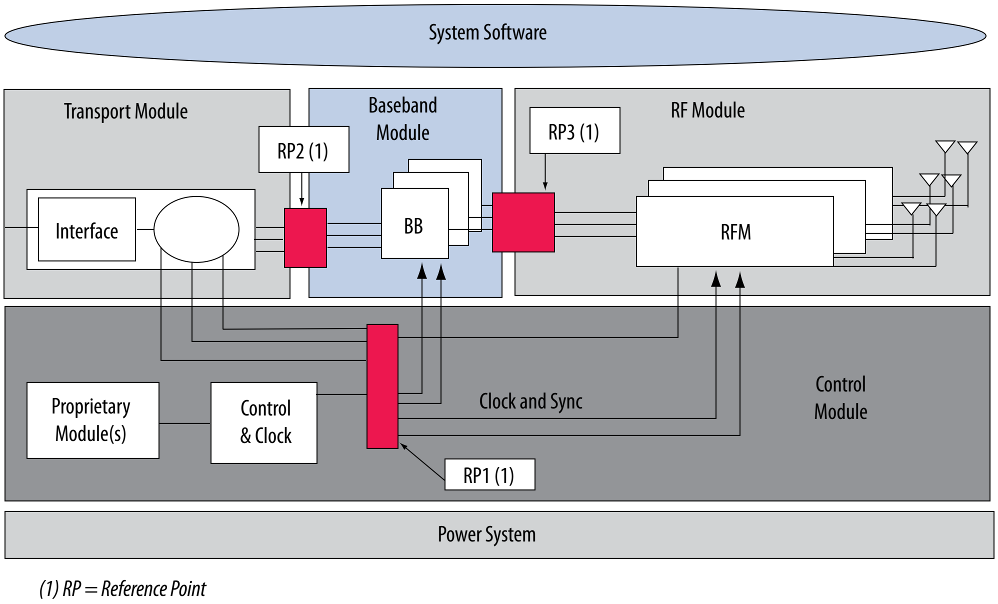
OBSAI was established by several OEMs to develop a set of specifications that can be used for configuring and connecting common modules into base transceiver stations (BTS).

The BTS has four main modules:

- Radio frequency (RF)
- Baseband
- Control
- Transport

In a typical BTS, the radio frequency module (RFM) receives signals using portable devices and converts the signals to digital data. The baseband module processes the encoded signal and brings it back to the baseband before transmitting it to the terrestrial network using the transport module. A control module maintains the coordination between these three functions.

Figure 4-39: Example of the OBSAI BTS Architecture



Using the deterministic latency option, you can implement the CPRI data rates in the following modes:

- Single-width mode—with 8/10-bit channel width
- Double-width mode—with 16/20-bit channel width

Table 4-10: Sample Channel Width Options for Supported Serial Data Rates

Serial Data Rate (Mbps)	Channel Width (FPGA-PCS Fabric)			
	Single-Width		Double-Width	
	8-Bit	16-Bit	16-Bit	32-Bit
614.4	Yes	Yes	No	No
1228.8	Yes	Yes	Yes	Yes
2457.6	No	Yes	Yes	Yes
3072	No	Yes	Yes	Yes
4915.2	No	No	No	Yes
6144	No	No	No	Yes

Serial Data Rate (Mbps)	Channel Width (FPGA-PCS Fabric)			
	Single-Width		Double-Width	
	8-Bit	16-Bit	16-Bit	32-Bit
9830.4 [36]	N/A	N/A	N/A	N/A

Related Information

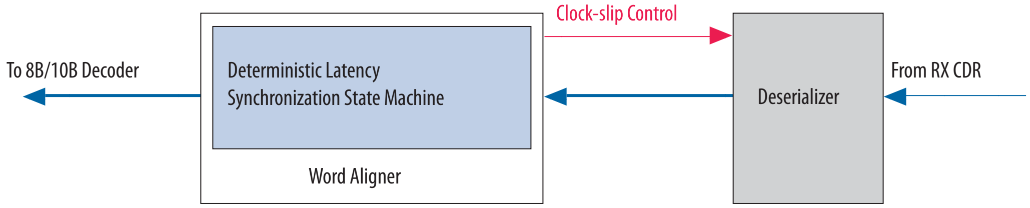
Transceiver Architecture in Arria V Devices

CPRI Enhancements

The deterministic latency state machine in the word aligner reduces the known delay variation from the word alignment process and automatically synchronizes and aligns the word boundary by slipping a clock cycle in the deserializer. Incoming data to the word aligner is aligned to the boundary of the word alignment pattern (K28.5). User logic is not required to manipulate the TX bit slipper for constant round-trip delay. In manual mode, the TX bit slipper is able to compensate one unit interval (UI).

The word alignment pattern (K28.5) position varies in byte deserialized data. Delay variation is up to ½ parallel clock cycle. You must add in extra user logic to manually check the K28.5 position in byte deserialized data for the actual latency.

Figure 4-40: Deterministic Latency State Machine in the Word Aligner



[36] CPRI 9830.4 Mbps uses PMA direct mode with 80-bits PMA-PLD data width and is available only in 10-Gbps channels. For transmit jitter compliance, refer to the Transceiver Architecture in Arria V Devices chapter for maximum channel conditions.
[37] Backward compatible with CPRI designs in Arria II devices.
[38] Enhanced deterministic latency feature in Arria V devices.

Table 4-11: Methods to Achieve Deterministic Latency Mode in Arria V Devices

Existing Feature [37]		Enhanced Feature [38]	
Description	Requirement	Description	Requirement
Manual alignment with bit position indicator provides deterministic latency. Delay variation up to 1 parallel clock cycle	Extra user logic to manipulate the TX bit slipper with a bit position indicator from the word aligner for constant total round-trip delay	Deterministic latency state machine alignment reduces the known delay variation in word alignment operation	None

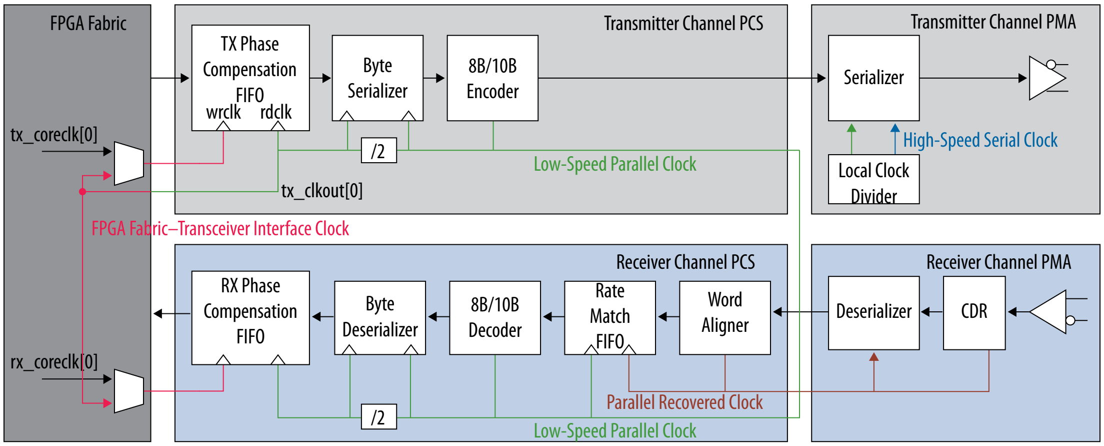
Related Information

Refer to the "Deterministic Latency PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide

Serial RapidIO

The RapidIO Trade Association defines a high-performance, packet-switched interconnect standard to pass data and control information between microprocessors, digital signal, communications, and network processors, system memories, and peripheral devices.

Figure 4-41: Transceiver Datapath in Serial RapidIO (SRIO) Mode



Arria V transceivers support SRIO physical layer specifications, versions 1.3 and 2.1, from 1.25 Gbps to 6.25 Gbps. The transceivers are compliant with x4 channel bonding, deskew state machine, and rate match FIFO.

Synchronization State Machine

The word aligner has a synchronization state machine that handles the receiver lane synchronization.

The synchronization state machine indicates synchronization when the receiver receives 127 K28.5 (10'b0101111100 or 10'b1010000011) synchronization code groups without receiving an intermediate invalid code group. After synchronization, the state machine indicates loss of synchronization when it detects three invalid code groups separated by less than 255 valid code groups, or when it is reset.

The `rx_syncstatus` port of each channel indicates the receiver synchronization:

- High—the lane is synchronized
- Low—the lane has fallen out of synchronization

Table 4-12: Synchronization State Machine Parameters in Serial RapidIO Mode

Parameters	Number
Number of valid K28.5 code groups received to achieve synchronization	127
Number of errors received to lose synchronization	3
Number of continuous good code groups received to reduce the error count by one	255

Rate Match FIFO

In SRIO mode, the rate match FIFO is capable of compensating for up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock.

The rate match FIFO operation begins after the word aligner synchronization status, `rx_syncstatus`, goes high. When the rate matcher receives either of the two 10-bit control patterns followed by the respective 10-bit skip pattern, it inserts or deletes the 10-bit skip pattern as necessary to avoid the rate match FIFO from overflowing or under-running.

In SRIO mode, the rate match FIFO can delete or insert a maximum of one skip pattern from a cluster.

Related Information

Refer to "Chapter 4: PCS and PMA Layers" of Part 6: LP-Serial Physical Layer Specification in the RapidIO Interconnect Specification

Document Revision History

Table 4-13: Document Revision History

Date	Version	Changes
January 2016	2016.01.28	• Removed x8 from list of PCIe Gen2 supported devices in table "Transceiver PCS Features for Arria V Devices". • Removed x8 in "Number of Bonded Channels" row from Gen2 column in figure "PCIe Gen1 and Gen2 PIPE Datapath Configuration".

Date	Version	Changes
March 2015	2015.03.17	• Removed fPLL information from the "Transceiver Clocking and Channel Placement Guidelines in XAUI Configuration" section. • Moved the Word Aligner block to the Receiver Standard PCS section in the "Transceiver Channel Datapath for XAUI Configuration" figure.
September 2014	2014.09.30	• Removed the "Transceivers in a PCIe Hard IP Configuration" figure and replaced it with the "PCIe Gen1 and Gen2 PIPE Datapath Configuration" and "PCIe Gen1 and Gen2 Hard IP and PHY IP Core for PCI Express Datapath Configuration" figures. • Added specific channel placement guidelines in the "PCIe Supported Configurations and Placement Guidelines" section. • Added channel placement guidelines in the XAUI "Transceiver Channel Placement Guidelines" section. • Added another example to the "Transceiver Channel Placement Guidelines in a XAUI Configuration" figure. • Removed second note from the "Transceiver Clocking for XAUI Soft PCS Implementation" figure.
March 2014	2014.03.07	• Updated the "Gigabit Ethernet" section. • Updated the "Rate Match FIFO" section in the "Gigabit Ethernet Transceiver Datapath" section. • Updated the XAUI "Transceiver Channel Placement Guidelines" section. • Added link to external reference in the "Deterministic Latency Protocols—CPRI and OBSAI" section.
May 2013	2013.05.06	• Added link to the known document issues in the Knowledge Base. • Added x2 information to the "PIPE Transceiver Channel Placement Guidelines" section. • Removed the "Receiver Electrical Idle Inference" section. • Updated the figures in the "PCIe Supported Configurations and Placement Guidelines" section. • Added the "Transceiver Clocking Guidelines for Soft PCS Implementation" section.

Date	Version	Changes
March 2013	2013.03.15	• Removed references to x2 channel configuration. • Changed references to the PCIe Specification to version 2.1. • Updated Table 4-1. • Updated Figure 4 -27. • Updated the "XAUI" section. • Updated the "XAUI Supported Features" section. • Updated the "Transceiver Clocking and Channel Placement Guidelines in XAUI Configuration" section. • Updated the "10GBASE-R" section. • Updated Figure 4-30. • Updated Figure 4-31. • Updated Figure 4-32. • Updated the "10GBASE-R Supported Features" section. • Updated the "10GBASE-R Transceiver Clocking" section.
November 2012	2012.11.19	• Reorganized content and updated template. • Added the "XAUI" section. • Added the "PCI Express" section.
June 2012	1.2	• Updated for the Quartus II software version 12.0. • Added the “Serial Digital Interface” section. • Added the “Gigabit-Capable Passive Optical Network (GPON)” section. • Added the “Serial Data Converter (SDC) JESD204” section. • Added the “SATA and SAS Protocols” section. • Updated Figure 4–2 and Figure 4–18. • Added Figure 4–19. • Updated Table 4–1, Table 4–8, and Table 4–9. • Updated the “CPRI Enhancements in Arria V Devices” section. • Added the “Serial RapidIO” section.
November 2011	1.1	Updated for the Quartus II software version 11.1.
August 2011	1.0	Initial release.

Transceiver Custom Configurations in Arria V Devices

5

2014.09.30

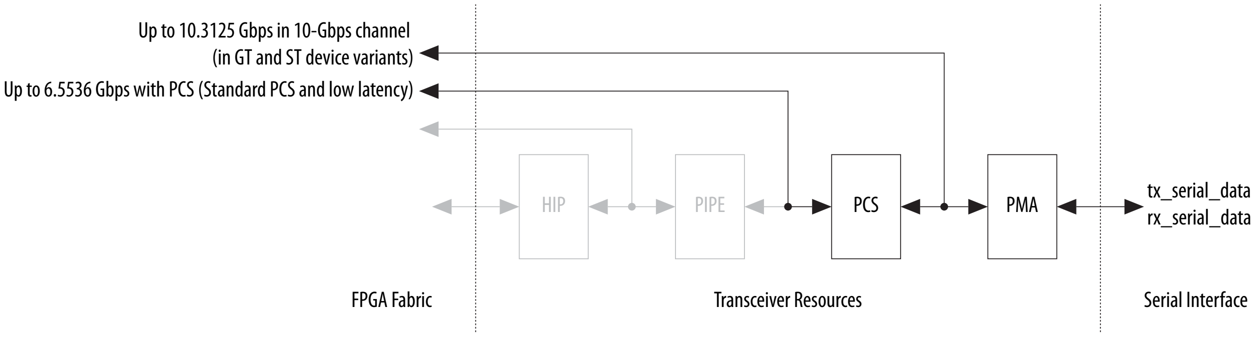
AV53005 Subscribe Send Feedback

For integration with the FPGA fabric, the full-duplex transceiver channel supports custom configuration with physical medium attachment (PMA), physical coding sublayer (PCS), and low latency custom configurations with the PMA and low latency PCS.

You can customize the transceiver with one of the following configurations:

- Standard PCS— Physical coding sublayer (PCS) and physical medium attachment (PMA)
- Standard PCS in low latency mode— Low latency PCS and PMA
- PMA Direct— PMA only

Figure 5-1: Custom Configuration Options


Up to 10.3125 Gbps in 10-Gbps channel
(in GT and ST device variants)
Up to 6.5536 Gbps with PCS (Standard PCS and low latency)
HIP
PIPE
PCS
PMA
tx_serial_data
rx_serial_data
FPGA Fabric
Transceiver Resources
Serial Interface

Related Information

Arria V Device Handbook: Known Issues

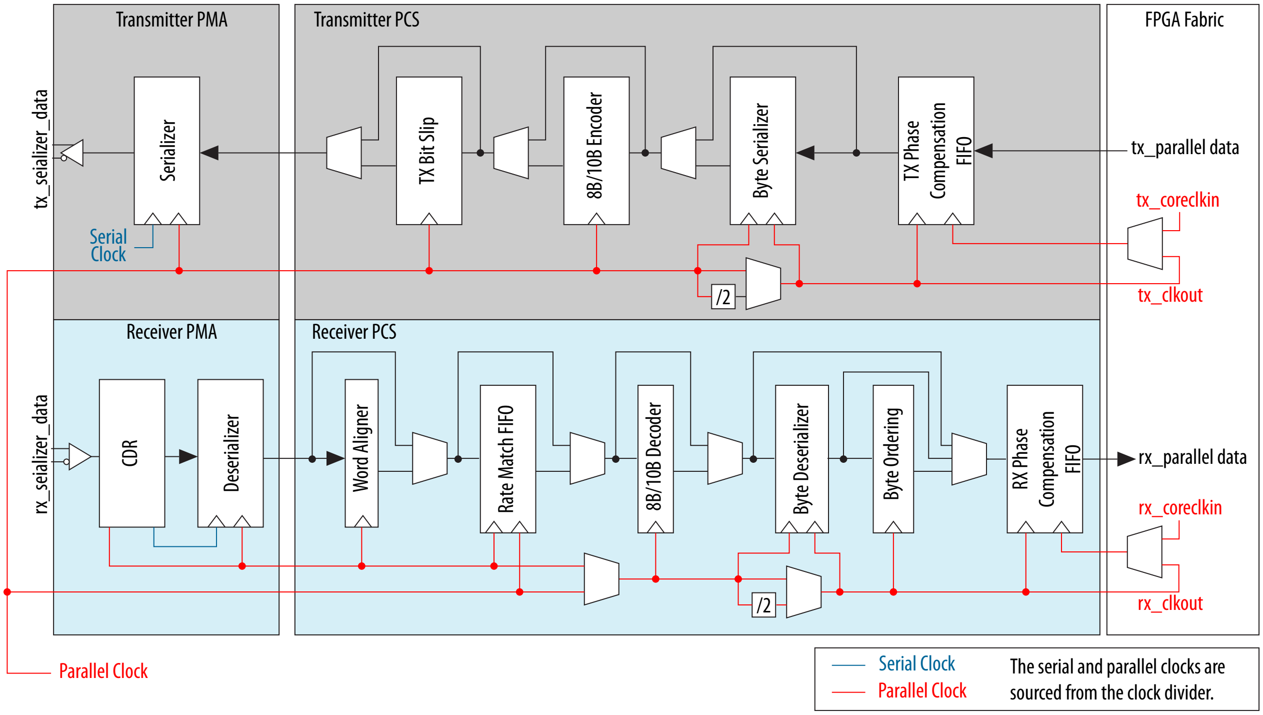
Lists the planned updates to the *Arria V Device Handbook* chapters.

Standard PCS Configuration

In this configuration, you can customize the transceiver channel to include a PMA and PCS with functions that your application requires. The transceiver channel interfaces with the FPGA fabric through the PCS.

Figure 5-2: Complete Datapath in a Custom Configuration

Based on your application requirements, you can enable, modify, or disable the blocks, as shown in the following figure.



Custom Configuration Channel Options

There are multiple channel options when you use Custom Configuration.

The supported interface width varies depending on the usage of the byte serializer/deserializer (SERDES), and the 8B/10B encoder or decoder. The byte serializer or deserializer is assumed to be enabled. Otherwise, the maximum data rate supported is half of the specified value.

The maximum supported data rate varies depending on the customization.

Table 5-1: Maximum Supported Data Rate for Fastest Speed Grade Device (–4 Commercial) in Custom Configuration

The following table lists an example of the maximum supported data rate in various custom configurations with the PMA and PCS using the fastest speed grade device.

Data Configuration	PMA-PCS Interface Width	PCS-FPGA Fabric Interface Width		Maximum Data Rate (Mbps)
		8B/10B Enabled	8B/10B Disabled	
Single-width	8	—	8	1,500
			16	3,000
	10	8	10	1,875
		16	20	3,750
Double-width	16	—	16	2,621.44
			32	5,242.88
	20	16	20	3,276.8
		32	40	6,553.6

In all the supported configuration options of the channel, the transmitter bit-slip function is optional, where:

- The blocks shown as “Disabled” are not used but incur latency.
- The blocks shown as “Bypassed” are not used and do not incur any latency.
- The transmitter bit-slip is disabled.

Figure 5-3: Configuration Options for Custom Single-Width Mode (8-bit PMA–PCS Interface Width)

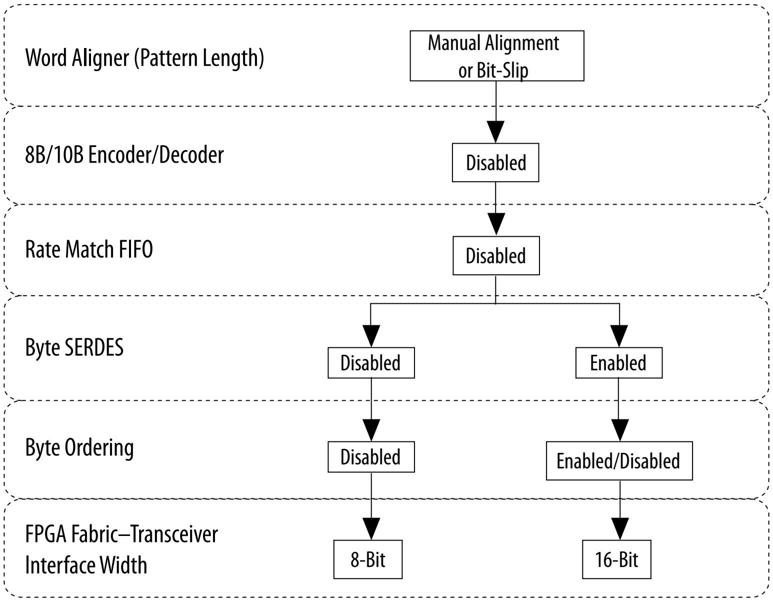


Figure 5-4: Configuration Options for Custom Single-Width Mode (10-bit PMA–PCS Interface Width)

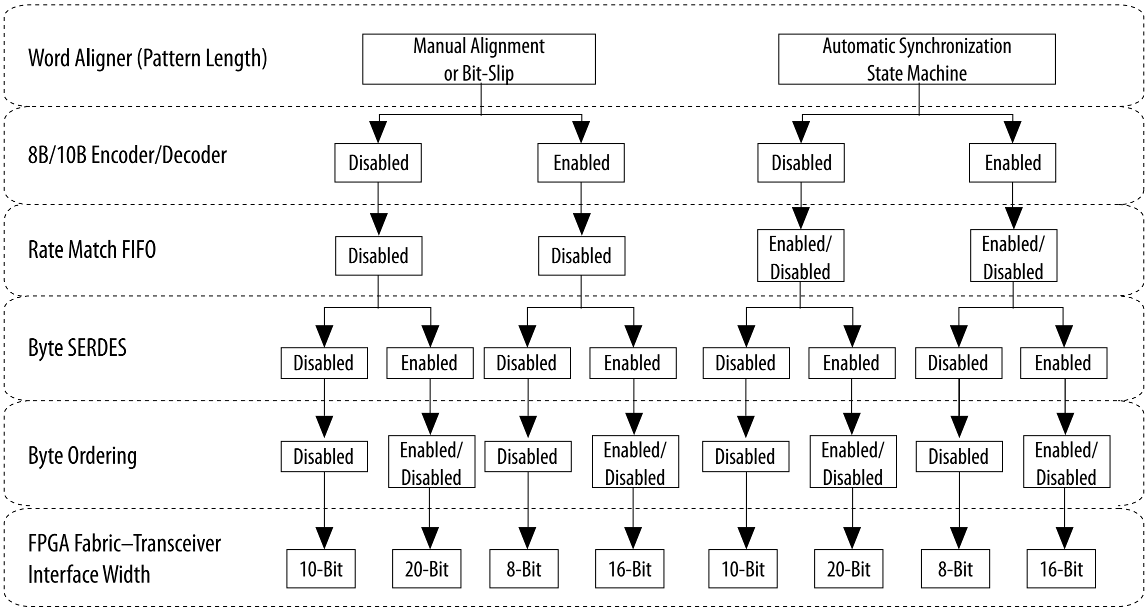


Figure 5-5: Configuration Options for Custom Double-Width Mode (16-bit PMA–PCS Interface Width)

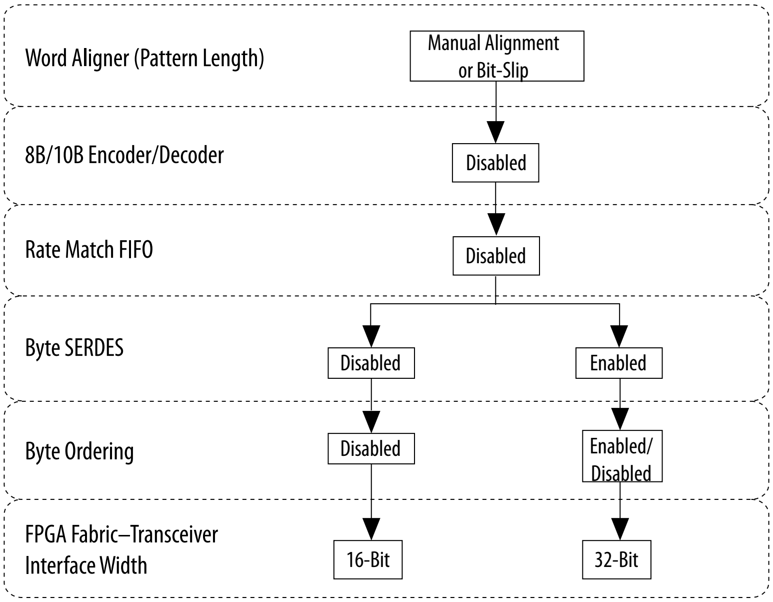
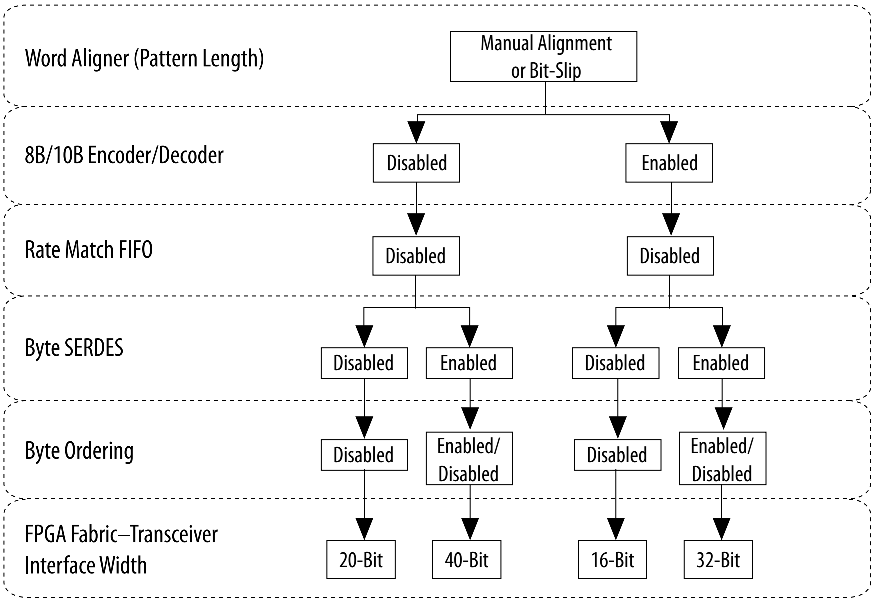


Figure 5-6: Configuration Options for Custom Double-Width Mode (20-bit PMA–PCS Interface Width)



Rate Match FIFO in Custom Configuration

In a custom configuration, the 20-bit pattern for the rate match FIFO is user-defined. The FIFO operates by looking for the 10-bit control pattern followed by the 10-bit skip pattern in the data, after the word aligner restores the word boundary. After finding the pattern, the FIFO performs a skip pattern insertion or deletion to ensure that the FIFO does not underflow or overflow a given parts per million (ppm) difference between the clocks.

The rate match FIFO operation requires 8B/10B-coded data.

Rate Match FIFO Behaviors in Custom Single-Width Mode

The different operations available in custom single-width mode for the rate match FIFO are symbol insertion, symbol deletion, full condition, and empty condition.

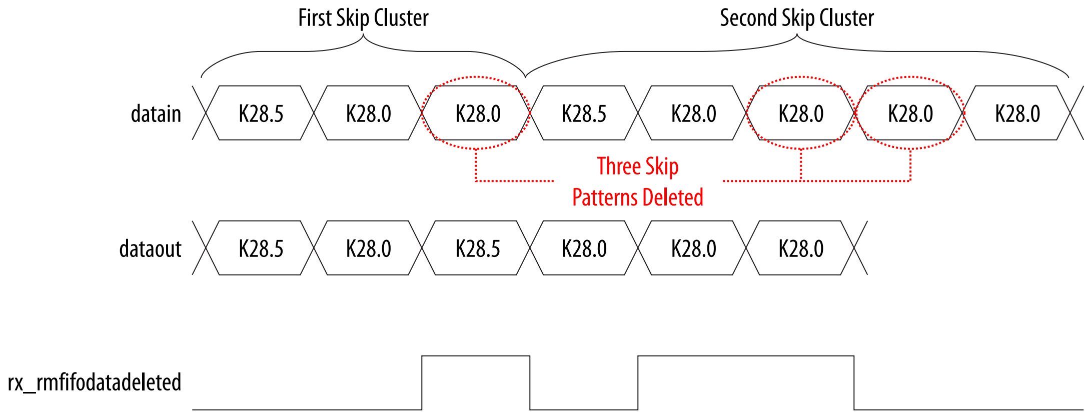
Table 5-2: Rate Match FIFO Behaviors in Custom Single-Width Mode (10-bit PMA–PCS Interface Width)

Operation	Behavior
Symbol Insertion	Inserts a maximum of four skip patterns in a cluster, only if there are no more than five skip patterns in the cluster after the symbol insertion.
Symbol Deletion	Deletes a maximum of four skip patterns in a cluster, only if there is one skip pattern left in the cluster after the symbol deletion.
Full Condition	Deletes the data byte that causes the FIFO to go full.
Empty Condition	Inserts a /K30.7/ (9'h1FE) after the data byte that caused the FIFO to go empty.

In this example, /K28.5/ is the control pattern and neutral disparity /K28.0/ is the skip pattern. The first skip cluster has a /K28.5/ control pattern followed by two /K28.0/ skip patterns. The second skip cluster has a /K28.5/ control pattern followed by four /K28.0/ skip patterns. The rate match FIFO deletes only one /K28.0/ skip pattern from the first skip cluster to maintain at least one skip pattern in the cluster after deletion. Two /K28.0/ skip patterns are deleted from the second cluster for the three skip pattern deletion requirement.

Figure 5-7: Rate Match Deletion in Custom Single-Width Mode

The following figure shows an example of rate match FIFO deletion in the case where three skip patterns are required to be deleted.



In this example, /K28.5/ is the control pattern and neutral disparity /K28.0/ is the skip pattern. The first skip cluster has a /K28.5/ control pattern followed by three /K28.0/ skip patterns. The second skip cluster has a /K28.5/ control pattern followed by one /K28.0/ skip pattern. The rate match FIFO inserts only two /K28.0/ skip patterns into the first skip cluster to maintain a maximum of five skip patterns in the cluster after insertion. One /K28.0/ skip pattern is inserted into the second cluster for a total of three skip patterns to meet the insertion requirement.

Figure 5-8: Rate Match Insertion in Custom Single-Width Mode

The following figure shows an example of rate match FIFO insertion in the case where three skip patterns are required to be inserted.

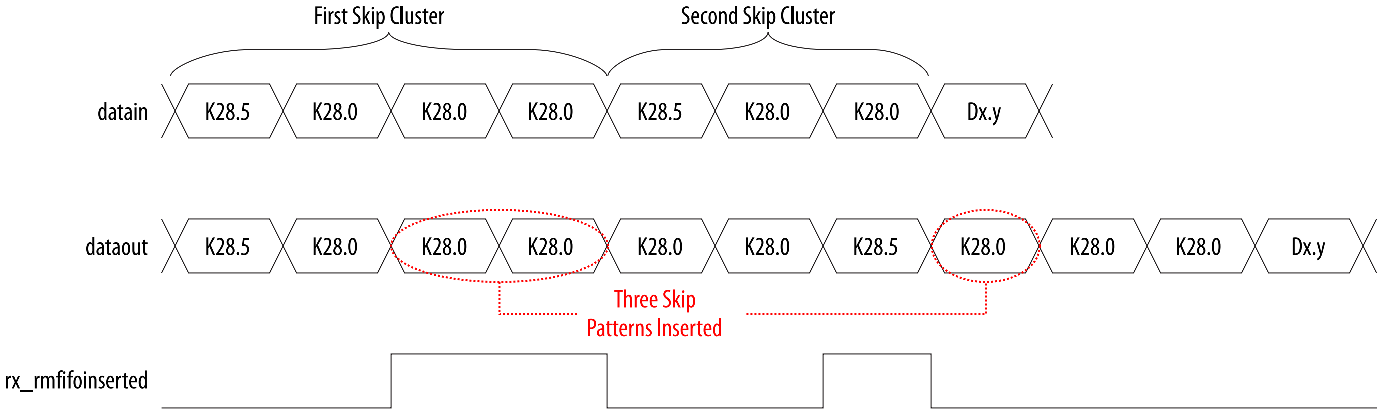


Figure 5-9: Rate Match FIFO Full Condition in Custom Single-Width Mode

The following figure shows the rate match FIFO full condition in custom single-width mode. The rate match FIFO becomes full after receiving data byte D4.

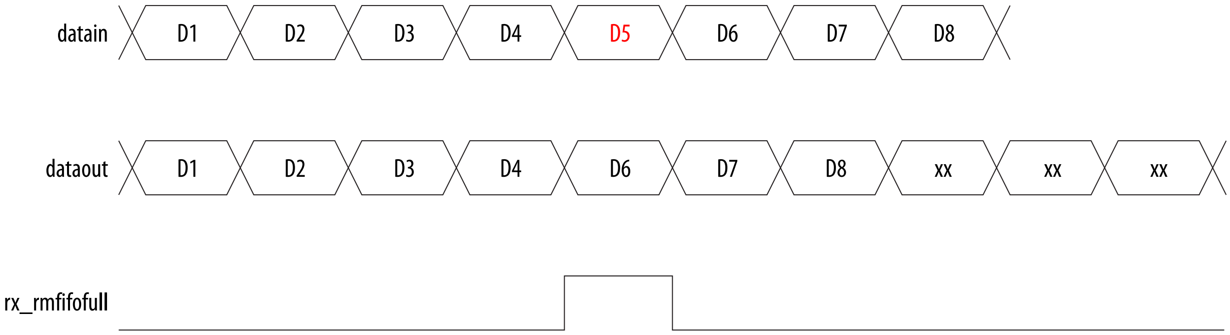
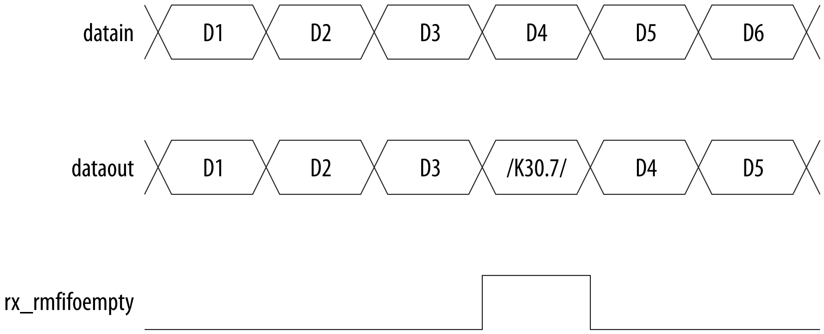


Figure 5-10: Rate Match FIFO Empty Condition in Custom Single-Width Mode

The following figure shows the rate match FIFO empty condition in custom single-width mode. The rate match FIFO becomes empty after reading out data byte D3.



Rate Match FIFO Behaviors in Custom Double-Width Mode

The different operations available in custom double-width mode for the rate match FIFO are symbol insertion, symbol deletion, full condition, and empty condition.

Table 5-3: Rate Match FIFO Behaviors in Custom Double-Width Mode (20-bit PMA–PCS Interface Width)

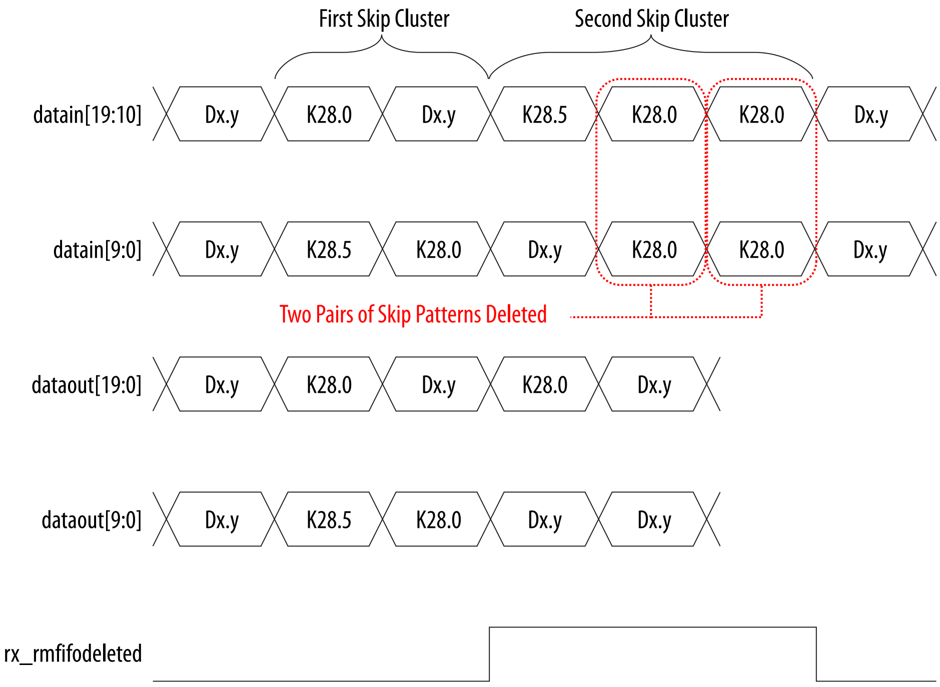
Operation	Behavior
Symbol Insertion	Inserts as many pairs (10-bit skip patterns at the LSByte and MSByte of the 20-bit word at the same clock cycle) of skip patterns as needed.
Symbol Deletion	Deletes as many pairs (10-bit skip patterns at the LSByte and MSByte of the 20-bit word at the same clock cycle) of skip patterns as needed.
Full Condition	Deletes the pair (20-bit word) of data bytes that causes the FIFO to go full.

Operation	Behavior
Empty Condition	Inserts a pair of /K30.7/ ({9'h1FE, 9'h1FE}) after the data byte that causes the FIFO to go empty.

In this example, /K28.5/ is the control pattern and neutral disparity /K28.0/ is the skip pattern. The first skip cluster has a /K28.5/ control pattern in the LSByte and /K28.0/ skip pattern in the MSByte of a clock cycle followed by one /K28.0/ skip pattern in the LSByte of the next clock cycle. The rate match FIFO cannot delete the two skip patterns in this skip cluster because they do not appear in the same clock cycle. The second skip cluster has a /K28.5/ control pattern in the MSByte of a clock cycle followed by two pairs of /K28.0/ skip patterns in the next two cycles. The rate match FIFO deletes both pairs of /K28.0/ skip patterns (for a total of four skip patterns deleted) from the second skip cluster to meet the three skip pattern deletion requirement.

Figure 5-11: Rate Match Deletion in Custom Double-Width Mode

The following figure shows an example of rate match FIFO deletion in the case where three skip patterns are required to be deleted.



In this example, /K28.5/ is the control pattern and neutral disparity /K28.0/ is the skip pattern. The first skip cluster has a /K28.5/ control pattern in the LSByte and /K28.0/ skip pattern in the MSByte of a clock cycle followed by one /K28.0/ skip pattern in the LSByte of the next clock cycle. The rate match FIFO inserts pairs of skip patterns in this skip cluster to meet the three skip pattern insertion requirement.

Figure 5-12: Rate Match Insertion in Custom Double-Width Mode

The following figure shows an example of rate match FIFO insertion in the case where three skip patterns are required to be inserted.

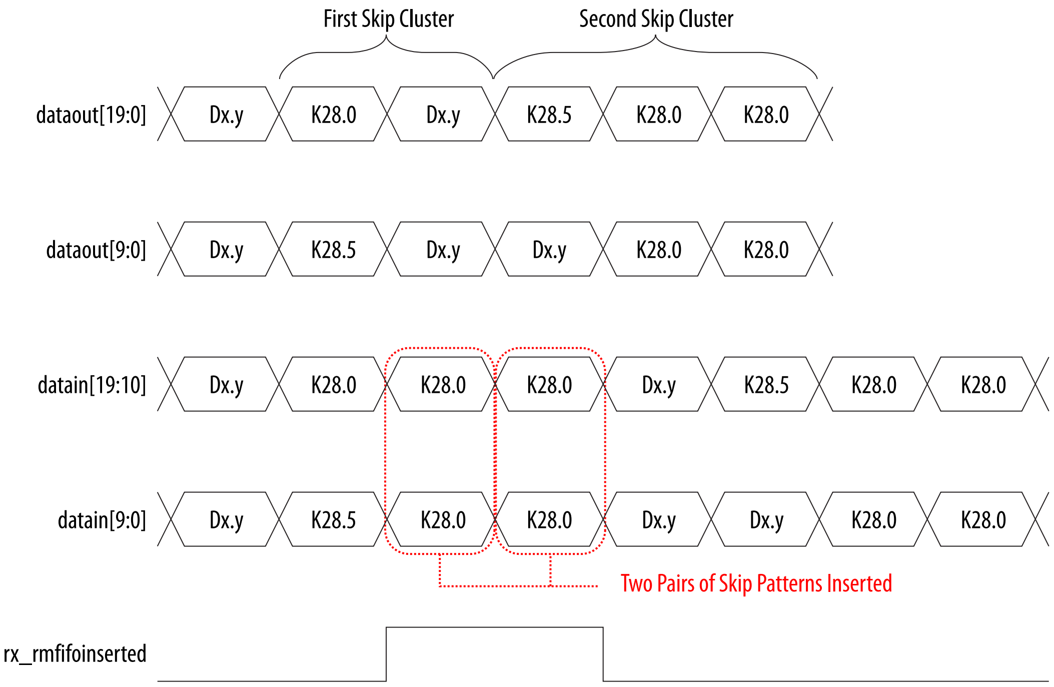


Figure 5-13: Rate Match FIFO Full Condition in Custom Double-Width Mode

The following figure shows the rate match FIFO full condition in custom double-width mode. The rate match FIFO becomes full after receiving the 20-bit word D5D6.

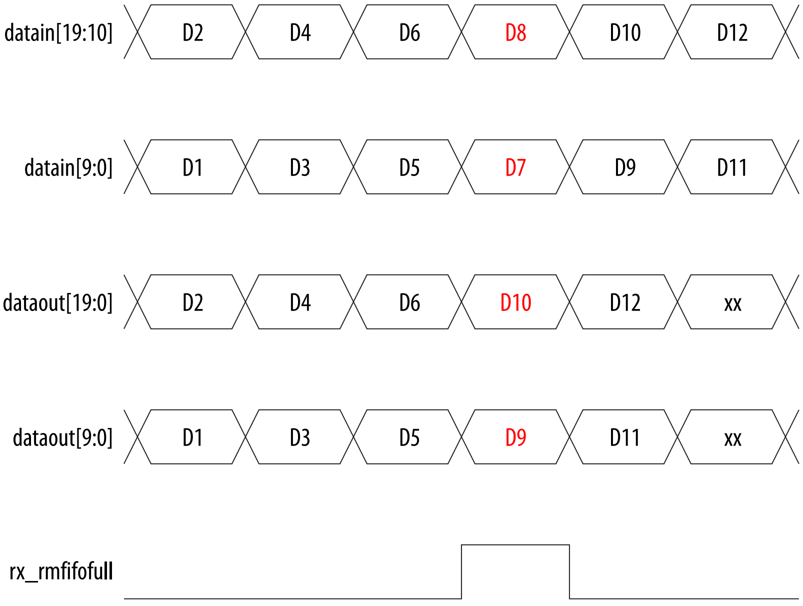
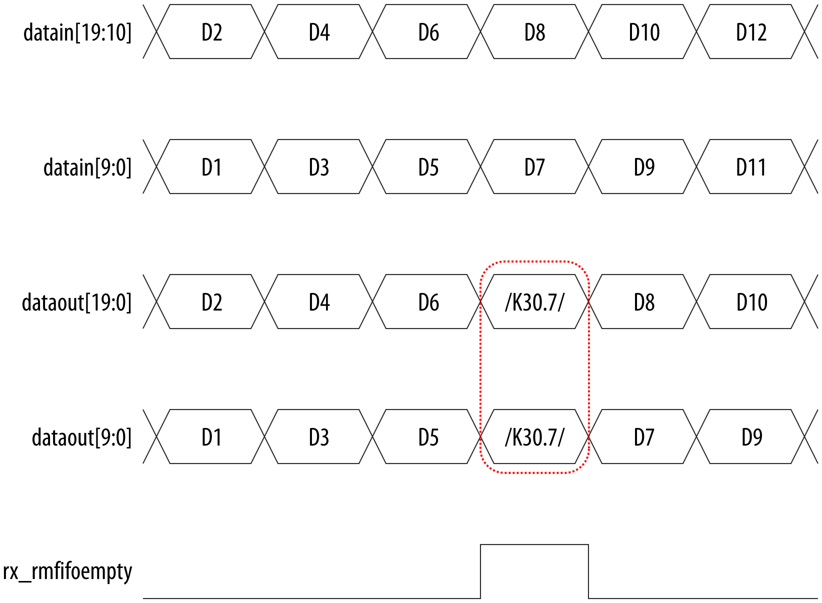


Figure 5-14: Rate Match FIFO Empty Condition in Custom Double-Width Mode

The following figure shows the rate match FIFO empty condition in custom double-width mode. The rate match FIFO becomes empty after reading out the 20-bit word D5D6.

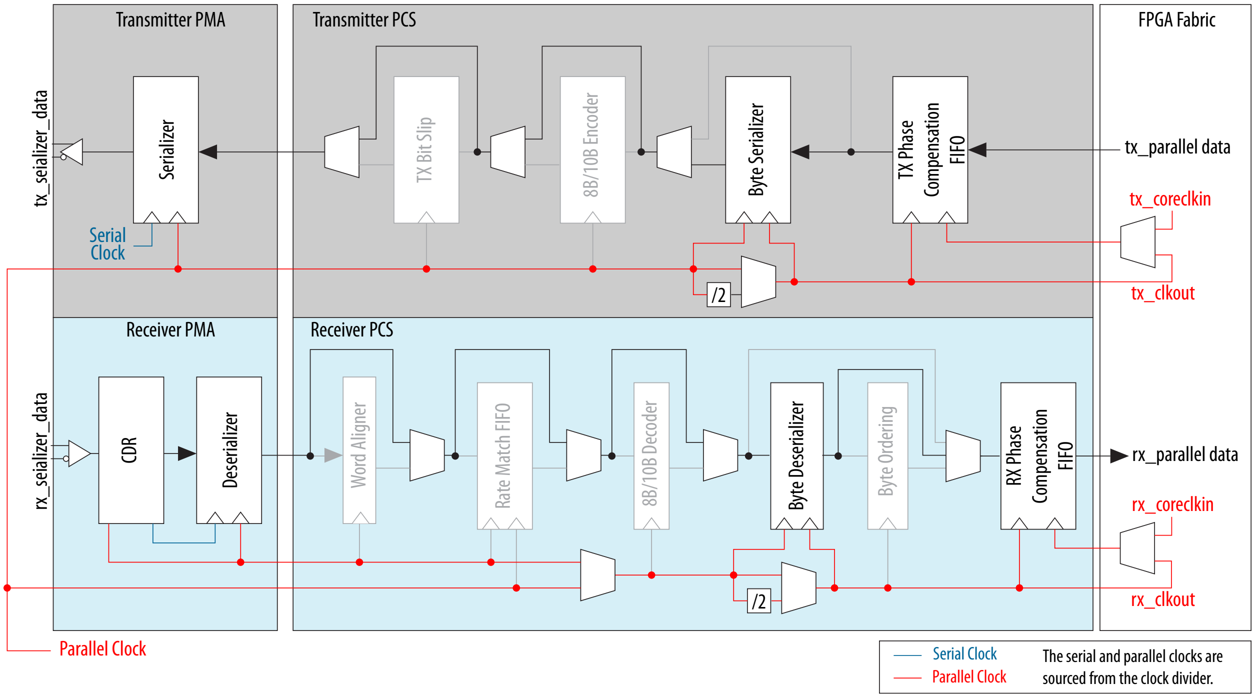


Standard PCS in Low Latency Configuration

In this configuration, you can customize the transceiver channel to include a PMA and PCS that bypasses most of the PCS logical functionality for a low latency datapath.

To provide a low latency datapath, the PCS includes only the phase compensation FIFO in phase compensation mode, and optionally, the byte serializer and byte deserializer blocks, as shown in the following figure. The transceiver channel interfaces with the FPGA fabric through the PCS.

Figure 5-15: Datapath in Low Latency Custom Configuration



The maximum supported data rate varies depending on the customization and is identical to the custom configuration except that the 8B/10B block is disabled

Low Latency Custom Configuration Channel Options

There are multiple channel options when you use Low Latency Custom Configuration.

In the following figures:

- The blocks shown as “Disabled” are not used but incur latency.
- The blocks shown as “Bypassed” are not used and do not incur any latency.
- The transmitter bit-slip is disabled.

Figure 5-16: Configuration Options for Low Latency Custom Single-Width Mode (8-bit PMA–PCS Interface Width)

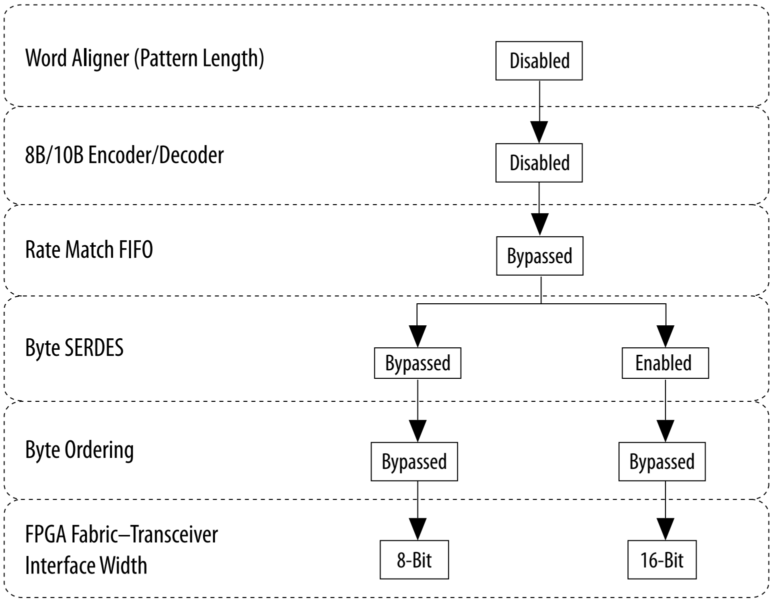


Figure 5-17: Configuration Options for Low Latency Custom Single-Width Mode (10-bit PMA–PCS Interface Width)

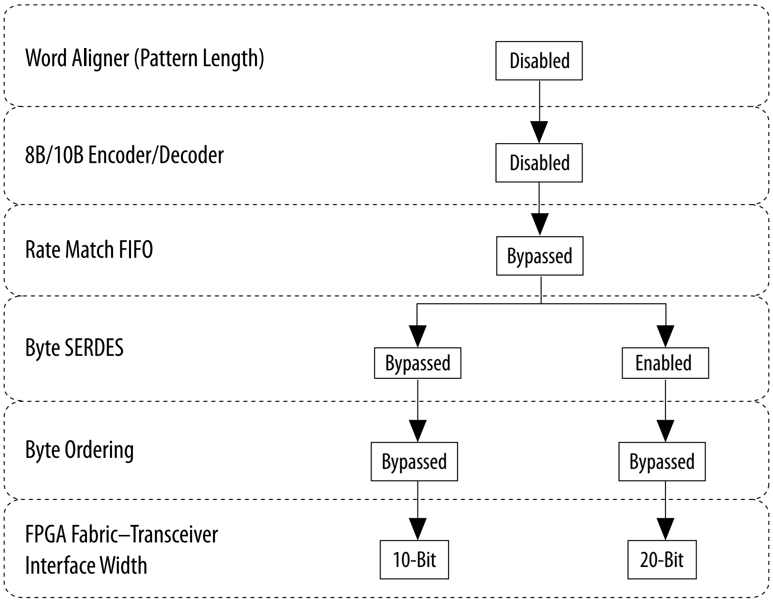


Figure 5-18: Configuration Options for Low Latency Custom Double-Width Mode (16-bit PMA–PCS Interface Width)

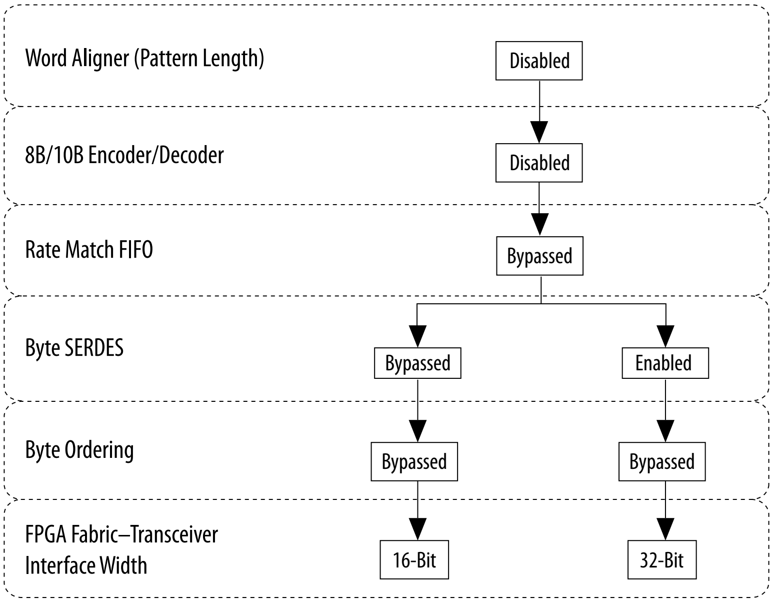
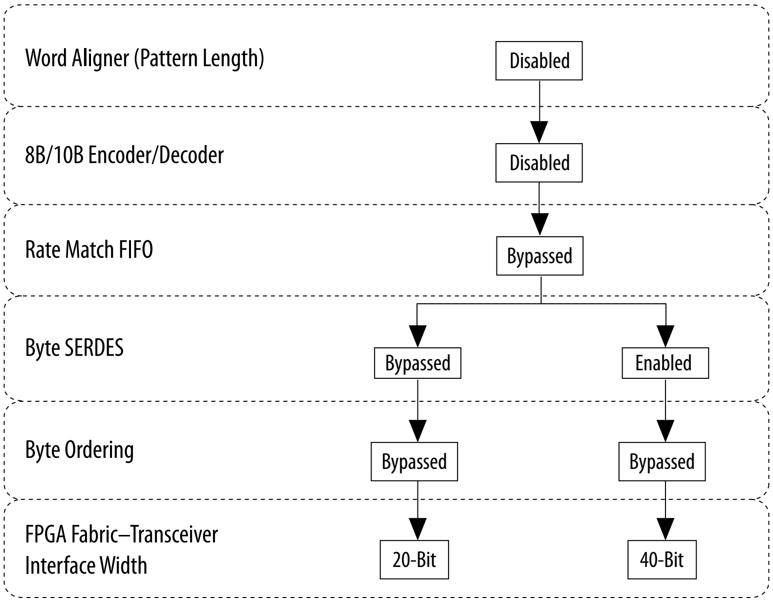


Figure 5-19: Configuration Options for Low Latency Custom Double-Width Mode (20-bit PMA–PCS Interface Width)

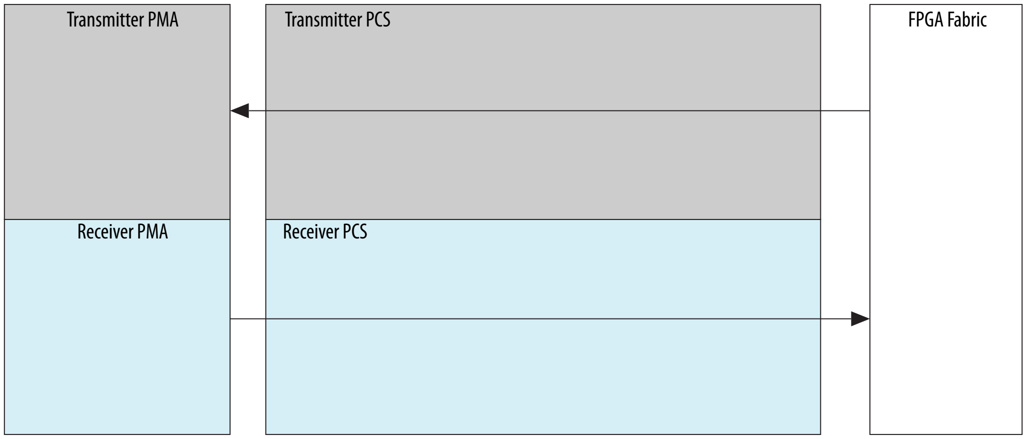


PMA Direct

You can customize the transceiver channel to include only the PMA, in a PMA Direct configuration for serial data rates up to 10.3125 Gbps.

In this configuration, the serializer and deserializer interface directly to the FPGA fabric, bypassing the PCS. The serializer and deserializer support 8-, 10-, 16-, 20-, 64-, and 80-bit configurations. You must implement the PCS functions in the FPGA fabric.

Figure 5-20: Transceiver Datapath in a PMA Direct Configuration



Document Revision History

Table 5-4: Document Revision History

Date	Version	Changes
September 2014	2014.09.30	Updated the "Configuration Options for Custom Double-Width Mode (20-bit PMA-PCS Interface Width)" figure with change to Rate Match FIFO row.
March 2014	2014.03.07	Updated "Maximum Supported Data Rate for Fastest Speed Grade Device (–4 Commercial) in Custom Configuration" table.
May 2013	2013.05.06	Added link to the known document issues in the Knowledge Base.
November 2012	2012.11.19	Reorganized content and updated template.
June 2012	1.2	• Updated for the Quartus II software version 12.0. • Updated the “PCS Datapath Latency” section. • Updated the “PMA Direct Configuration” section. • Updated Figure 5–1 and Figure 5–14.

Date	Version	Changes
November 2011	1.1	Updated for the Quartus II software version 11.1.
August 2011	1.0	Initial release.

Transceiver Configurations in Arria V GZ Devices 6

2014.09.30

AV53008 Subscribe Send Feedback

Arria® V GZ devices have a dedicated transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) circuitry.

To implement a protocol, use a PHY IP listed in **Table 6-12**.

Arria V GZ devices support the following communication protocols:

- 10GBASE-R and 10GBASE-KR
- Interlaken
- PCI Express® (PCIe®)—Gen1, Gen2, and Gen3
- CPRI and OBSAI—Deterministic Latency Protocols
- XAUI

Support for other communication protocols or user-defined protocols can be enabled with the following PHY IPs:

- Native PHY IP using standard PCS and 10G PCS hardware options including reconfigurability between different PCS options
- Custom PHY IP using the standard PCS in a custom datapath
- Low Latency PHY IP using the standard or 10G PCS in a low latency datapath configuration

Related Information

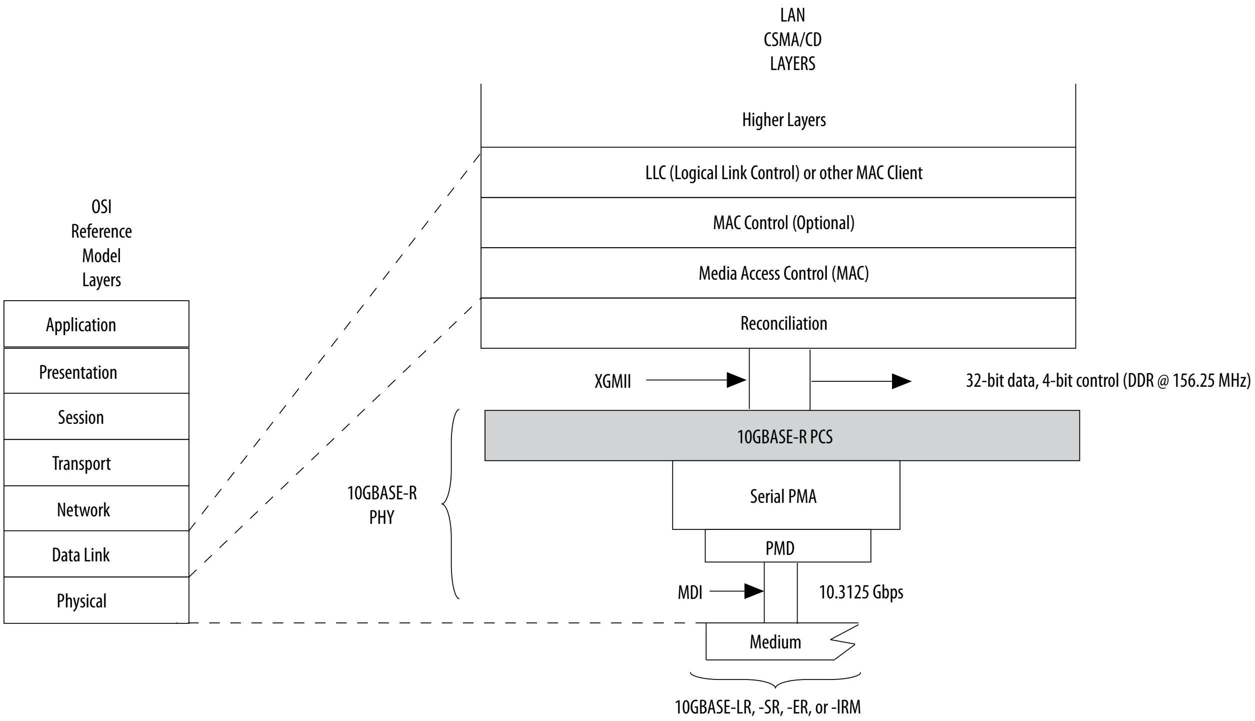
- **Arria V Device Handbook: Known Issues**
  Lists the planned updates to the *Arria V Device Handbook* chapters.
- **Upcoming Arria V Device Features**
- **Altera Transceiver PHY IP Core User Guide**

10GBASE-R and 10GBASE-KR

10GBASE-R is used in optical module LAN applications such as optical routers, servers, and switches, and 10GBASE-KR is used in electrical backplane applications such as blade servers using Arria V GZ transceivers.

10GBASE-R is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in clause 49 of the IEEE 802.3-2008 specification. The 10GBASE-R PHY uses the XGMII interface to connect to the IEEE802.3 media access control (MAC) and reconciliation sublayer (RS). The IEEE 802.3-2008 specification requires each 10GBASE-R link to support a 10 Gbps data rate at the XGMII interface and a 10.3125 Gbps serial line rate with 64B/66B encoding.

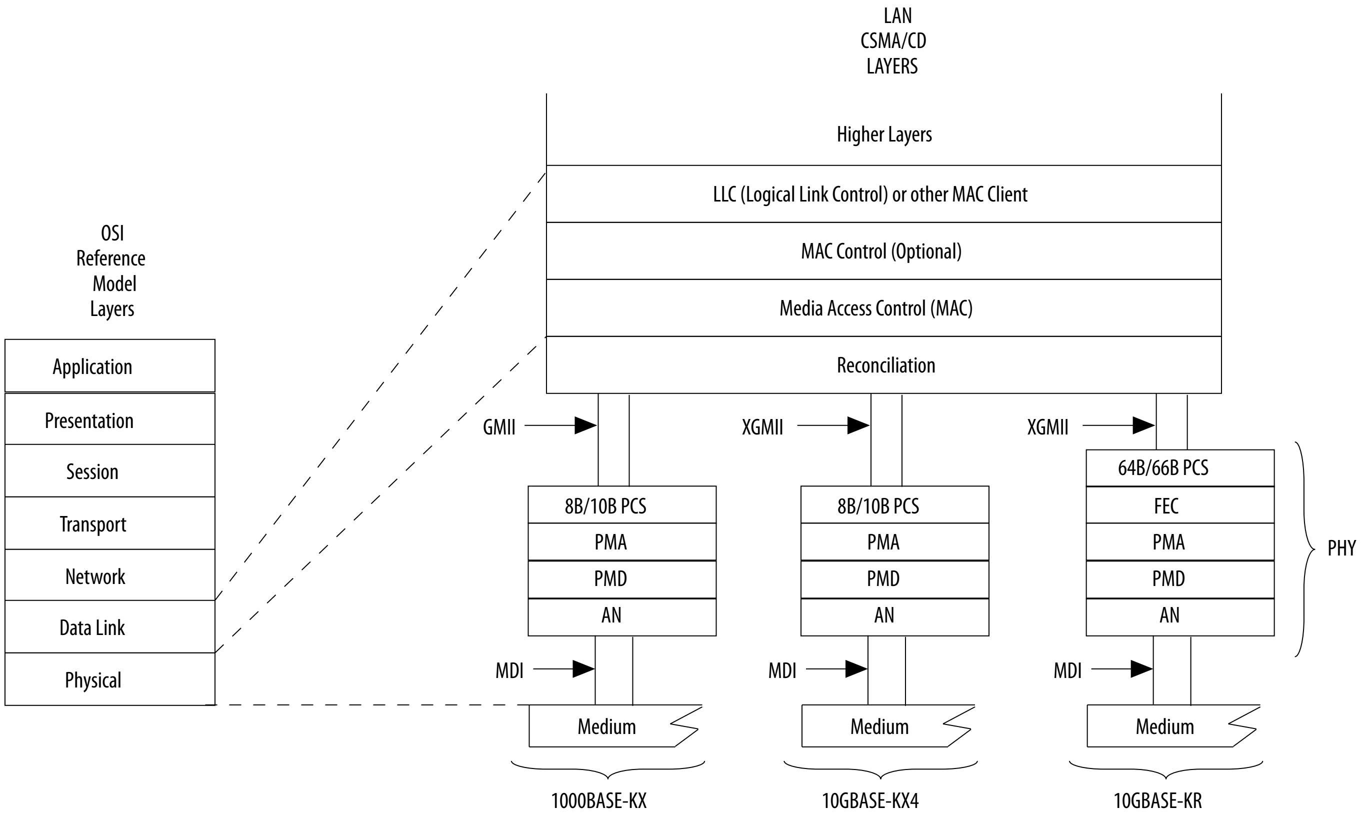
Figure 6-1: 10GBASE-R PHY Connection to IEEE802.3 MAC and RS



Note: To implement a 10GBASE-R link, instantiate the **10GBASE-R PHY IP** core in the IP Catalog, under **Ethernet** in the Interfaces menu.

The IEEE 802.3ap-2007 specification also requires each backplane link to support multi-data rates of 1Gbps and 10 Gbps speeds. 10GBASE-KR and 1000BASE-KX is the electrical backplane physical layer implementation for the 10 Gigabit and 1 Gigabit Ethernet link defined in clause 72 and clause 70 respectively of the IEEE 802.3ap-2007 specification. The 10 Gbps backplane ethernet 10GBASE-KR implementation uses the XGMII interface to connect to the reconciliation sublayer (RS) with 64B/66B PCS encoding, the optional Forward Error Correction (FEC), and Auto-Negotiation (AN) support to the Highest Common Denominator (HCD) technology with the partner link. The optional FEC, LT, and AN logic is implemented in the core fabric. The 1Gbps backplane ethernet 1000BASE-KX implementation uses the GMII interface to connect to the reconciliation sublayer (RS) with 8B/10B PCS encoding and Auto-Negotiation support to the HCD technology with the partner link.

Figure 6-2: 10GBASE-KR PHY Connection to IEEE802.3 MAC and RS



Note: To implement a 10GBASE-KR link with 1000BASE-KX support, instantiate the **1G/10GbE PHY IP** and **10GBASE-KR PHY IP** cores in the IP Catalog, under **Ethernet** in the Interfaces menu.

An additional license is required in order to use the 1G/10GbE and 10GBASE-KR PHY IP Core which also supports 10GBASE-R and 1000BASE-X links and auto-negotiation between the 10 Gigabit and 1 Gigabit Ethernet data rates.

Related Information

- Altera Transceiver PHY IP Core User Guide
- 10-Gbps Ethernet MAC MegaCore Function User Guide

10GBASE-R and 10GBASE-KR Transceiver Datapath Configuration

The following figures show the transceiver blocks and settings enabled in 10GBASE-R and 10GBASE-KR configurations.

10GBASE-R

Figure 6-3: 10GBASE-R Datapath Configuration

The blocks shown as "Disabled" are not used, but incur latency. The blocks shown as "Bypassed" are not used and do not incur latency.

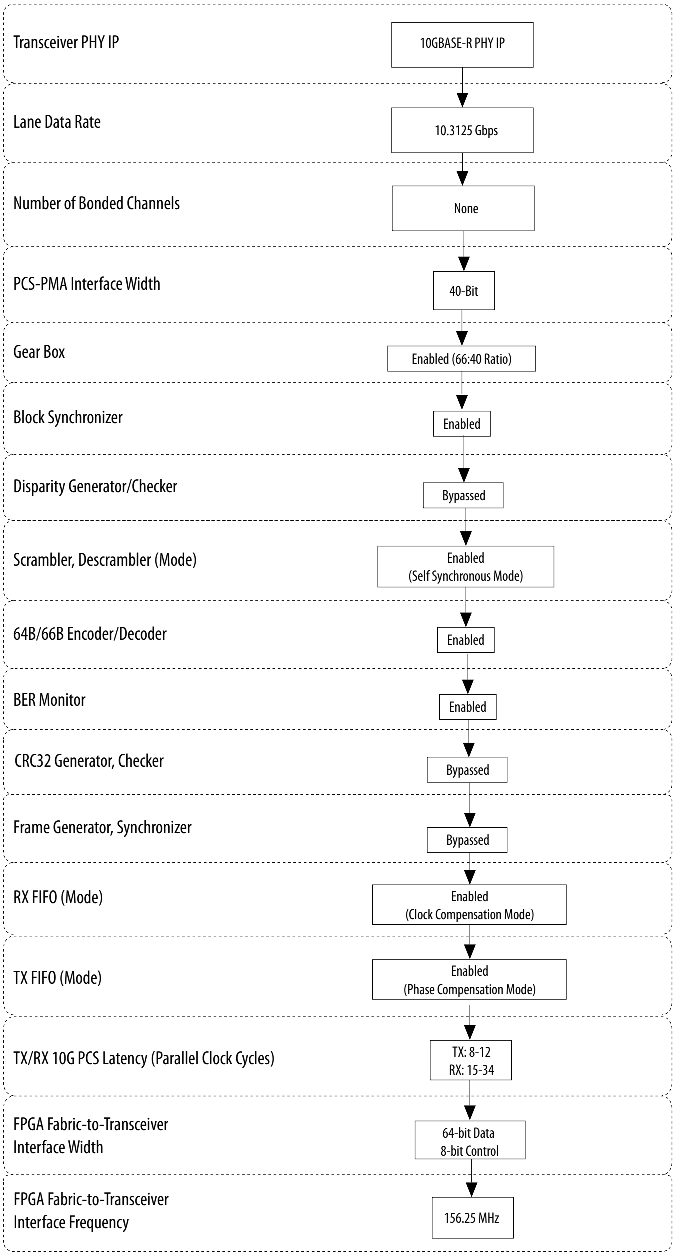
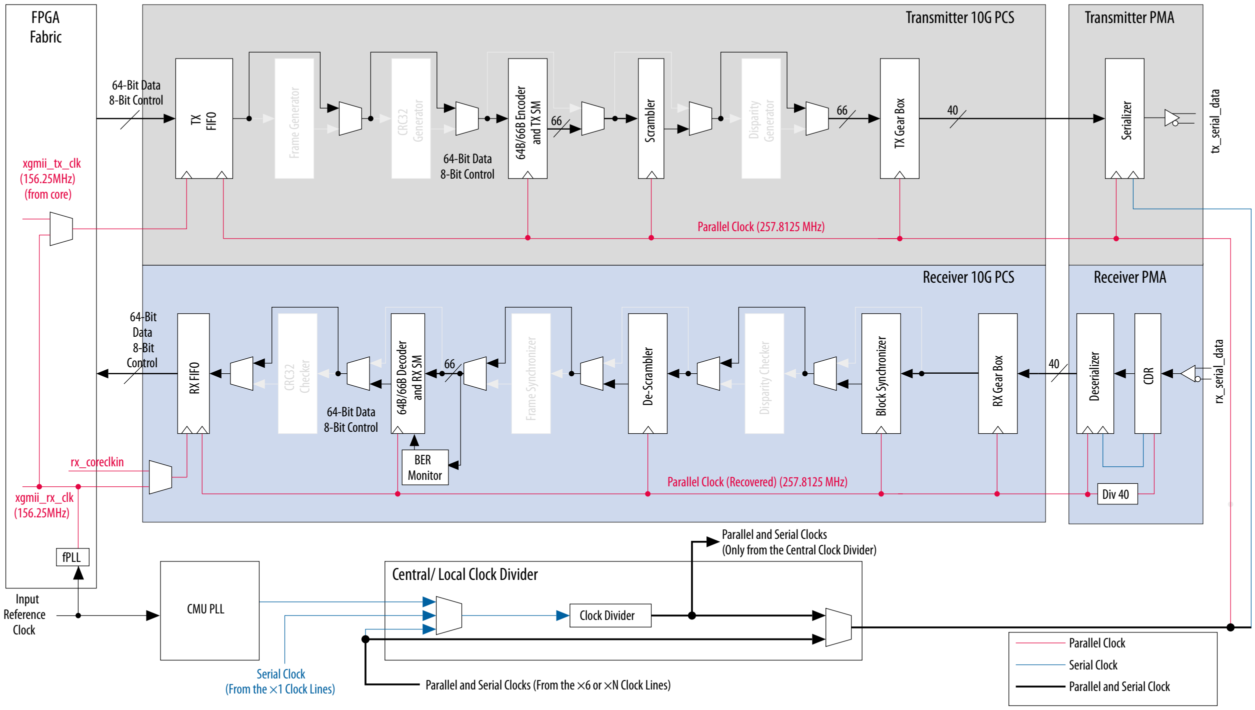


Figure 6-4: Transceiver Channel Datapath for a 10GBASE-R Configuration



10GBASE-KR

Figure 6-5: 10GBASE-R/KR and 1000Base-X/KX Datapath Configuration

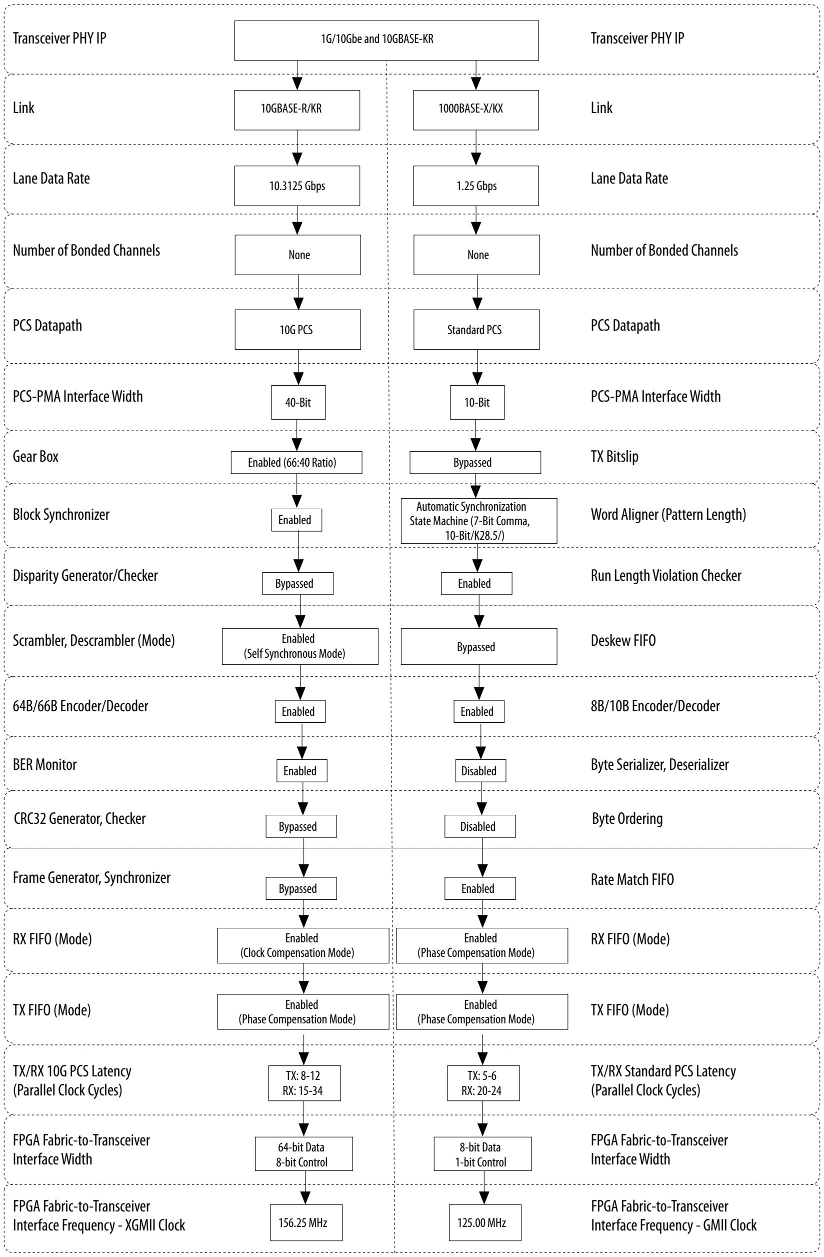
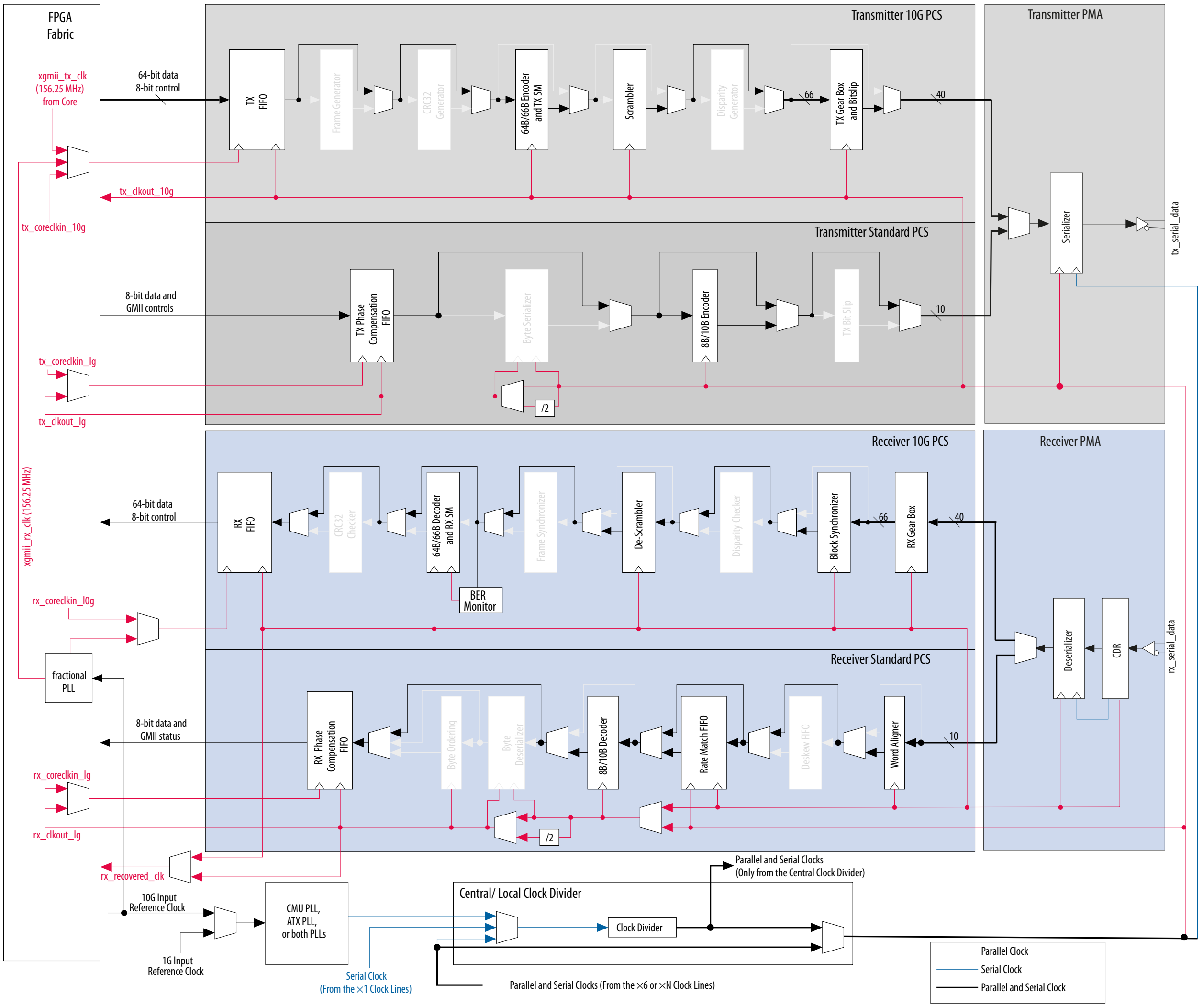

Transceiver PHY IP
1G/10Gbe and 10GBASE-KR
Transceiver PHY IP
Link
10GBASE-R/KR
1000BASE-X/KX
Link
Lane Data Rate
10.3125 Gbps
1.25 Gbps
Lane Data Rate
Number of Bonded Channels
None
None
Number of Bonded Channels
PCS Datapath
10G PCS
Standard PCS
PCS Datapath
PCS-PMA Interface Width
40-Bit
10-Bit
PCS-PMA Interface Width
Gear Box
Enabled (66:40 Ratio)
Bypassed
TX Bitslip
Block Synchronizer
Enabled
Automatic Synchronization State Machine (7-Bit Comma, 10-Bit/K28.5/)
Word Aligner (Pattern Length)
Disparity Generator/Checker
Bypassed
Enabled
Run Length Violation Checker
Scrambler, Descrambler (Mode)
Enabled (Self Synchronous Mode)
Bypassed
Deskew FIFO
64B/66B Encoder/Decoder
Enabled
Enabled
8B/10B Encoder/Decoder
BER Monitor
Enabled
Disabled
Byte Serializer, Deserializer
CRC32 Generator, Checker
Bypassed
Disabled
Byte Ordering
Frame Generator, Synchronizer
Bypassed
Enabled
Rate Match FIFO
RX FIFO (Mode)
Enabled (Clock Compensation Mode)
Enabled (Phase Compensation Mode)
RX FIFO (Mode)
TX FIFO (Mode)
Enabled (Phase Compensation Mode)
Enabled (Phase Compensation Mode)
TX FIFO (Mode)
TX/RX 10G PCS Latency (Parallel Clock Cycles)
TX: 8-12 RX: 15-34
TX: 5-6 RX: 20-24
TX/RX Standard PCS Latency (Parallel Clock Cycles)
FPGA Fabric-to-Transceiver Interface Width
64-bit Data 8-bit Control
8-bit Data 1-bit Control
FPGA Fabric-to-Transceiver Interface Width
FPGA Fabric-to-Transceiver Interface Frequency - XGMII Clock
156.25 MHz
125.00 MHz
FPGA Fabric-to-Transceiver Interface Frequency - GMII Clock

Figure 6-6: Transceiver Channel Datapath for 10GBASE-R/KR and 1000BASE-X/KX Configuration



10GBASE-R and 10GBASE-KR Supported Features

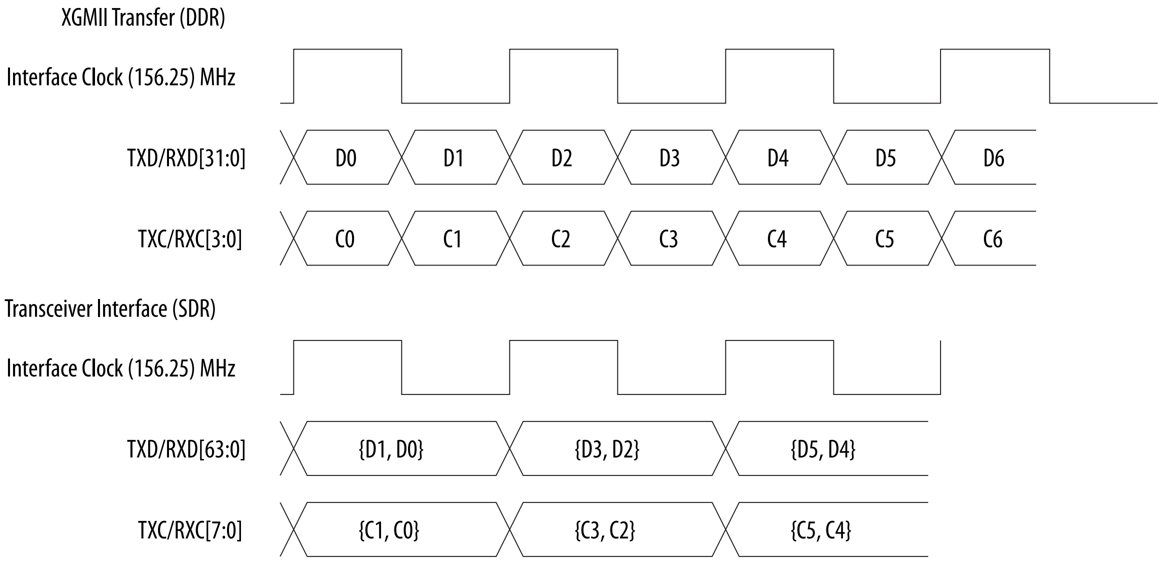
The following features are supported by the transceivers in 10GBASE-R and 10GBASE-KR configurations.

64-Bit Single Data Rate (SDR) Interface to the MAC/RS in 10GBASE-R and 10GBASE-KR Configurations

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the 10GBASE-R and 10GBASE-KR PCS and the Ethernet MAC/RS. The XGMII interface defines the 32-bit data and 4-bit wide control character clocked between the MAC/RS and the PCS at both the positive and negative edge (double data rate – DDR) of the 156.25 MHz interface clock.

The transceivers do not support the XGMII interface to the MAC/RS as defined in the IEEE 802.3-2008 specification. Instead, they support a 64-bit data and 8-bit control SDR interface between the MAC/RS and the PCS.

Figure 6-7: XGMII Interface (DDR) versus Arria V GZ Transceiver Interface (SDR) for 10GBASE-R and 10GBASE-KR Configurations



64B/66B Encoding/Decoding in 10GBASE-R and 10GBASE-KR Configurations

The transceivers in 10GBASE-R and 10GBASE-KR configurations support 64B/66B encoding and decoding as specified in Clause 49 of the IEEE802.3-2008 specification. The 64B/66B encoder receives 64-bit data and 8-bit control code from the transmitter FIFO and converts it into 66-bit encoded data. The 66-bit encoded data contains two overhead sync header bits that the receiver PCS uses for block synchronization and bit-error rate (BER) monitoring.

The 64B/66B encoding also ensures enough transitions on the serial data stream for the receiver clock data recovery (CDR) to maintain its lock on the incoming data.

Transmitter and Receiver State Machines in 10GBASE-R and 10GBASE-KR Configurations

The transceivers in 10GBASE-R and 10GBASE-KR configurations implement the transmitter and receiver state diagrams shown in Figure 49-14 and Figure 49-15 of the IEEE802.3-2008 specification.

Besides encoding the raw data specified in the 10GBASE-R and 10GBASE-KR PCS, the transmitter state diagram performs functions such as transmitting local faults (LBLOCK_T) under reset, as well as transmitting error codes (EBLOCK_T) when the 10GBASE-R PCS rules are violated.

Besides decoding the incoming data specified in the 10GBASE-R and 10GBASE-KR PCS, the receiver state diagram performs functions such as sending local faults (LBLOCK_R) to the MAC/RS under reset and substituting error codes (EBLOCK_R) when the 10GBASE-R and 10GBASE-KR PCS rules are violated.

Block Synchronizer in 10GBASE-R and 10GBASE-KR Configurations

The block synchronizer in the receiver PCS determines when the receiver has obtained lock to the received data stream. It implements the lock state diagram shown in Figure 49-12 of the IEEE 802.3-2008 specification.

The block synchronizer provides a status signal to indicate whether it has achieved block synchronization or not.

Self-Synchronous Scrambling/Descrambling in 10GBASE-R and 10GBASE-KR Configurations

The scrambler/descrambler blocks in the transmitter/receiver PCS implements the self-synchronizing scrambler/descrambler polynomial 1 + x39 + x58, as described in clause 49 of the IEEE 802.3-2008 specification. The scrambler/descrambler blocks are self-synchronizing and do not require an initialization seed. Barring the two sync header bits in each 66-bit data block, the entire payload is scrambled or descrambled.

BER Monitor in 10GBASE-R and 10GBASE-KR Configurations

The BER monitor block in the receiver PCS implements the BER monitor state diagram shown in Figure 49-13 of the IEEE 802.3-2008 specification. The BER monitor provides a status signal to the MAC whenever the link BER threshold is violated.

The 10GBASE-R core and the 1G/10GbE and 10GBASE-KR PHY IP core (10GBASE-KR mode) provide a status flag to indicate a high BER whenever 16 synchronization header errors are received within a 125 µs window.

Clock Compensation in 10GBASE-R and 10GBASE-KR Configurations

The receiver FIFO in the receiver PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. The receiver FIFO does so by inserting Idles (/I/) and deleting Idles (/I/) or Ordered Sets (/O/), depending on the ppm difference.

- **Idle Insertion** — The receiver FIFO inserts eight /I/ codes following an /I/ or /O/ to compensate for clock rate disparity.
- **Idle (/I/) or Sequence Ordered Set (/O/) Deletion** — The receiver FIFO deletes either four /I/ codes or ordered sets (/O/) to compensate for the clock rate disparity. The receiver FIFO implements the following IEEE802.3-2008 deletion rules:
  - Deletes the lower four /I/ codes of the current word when the upper four bytes of the current word do not contain a Terminate /T/ control character.
  - Deletes one /O/ ordered set only when the receiver FIFO receives two consecutive /O/ ordered sets.

10GBASE-KR and 1000BASE-KX Link Training

The Link Training function defined in clause 72 of IEEE 802.3ap-2007 specification is implemented in the core fabric. The 1G/10GbE and 10GBASE-KR PHY IP Link Training logic includes the Training Frame Generator, Training Frame Synchronizer, PRBS11 generator, control channel codec, Local Device (LD) transceiver transmit PMA pre-emphasis coefficient status reporting, the Link Partner (LP) transmit PMA pre-emphasis coefficient update request, and the receiver link training status.

Arria V GZ channels employ three PMA transmit driver pre-emphasis taps: pre-tap, main tap, and first post-tap as required and defined by clause 72, Section 72.7.1.10 Transmitter output waveform for 10GBASE-KR PHY operation. The pre-emphasis coefficients is dynamically adjusted by the PHY IP during the Link Training process.

10GBASE-KR and 1000BASE-KX Auto-Negotiation

The Auto-Negotiation function defined in clause 73 of IEEE 802.3ap-2007 specification must be implemented in the core fabric. The 1G/10GbE and 10GBASE-KR PHY IP Auto-Negotiation logic includes the Differential Manchester Encoding (DME) page codec, AN page lock and synchronizer, and the Transmit, Receive, and Arbitration logic state machines.

10GBASE-KR Forward Error Correction

The FEC function defined in clause 74 of IEEE 802.3ap-2007 specification must be implemented in the core fabric. In Arria V GZ devices, the hard PCS does not support applications that require FEC functionality. To implement a 10GBASE-KR link with FEC support, the entire PCS functionality and the FEC logic must be implemented in the core fabric and the transceiver configured in Low Latency Configuration using the Native PHY IP.

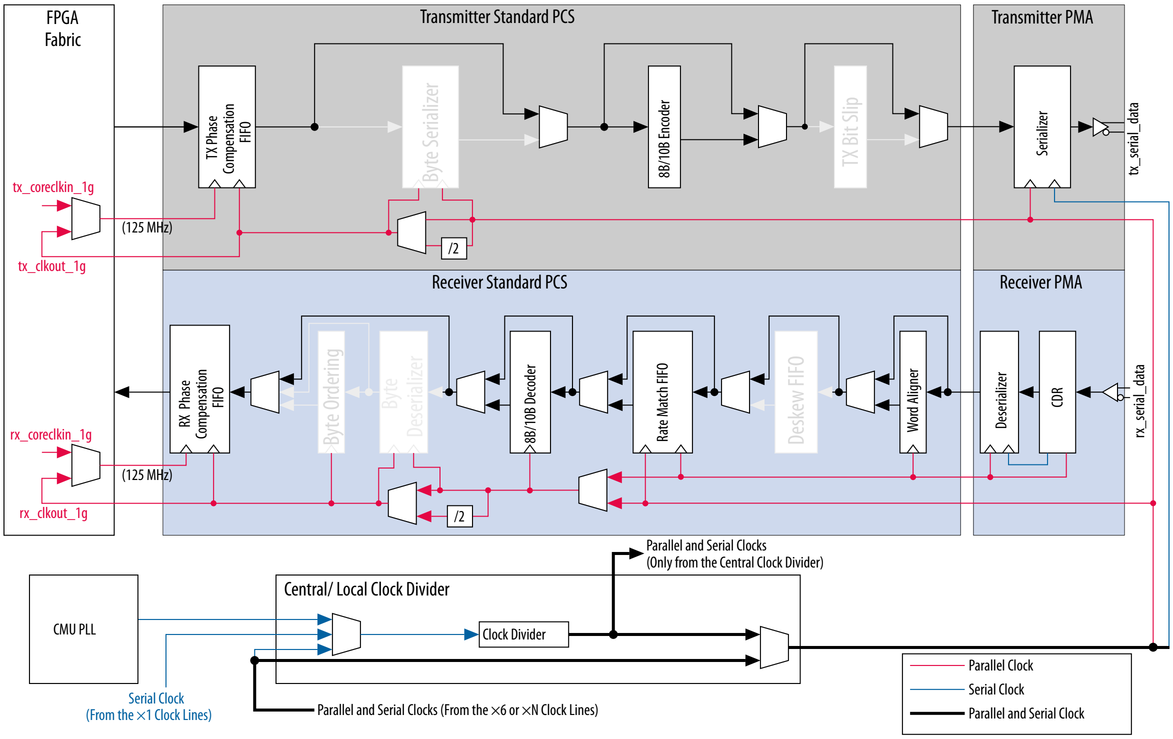
Related Information

Native PHY IP Configuration on page 6-86

1000BASE-X and 1000BASE-KX Transceiver Datapath

The following figure shows the transceiver datapath and clock frequencies in 1000BASE-X and 1000BASE-KX configurations.

Figure 6-8: 1000BASE-X and 1000BASE-KX Datapath Configurations



1000BASE-X and 1000BASE-KX Supported Features

The following features are supported by the transceivers in 1000BASE-X and 1000BASE-KX configurations.

8B/10B Encoder in 1000BASE-X and 1000BASE-KX Configurations

In 1000BASE-X and 1000BASE-KX modes, the 8B/10B encoder clocks in 8-bit data and 1-bit control identifiers from the transmitter phase compensation FIFO and generates 10-bit encoded data. The 10-bit encoded data is fed to the serializer.

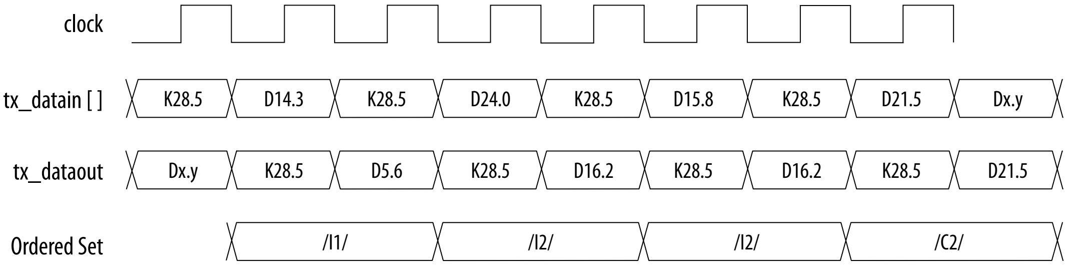
Idle Ordered-Set Generation in 1000BASE-X and 1000BASE-KX Configurations

The IEEE 802.3 specification requires the 1000BASE-X and 1000BASE-KX PHY to transmit idle ordered sets (/I/) continuously and repetitively whenever the GMII is idle. This ensures that the receiver maintains bit and word synchronization whenever there is no active data to be transmitted.

In 1000BASE-X and 1000BASE-KX functional modes, any /Dx.y/ following a /K28.5/ comma is replaced by the transmitter with either a /D5.6/ (/I1/ ordered set) or a /D16.2/ (/I2/ ordered set), depending on the current running disparity. The exception is when the data following the /K28.5/ is /D21.5/ (/C1/ ordered set) or /D2.2/ (/C2/) ordered set. If the running disparity before the /K28.5/ is positive, an /I1/ ordered set is generated. If the running disparity is negative, a /I2/ ordered set is generated. The disparity at the end of a /I1/ is the opposite of that at the beginning of the /I1/. The disparity at the end of a /I2/ is the same as the beginning running disparity (right before the idle code). This ensures a negative running disparity at the end of an idle ordered set. A /Kx.y/ following a /K28.5/ is not replaced.

Note: /D14.3/, /D24.0/, and /D15.8/ are replaced by /D5.6/ or /D16.2/ (for /I1/, /I2/ ordered sets). /D21.5/ (part of the /C1/ order set) is not replaced.

Figure 6-9: Example of Automatic Ordered Set Generation



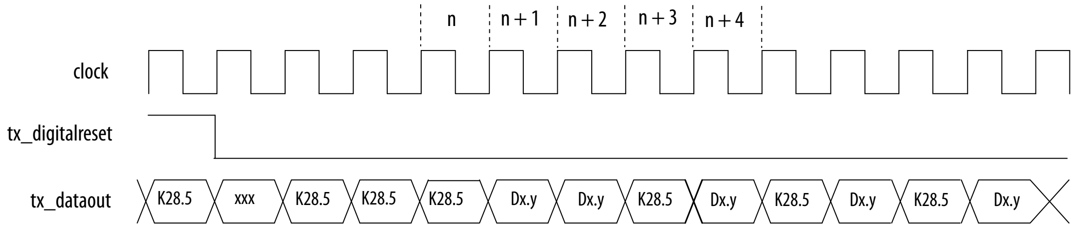
Reset Condition in 1000BASE-X and 1000BASE-KX Configurations

After deassertion of `tx_digitalreset`, the 1000BASE-X and 1000BASE-KX transmitters automatically transmit three /K28.5/ comma code groups before transmitting user data on the `tx_datain` port. This could affect the synchronization state machine behavior at the receiver.

Depending on when you start transmitting the synchronization sequence, there could be an even or odd number of /Dx.y/ code groups transmitted between the last of the three automatically sent /K28.5/ code groups and the first /K28.5/ code group of the synchronization sequence. If there is an even number of /Dx.y/ code groups received between these two /K28.5/ code groups, the first /K28.5/ code group of the synchronization sequence begins at an odd code group boundary (`rx_even` = FALSE). An IEEE802.3-compliant 1000BASE-X or 1000BASE-KX synchronization state machine treats this as an error condition and goes into the loss of sync state.

The following figure shows an example of even numbers of /Dx.y/ between the last automatically sent /K28.5/ and the first user-sent /K28.5/. The first user-sent /K28.5/ code group received at an odd code group boundary in cycle n + 3 takes the receiver synchronization state machine in the loss of sync state. The first synchronization ordered set /K28.5/Dx.y/ in cycles n + 3 and n + 4 is discounted and three additional ordered sets are required for successful synchronization.

Figure 6-10: Example of Reset Condition in 1000BASE-X and 1000BASE-KX Configurations



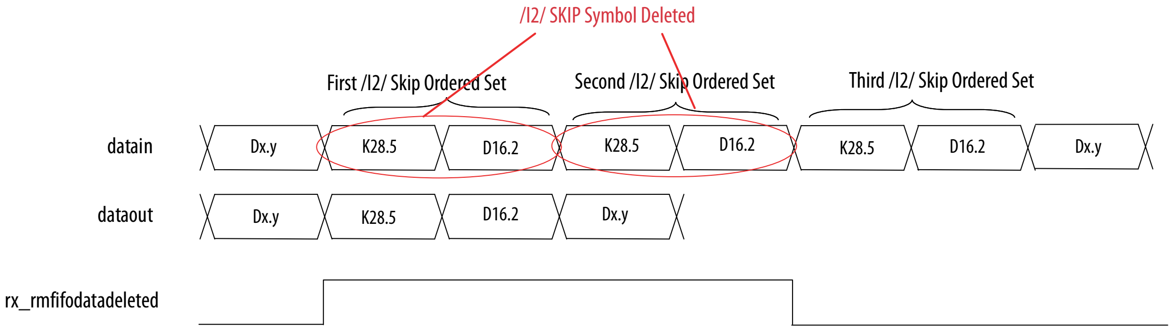
Rate Match FIFO in 1000BASE-X and 1000BASE-KX Configurations

In 1000BASE-X and 1000BASE-KX modes, the rate match FIFO is capable of compensating for up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The 1000BASE-X and 1000BASE-KX protocols require the transmitter to send idle ordered sets /I1/ (/K28.5/D5.6/) and /I2/ (/K28.5/D16.2/) during inter-packet gaps adhering to the rules listed in the IEEE 802.3 specification.

The rate match operation begins after the synchronization state machine in the word aligner indicates synchronization is acquired by driving the `rx_syncstatus` signal high. The rate matcher deletes or inserts both symbols (/K28.5/ and /D16.2/) of the /I2/ ordered sets even if it requires deleting only one symbol to prevent the rate match FIFO from overflowing or under-running. It can insert or delete as many /I2/ ordered sets as necessary to perform the rate match operation.

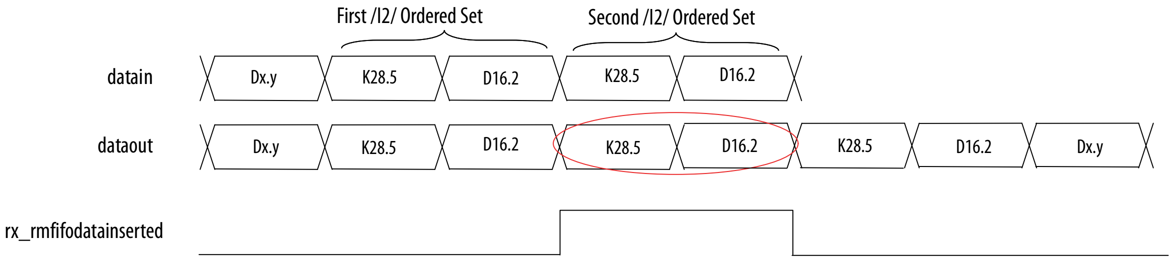
The following figure shows an example of rate match FIFO deletion where three symbols are required to be deleted. Because the rate match FIFO can only delete /I2/ ordered set, it deletes two /I2/ ordered sets (four symbols deleted).

Figure 6-11: Example of Rate Match Deletion in 1000BASE-X and 1000BASE-KX Configurations



The following figure shows an example of rate match FIFO insertion in the case where one symbol is required to be inserted. Because the rate match FIFO can only delete /I2/ ordered set, it inserts one /I2/ ordered set (two symbols inserted).

Figure 6-12: Example Rate Match Insertion in 1000BASE-X and 1000BASE-KX Configurations



Two register bits, `rx_rmfifodatadeleted` and `rx_rmfifodatainserted`, indicate rate match FIFO deletion and insertion events. Both the `rx_rmfifodatadeleted` and `rx_rmfifodatainserted` status flags are latched High during deleted and inserted /I2/ ordered sets.

Note: If you have the autonegotiation state machine in the FPGA, note that the rate match FIFO is capable of inserting or deleting the first two bytes (/K28.5//D2.2/) of /C2/ ordered sets during autonegotiation. However, the insertion or deletion of the first two bytes of /C2/ ordered sets can cause the autonegotiation link to fail. For more information, refer to the **Altera Knowledge Base Support Solution**.

Word Aligner in 1000BASE-X and 1000BASE-KX Configurations

The word aligner in 1000BASE-X and 1000BASE-KX functional modes is configured in automatic synchronization state machine mode. The Quartus II software automatically configures the synchronization state machine to indicate synchronization when the receiver receives three consecutive synchronization ordered sets. A synchronization ordered set is a /K28.5/ code group followed by an odd number of valid /Dx.y/ code groups. The fastest way for the receiver to achieve synchronization is to receive three continuous {/K28.5/, /Dx.y/} ordered sets.

Receiver synchronization is indicated on the `rx_syncstatus` port of each channel. A high on the `rx_syncstatus` port indicates that the lane is synchronized; a low on the `rx_syncstatus` port indicates that the lane has fallen out of synchronization. The receiver loses synchronization when it detects four invalid code groups separated by less than three valid code groups or when it is reset.

Synchronization State Machine Parameters in 1000BASE-X and 1000BASE-KX Configurations

Table 6-1: Synchronization State Machine Parameters in 1000BASE-X or 1000BASE-KX Mode

Synchronization State Machine Parameters	Settings
Number of valid {/K28.5/, /Dx,y/} ordered sets received to achieve synchronization	3
Number of errors received to lose synchronization	4
Number of continuous good code groups received to reduce the error count by 1	4

Transceiver Clocking in 10GBASE-R, 10GBASE-KR, 1000BASE-X, and 1000BASE-KX Configurations

The CMU PLL or the auxiliary transmit (ATX) PLLs in a transceiver bank generate the transmitter serial and the fractional PLL for the parallel clocks for the 10GBASE-R, 10GBASE-KR, 1000BASE-X, and 1000BASE-KX channels. The following table lists the configuration details.

Table 6-2: Input Reference Clock Frequency and Interface Speed Specifications for 10GBASE-R, 10GBASE-KR, and 1000BASE-KX Configurations

PHY IP Type	PHY Type	Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
10GBASE-R PHY IP	10GBASE-R	644.53125, 322.265625	64-bit data, 8-bit control	156.25
1G/10GbE and 10GBASE-KR PHY IP	10GBASE-R and 10GBASE-KR	644.53125, 322.265625	64-bit data, 8-bit control	156.25
1G/10GbE and 10GBASE-KR PHY IP	1000BASE-X and 1000BASE-KX	125, 62.5	8-bit data, `gmii_tx_en` and `gmii_tx_err` control	125

Interlaken

Interlaken is a scalable, chip-to-chip interconnect protocol that enables transmission speeds from 10 to more than 100 Gbps.

Arria V GZ devices support a transmission speed of up to 12.5 Gbps per lane in an Interlaken configuration. All the PCS blocks in the Interlaken configuration conform to the Interlaken Protocol Definition, Rev 1.2.

To implement an Interlaken link, instantiate the **Interlaken PHY IP** core in the IP Catalog, under **Interlaken** in the Interfaces menu.

Related Information

Refer to the Interlaken PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide

Transceiver Datapath Configuration

Figure 6-13: Interlaken Datapath Configuration

Blocks shown as “Disabled” are not used but incur latency. Blocks shown as “Bypassed” are not used and do not incur any latency. The maximum data rates and frequencies are for the fastest speed grade devices.

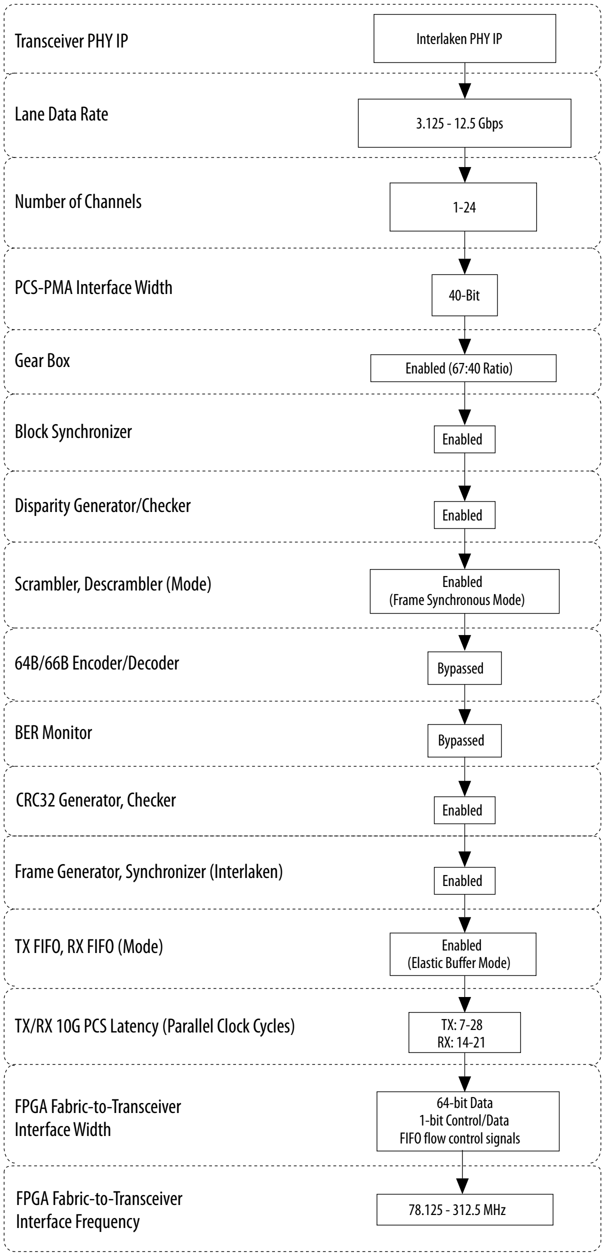
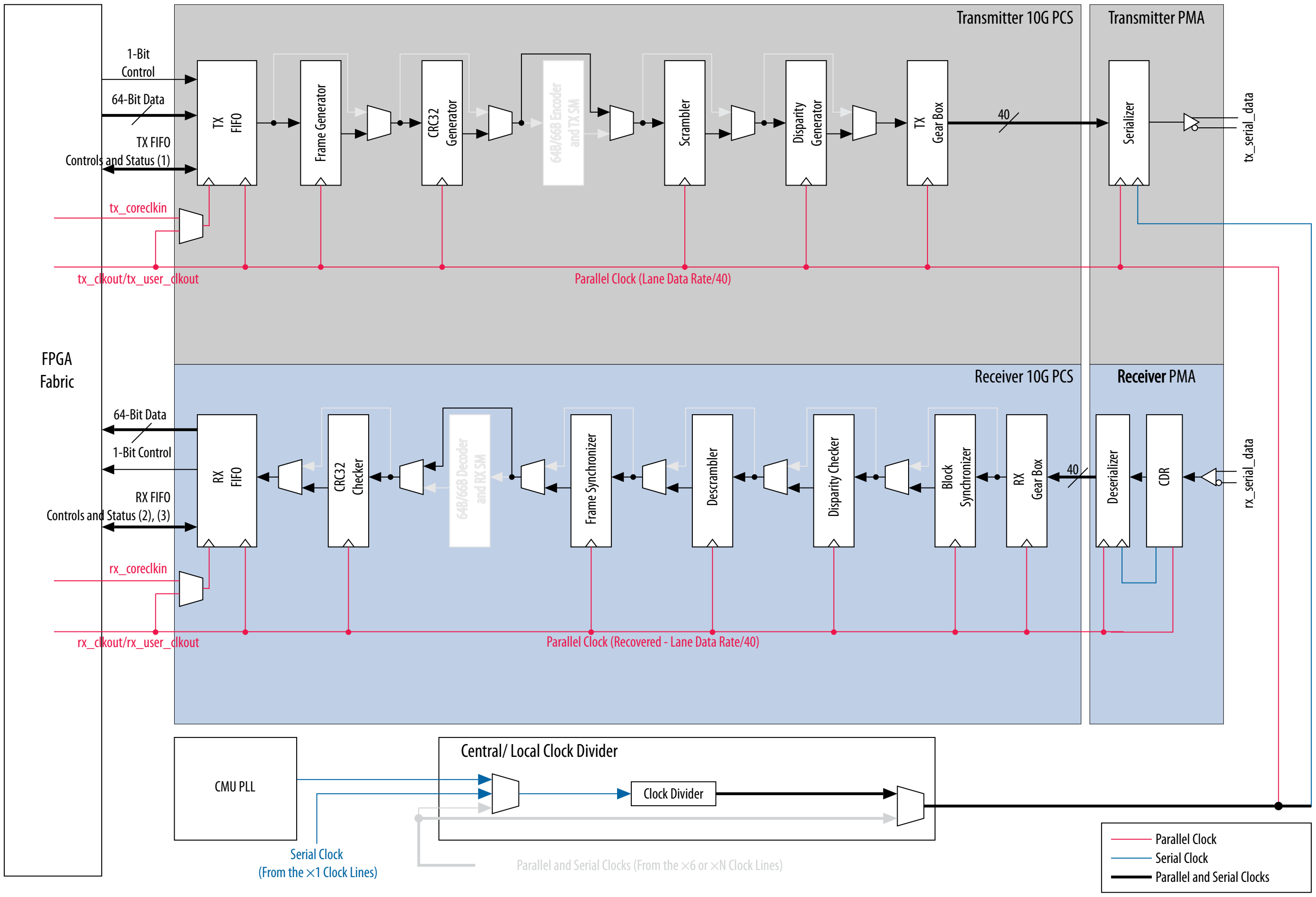


Figure 6-14: Transceiver Channel Datapath for Interlaken Configuration



Notes:
(1) TX FIFO Control and Status (transmit backpressure and datavalid, synchronization done)
(2) RX FIFO Control (receive FIFO read enable and datavalid)
(3) RX FIFO Status (receive FIFO overflow and partially empty)

Supported Features

The Interlaken protocol supports a number of framing layer functions. The functions are defined in the Interlaken Protocol Definition, Rev 1.2.

Table 6-3: Supported Features in Interlaken Configuration

Feature	Supported
Metaframe generation and payload insertion	Yes
Block synchronization (word alignment) and metaframe synchronization (frame synchronization)	Yes
64B/67B framing	Yes
±96 bits disparity maintenance	Yes

Feature	Supported
Frame synchronous scrambling and descrambling	Yes
Diagnostic word generation	Yes
Framing Layer Control Word Forwarding	Yes
CRC-32 generation and checking of lane data integrity	Yes
Multi-lane deskew alignment	No
Transmit and receive FIFO backpressure control and handshake	Yes

Block Synchronizer

The block synchronizer in the receiver PCS achieves and maintains a 64B/67B word boundary lock. This block searches for valid synchronization header bits within the data stream and achieves lock after 64 consecutive legal synchronization patterns are found. After a 64B/67B word boundary lock is achieved, the block synchronizer continuously monitors and flags for invalid synchronization header bits. If 16 or more invalid synchronization header bits are found within 64 consecutive word boundaries, the block synchronizer deasserts the lock state and searches again for valid synchronization header bits.

The block synchronizer implements the flow diagram shown in Figure 13 of Interlaken Protocol Definition v1.2 and provides the word lock status to the FPGA fabric.

64B/67B Frame Generator

The transmit frame generator implements 64B/67B encoding, as explained in Interlaken Protocol Definition v1.2. The Interlaken metaframe generator synchronously generates the framing layer control words, frame synchronizer, scrambler state, skip words, and diagnostic word, and maps the transmitter data into the payload of the metaframes. The metaframe length is programmable from 5 to a maximum value of 8191, 8-byte words.

Note: Ensure that the metaframe length is programmed to the same value for both the transmitter and receiver.

Frame Synchronizer

The receive frame synchronizer delineates the metaframe boundaries and searches for each of the framing layer control words: Synchronization, Scrambler State, Skip, and Diagnostic. When four consecutive synchronization words have been identified, the frame synchronizer achieves the frame locked state. Subsequent metaframes are then checked for valid synchronization and scrambler state words. If four consecutive invalid synchronization words or three consecutive mismatched scrambler state words are received, the frame synchronizer loses frame lock. In addition, the frame synchronizer provides a receiver metaframe lock status to the FPGA fabric.

Running Disparity

The disparity generator inverts the sense of bits in each transmitted word to maintain a running disparity of ± 96 bit boundary. It supplies a framing bit in bit position 66 as explained in Table 4 of Interlaken Protocol Definition Revision 1.2. The framing bit enables the disparity checker to identify whether bits[63:0] for that word are inverted.

Frame Synchronous Scrambling/Descrambling

The scrambler/descrambler block in the transmitter/receiver PCS implements the scrambler/descrambler polynomial x58 + x39 + 1 per Interlaken Protocol Definition Revision 1.2. Synchronization and Scrambler State Words, as well as the 64B/67B framing bits are not scrambled/descrambled. The Interlaken PHY IP core automatically programs random linear feedback shift register (LFSR) initialization seed values per lane.

The receiver PCS synchronizes the scrambler with the metaframe as described in the state flow shown in Figure 1 of Interlaken Protocol Definition Revision 1.2.

The frame synchronizer features a whole set of error and performance monitoring ports to the FPGA fabric interface and register status bits when using the Avalon® Memory-Mapped Management Interface. A receiver ready port, frame lock status, and cyclic redundancy check (CRC)-32 error detection port is available to the FPGA fabric. The Avalon Memory-Mapped Management Interface provides additional functionality with word boundary lock, frame lock status, synchronization word error detection, scrambler mismatch error, and CRC-32 error detection status register bits.

Skip Word Insertion

The frame generator generates the mandatory fixed location skip words with every metaframe following the scrambler state word and generates additional skip words based on the transmitter FIFO capacity state.

Skip Word Deletion

The frame synchronizer does not delete skip words. Instead, the frame synchronizer forwards the skip words it receives to the MAC layer so the MAC can maintain and perform deskew alignment.

Diagnostic Word Generation and Checking of Lane Data Integrity (CRC-32)

The CRC-32 generator calculates the CRC for each metaframe and appends it to the diagnostic word of the metaframe. An optional CRC-32 error flag is also provided to the FPGA fabric.

Framing Layer Control Word Forwarding

The four metaframe framing layer control words-Synchronization, Scrambler State, Skip, and Diagnostic Words-are not deleted but forwarded to the MAC layer. This action enables the MAC layer to employ multi-lane deskew alignment within the FPGA fabric.

Note: The Scrambler State word seed is zeroed (Bit[57:0]) before it is forwarded to the MAC layer.

Multi-Lane Deskew Alignment

The Interlaken PHY IP does not support multi-lane deskew alignment. You must implement the multi-lane deskew alignment state machine in the core fabric or the Interlaken Intel® FPGA IP core function within the FPGA fabric.

Transmit and Receive FIFO Control and Status

The Interlaken PCS configures the transmit and receive FIFOs in elastic buffer mode. In this mode of operation, a lane synchronization, backpressure and FIFO control, and status port signals are provided to the MAC layer for handshaking.

Transceiver Multi-Lane Bonding and Transmit Skew

A soft-bonding IP is used for Interlaken bonding in the transceivers. The transceiver clocking in each lane is configured as non-bonded. For multi-lane designs, a dedicated PLL reference clock pin that is equidistant from the transmit PLLs in each bank must be selected. You must tightly match lane board traces to minimize lane-to-lane skew.

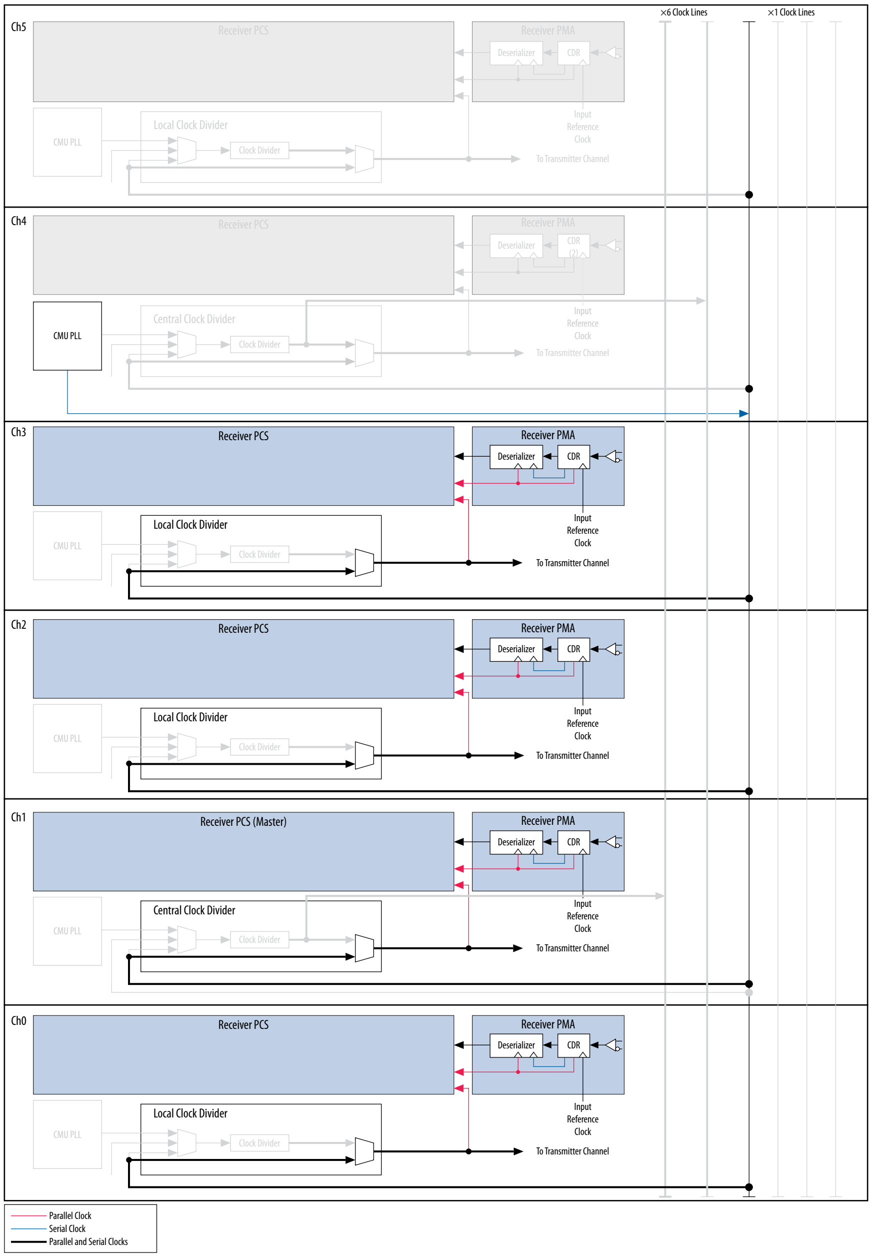
Related Information

- **For more information about Interlaken PHY IP control and status signals associated with each feature, refer to the Interlaken PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide**
- **Interlaken MegaCore Function User Guide**

Transceiver Clocking

Describes the transceiver clocking for the Interlaken protocol.

Figure 6-15: Clocking Resources Available in a Four-Lane Interlaken Configuration



A CMU PLL may provide a clock for up to five Interlaken lanes within a transceiver bank. If an ATX PLL is used, the PLL can clock up to six Interlaken lanes in a transceiver bank.

Note: To enable the ATX PLL, you must select **ATX PLL** for the **PLL type** parameter in the Interlaken PHY IP.

PCI Express (PCIe)—Gen1, Gen2, and Gen3

The PCIe specification (version 3.0) provides implementation details for a PCIe-compliant physical layer device at Gen1 (2.5 Gbps), Gen2 (5 Gbps), and Gen3 (8 Gbps) signaling rates.

The devices have built-in PCIe hard IP blocks to implement the PHY MAC layer, data link layer, and transaction layer of the PCIe protocol stack. Up to four PCIe hard IP block reside within an Arria V GZ device. If you enable the PCIe hard IP block, the transceiver interfaces with the hard IP block. Otherwise, the transceiver interfaces directly through the PIPE interface. You must then implement a Soft-IP MAC layer, data link layer, and transaction layer to the PIPE interface from the core fabric.

You can configure the transceivers in a PCIe functional configuration using one of the following methods:

- Arria V GZ Hard IP for PCI Express
- PHY IP core for PCI Express (PIPE)

The following table shows the two methods supported by transceivers in a PCIe functional configuration.

Table 6-4: Support for Transceivers

Support	Arria V GZ Hard IP for PCI Express	PHY IP Core for PCI Express (PIPE)
Gen1, Gen2, and Gen3 data rates	Yes	Yes
MAC, data link, and transaction layer	Yes	—
Transceiver interface	Hard IP through PIPE 3.0-like	PIPE 2.0 for Gen1 and Gen2 PIPE 3.0-like for Gen3 with Gen1/Gen2 support

To implement the PHY IP Core for PCI Express (PIPE) configuration, instantiate the **PHY IP Core for PCI Express (PIPE)** in the IP Catalog, under **PCI Express** in the Interfaces menu.

Arria V GZ transceivers support x1, x2, x4, and x8 lane configurations. In a PCIe x1 configuration, the PCS and PMA blocks of each channel are clocked and reset independently. PCIe x2, x4, and x8 configurations support channel bonding for two-lane, four-lane, and eight-lane PCIe links. In these bonded channel configurations, the PCS and PMA blocks of all bonded channels share common clock and reset signals.

Related Information

- **Arria V Hard IP for PCI Express User Guide**
- **Refer to the PHY IP Core for PCI Express (PIPE) chapter in the Altera Transceiver PHY IP Core User Guide**

Transceiver Datapath Configuration

The transceiver datapaths for PCI Express are different depending on whether or not Gen3 is enabled.

Figure 6-16: PCIe Gen1 and Gen2 PIPE Datapath Configuration

This transceiver datapath configuration is for a configuration without Gen3 enabled.

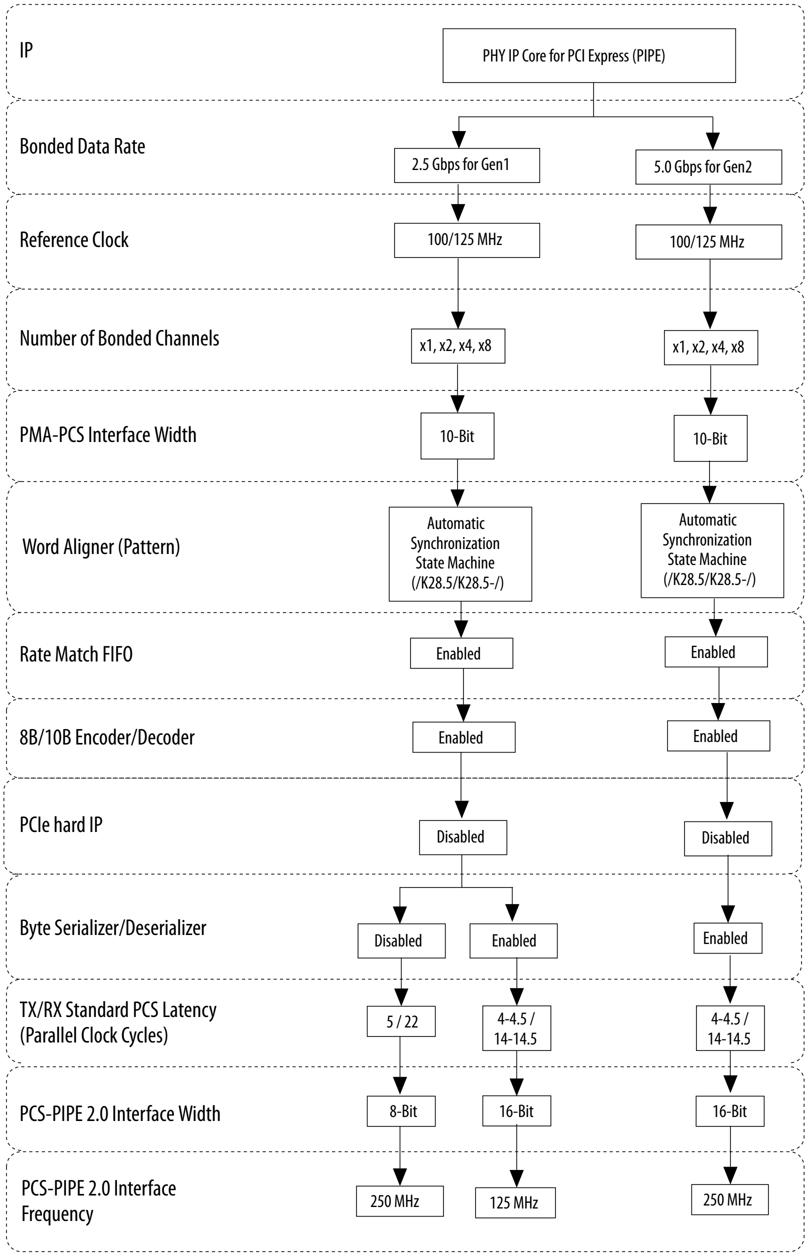
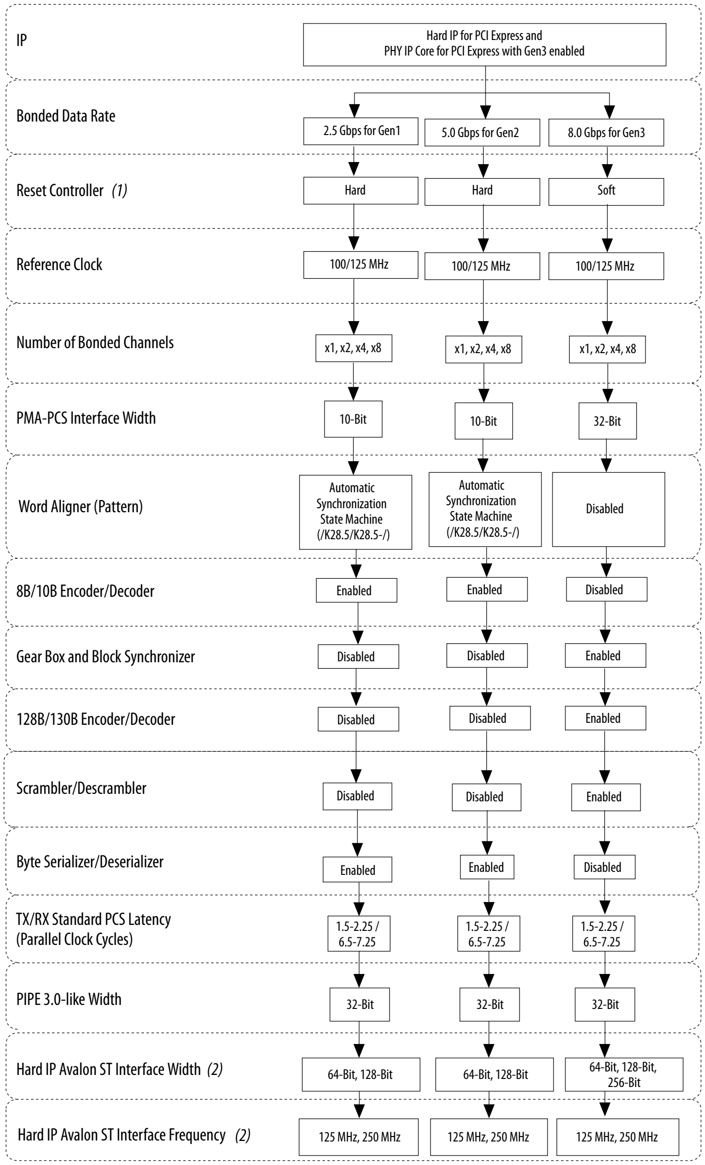


Figure 6-17: PCIe Gen1, Gen2, and Gen3 Hard IP and PHY IP Core for PCI Express Datapath Configuration

This transceiver datapath configuration is for a configuration with Gen3 enabled.



Notes:

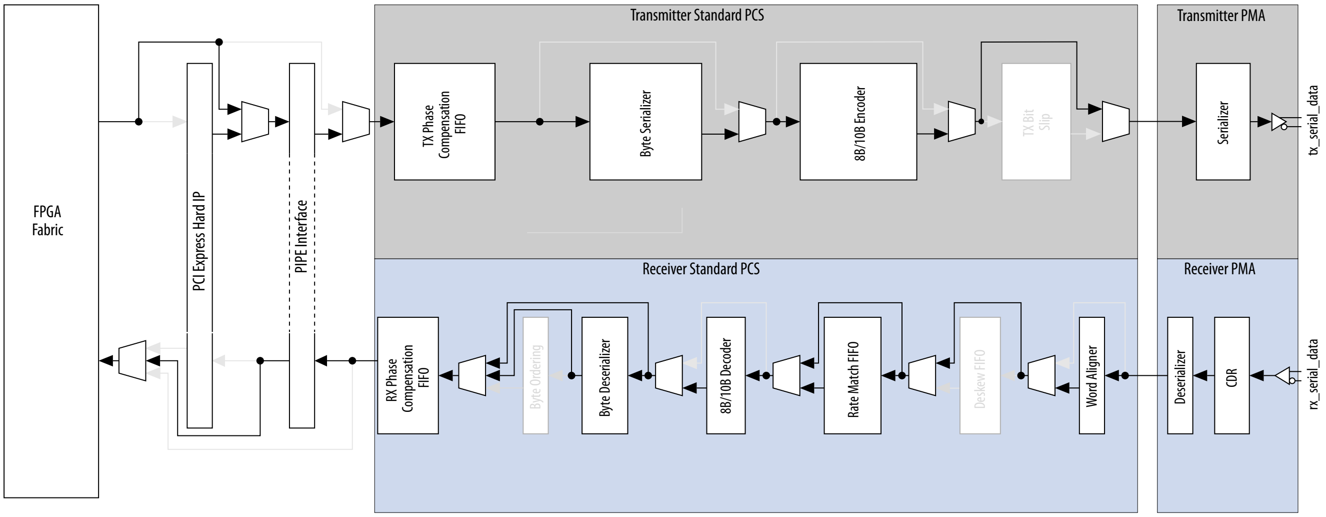
(1) The PHY IP Core for PCI Express (PIPE configuration) employs the Embedded Reset Controller IP. It does not use the Hard or Soft Reset Controller employed in the Hard IP for PCI Express (HIP configuration).

(2) Does not apply to PHY IP Core for PCI Express configuration. Applies only to Hard IP for PCI Express configuration.

Transceiver Channel Datapath

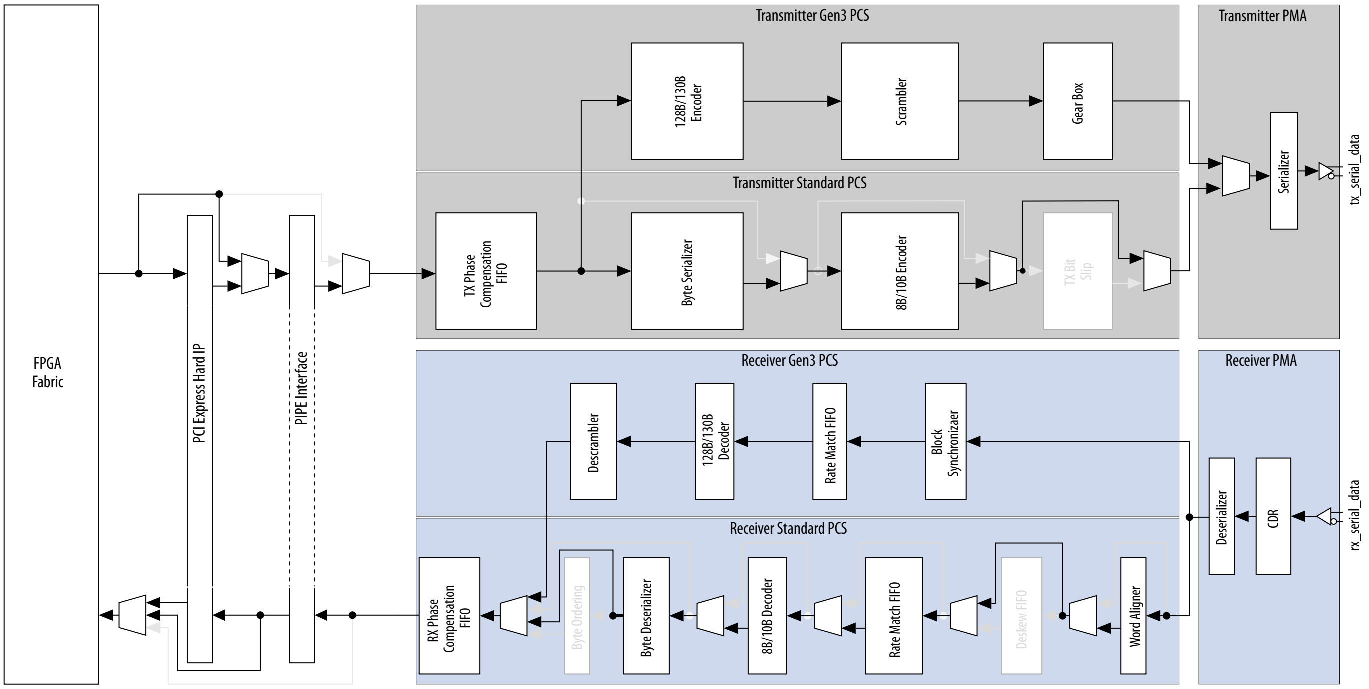
The following figure shows the Arria V GZ transmitter and receiver channel datapath for PCIe Gen1/Gen2 configurations when using PIPE configuration with Gen3 disabled. In this configuration, the transceiver connects to a PIPE 2.0 compliant interface.

Figure 6-18: Transceiver Channel Datapath for PCIe Gen1/Gen2 in PIPE Configuration with Gen3 Disabled



The following figure shows the Arria V GZ transmitter and receiver channel datapath for PCIe Gen1/Gen2/Gen3 configurations with a 32-bit PIPE 3.0-like interface and PCI Express Base Specification Version 3.0 is enabled.

Figure 6-19: Transceiver Channel Datapath for PCIe Gen1/Gen2/Gen3 Configurations



Related Information

Transceiver Architecture in Arria V Devices

Supported Features for PCIe Configurations

The features supported for a PCIe configuration are different for the 2.5 Gbps, 5 Gbps, and 8 Gbps data rate configurations.

Table 6-5: Supported Features for PCIe Configurations

Feature	Gen1 (2.5 Gbps)	Gen2 (5 Gbps)	Gen3 (8 Gbps)
x1, x2, x4, x8 link configurations	Yes	Yes	Yes
PCIe-compliant synchronization state machine	Yes	Yes	Yes
±300 ppm (total 600 ppm) clock rate compensation	Yes	Yes	Yes
8-bit FPGA fabric-transceiver interface (PIPE 2.0)	Yes	—	—
16-bit FPGA fabric-transceiver interface (PIPE 2.0)	Yes	Yes	—
32-bit FPGA fabric-transceiver interface (PIPE 3.0-like)	—	—	Yes
64-bit Hard IP Avalon-ST interface width (Hard IP only)	Yes	Yes	Yes
128-bit Hard IP Avalon-ST interface width (Hard IP only)	Yes	Yes	Yes
256-bit Hard IP Avalon-ST interface width (Hard IP only)	—	Yes	Yes
Transmitter driver electrical idle	Yes	Yes	Yes
Receiver Detection	Yes	Yes	Yes
8B/10B encoder/decoder disparity control	Yes	Yes	—
128B/130B encoder/decoder	—	—	Yes
Power state management	Yes	Yes	Yes
Receiver PIPE status encoding (pipe_rxstatus[2:0])	Yes	Yes	Yes
Dynamic switching between 2.5 Gbps and 5 Gbps signaling rate	—	Yes	—
Dynamic switching between 2.5 Gbps, 5 Gbps, and 8 Gbps signaling rate	—	—	Yes
Dynamic transmitter margining for differential output voltage control	—	Yes	Yes
Dynamic transmitter buffer de-emphasis of -3.5 dB and -6 dB	—	Yes	Yes
Dynamic Gen3 transceiver pre-emphasis, de-emphasis, and equalization	—	—	Yes

PIPE 2.0 Interface

In a PCIe PIPE configuration, each channel has a PIPE interface block that transfers data, control, and status signals between the PHY-MAC layer and the transceiver channel PCS and PMA blocks. The PIPE configuration complies with the PIPE 2.0 specification. If you use a PIPE configuration, you must implement the PHY-MAC layer using soft IP in the FPGA fabric.

Besides transferring data, control, and status signals between the PHY-MAC layer and the transceiver, the PIPE interface block implements the following functions required in a PCIe-compliant physical layer device:

- Forcing the transmitter driver into the electrical idle state
- Initiating the receiver detect sequence
- Controlling the 8B/10B encoder/decoder
- Controlling the 128B/130B encoder/decoder
- Managing the PCIe power states
- Indicating the completion of various PHY functions
- Encoding the receiver status and error conditions on the `pipe_rxstatus[2:0]` signal, conforming to the PCIe PIPE 3.0 specification

Transceiver datapath clocking varies between non-bonded (x1) and bonded (x2, x4, and x8) configurations.

Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates

In a PIPE configuration, the PIPE Parameter Editor provides an input signal (`pipe_rate`) that is functionally equivalent to the RATE signal specified in the PCIe specification. A low-to-high transition on this input signal (`pipe_rate`) initiates a data rate switch from Gen1 to Gen2. A high-to-low transition on the input signal initiates a data rate switch from Gen2 to Gen1. The signaling rate switch between Gen1 and Gen2 is achieved by changing the transceiver datapath clock frequency between 250 MHz and 500 MHz, while maintaining a constant, 16-bit width transceiver interface.

Transmitter Electrical Idle Generation

The PIPE interface block in Arria V GZ devices puts the transmitter buffer in the channel in an electrical idle state when the electrical idle input signal is asserted. During electrical idle, the transmitter buffer differential and common configuration output voltage levels are compliant to the PCIe Base Specification 2.0 for both PCIe Gen1 and Gen2 data rates.

The PCIe specification requires the transmitter driver to be in electrical idle in certain power states. For more information about input signal levels required in different power states, refer to “Power State Management”.

Power State Management

The PCIe specification defines four power states—P0, P0s, P1, and P2—that the physical layer device must support to minimize power consumption:

- P0 is the normal operating state during which packet data is transferred on the PCIe link.
- P0s, P1, and P2 are low-power states into which the physical layer must transition as directed by the PHY-MAC layer to minimize power consumption.

The PIPE interface in Arria V GZ transceivers provides an input port for each transceiver channel configured in a PIPE configuration.

Note: When transitioning from the P0 power state to lower power states (P0s, P1, and P2), the PCIe specification requires the physical layer device to implement power saving measures. Arria V GZ transceivers do not implement these power saving measures except for putting the transmitter buffer in electrical idle in the lower power states.

8B/10B Encoder Usage for Compliance Pattern Transmission Support

The PCIe transmitter transmits a compliance pattern when the Link Training and Status State Machine (LTSSM) enters the Polling.Compliance substate. The Polling.Compliance substate is used to assess if the transmitter is electrically compliant with the PCIe voltage and timing specifications.

Receiver Electrical Idle Inference

The PCIe protocol allows inferring the electrical idle condition at the receiver instead of detecting the electrical idle condition with analog circuitry.

In all PIPE configurations, (x1, x2, x4, and x8), each receiver channel PCS has an optional Electrical Idle Inference module that implements the electrical idle inference conditions specified in the PCIe Base Specification 2.0.

Receiver Status

The PCIe specification requires the PHY to encode the receiver status on a 3-bit status signal (`pipe_rxstatus[2:0]`). This status signal is used by the PHY-MAC layer for its operation. The PIPE interface block receives status signals from the transceiver channel PCS and PMA blocks, and encodes the status on the `pipe_rxstatus[2:0]` signal to the FPGA fabric. The encoding of the status signals on the `pipe_rxstatus[2:0]` signal conforms to the PCIe specification.

Receiver Detection

The PIPE interface block in Arria V GZ transceivers provides an input signal (`pipe_txdetectrx_loopback`) for the receiver detect operation required by the PCIe protocol during the Detect state of the LTSSM. When the `pipe_txdetectrx_loopback` signal is asserted in the P1 power state, the PCIe interface block sends a command signal to the transmitter driver in that channel to initiate a receiver detect sequence. In the P1 power state, the transmitter buffer must always be in the electrical idle state. After receiving this command signal, the receiver detect circuitry creates a step voltage at the output of the transmitter buffer. If an active receiver (that complies with the PCIe input impedance requirements) is present at the far end, the time constant of the step voltage on the trace is higher when compared with the time constant of the step voltage when the receiver is not present. The receiver detect circuitry monitors the time constant of the step signal seen on the trace to determine if a receiver was detected. The receiver detect circuitry requires a 125-MHz clock for operation that you must drive on the `fixedclk` port.

Note: For the receiver detect circuitry to function reliably, the transceiver on-chip termination must be used and the AC-coupling capacitor on the serial link and the receiver termination values used in your system must be compliant with the PCIe Base Specification 2.0.

The PIPE core provides a 1-bit PHY status (`pipe_phystatus`) and a 3-bit receiver status signal (`pipe_rxstatus[2:0]`) to indicate whether a receiver was detected or not, as per the PIPE 2.0 specifications.

Gen1 and Gen2 Rate Match FIFO

In compliance with the PCIe protocol, Arria V GZ receiver channels have a rate match FIFO to compensate for small clock frequency differences up to ±300 ppm between the upstream transmitter and the local receiver clocks.

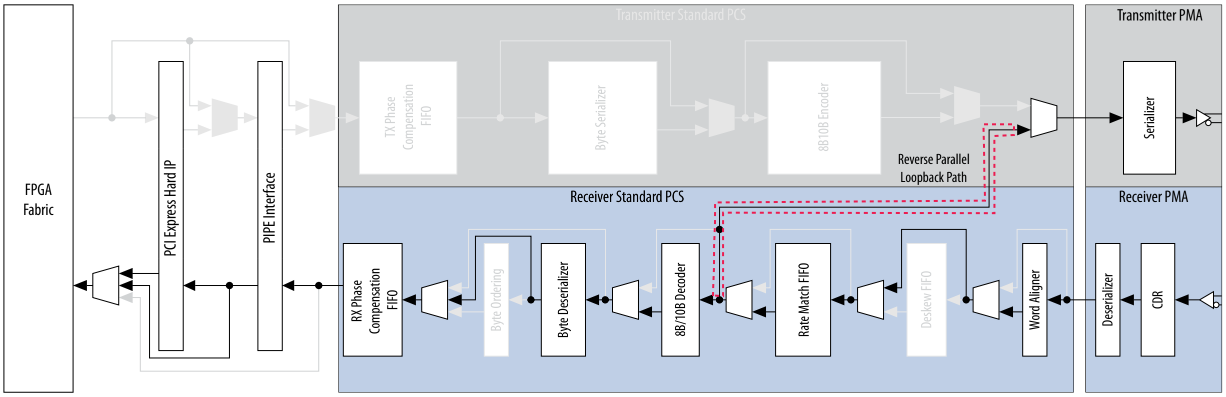
PCIe Reverse Parallel Loopback

PCIe reverse parallel loopback is only available in a PCIe functional configuration for Gen1, Gen2, and Gen3 data rates. The received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. The data is then looped back to the transmitter serializer and transmitted out through the transmitter buffer. The received data is also available to the FPGA fabric through the port. This loopback mode is compliant with the PCIe specification 2.0. Arria V GZ devices provide an input signal to enable this loopback mode.

Note: This is the only loopback option supported in PIPE configurations.

Figure 6-20: PCIe Reverse Parallel Loopback Mode Datapath

The grayed-out blocks are Inactive.



Related Information

- Refer to the PHY IP Core for PCI Express (PIPE) chapter in the Altera Transceiver PHY IP Core User Guide
- Refer to the "PCS Architecture" section in the Transceiver Architecture in Arria V Devices chapter
- For the power state requirements when switching between Gen1 and Gen2 data rates, refer to the PCIe Base Specification 2.0.

Supported Features for PCIe Gen3

The PCIe Gen3 hard PCS supports the Gen3 base specification. PCIe Gen3 operations can be configured using the Arria V GZ Hard IP for PCI Express IP or PHY IP Core for PCI Express.

In Arria V GZ Hard IP for PCI Express, selecting **PCIe Base Specification Version 3.0** or **PCI Express Base Specification Version 2.1** enables a 32-bit wide PIPE 3.0-like interface for Gen1, Gen2, and Gen3 operations.

In PHY IP Core for PCI Express, selecting Gen3 enables the 32-bit wide PIPE 3.0-like interface and selecting Gen1 or Gen2 enables the 16-bit/8-bit wide PIPE 2.0 interface for Gen1 and Gen2 operation.

Block Synchronization (Word Aligner)

The block synchronizer aligns the recovered serial data coming from the CDR to 130-bit word boundaries. The block synchronizer delineates the word boundaries by searching and identifying the Electrical IDLE Exit Sequence Ordered Set (EIEOS) or the Last FTS OS and SKP ordered set to correctly identify the word boundary from the incoming serial data stream. The block synchronizer continues to realign to a new block boundary following the receipt of an SKP ordered set because of varying word lengths.

Gen3 Rate Match FIFO

To accommodate PCIe protocol requirements and to compensate for clock frequency differences of up to ±300 ppm between source and termination equipment, receiver channels have a rate match FIFO. The rate match FIFO adds or deletes four SKP characters (32 bits) to keep the FIFO from becoming empty or full. It monitors the block synchronizer for a `skip_found` signal. If the rate match FIFO is almost full, the FIFO deletes four SKP characters. If the rate match FIFO is nearly empty, the FIFO inserts an SKP character at the start of the next available SKP ordered set.

128B/130B Encoder/Decoder

Unlike PCIe Gen1 and Gen2, the PCIe Gen3 encoder/decoder does not use 8B/10B encoding. The PCIe Gen3 encoder/decoder uses a 2-bit sync header and a 128-bit data word. The PCS encoder appends the two sync header bits to every 128 bits of data and enables scrambling for the data packets except for ordered set packets and the first symbol of a TS1/TS2 ordered set. The encoder/decoder continuously enables or disables scrambling, based on whether the payload being processed is an ordered set or a data packet. If an Electrical IDLE Exit Ordered Set or a Fast Training Sequence Ordered Set is received, the scrambler is reset to the initial seed value. The encoder/decoder also monitors the data stream for ordered set and sync header bit violations.

Gen3 Gear Box

The PCIe 3.0 base specification requires a block size of 130 bits with the exception of SKP ordered sets, which can be 66, 98, 130, 162, or 194 bits in length. The 130-bit block of data generated by the 128B/130B encoder and variable length SKP characters must be reordered in 32-bit parallel data segments that the PMA serializer can accept. The transceivers employ a gear box to accommodate this fractional bit difference between the 130-bit data word and a fixed 32-bit serialization PMA factor for Gen3.

Scrambler/Descrambler

Scrambling and descrambling are used during PCIe Gen3 operation to guarantee adequate transitions for the receiver in order to correctly regenerate the recovered clock. The 2-bit sync header bit, ordered set, and the first symbol of the TS1/TS2 ordered set are never scrambled.

PIPE 3.0-Like Gen3 Interface

PCIe Gen3 is a new feature added to the transceivers. The PCS supports PCI Express 3.0 base specification. The PIPE interface has been expanded to a 32-bit wide PIPE 3.0-like interface. The PIPE interface controls PHY functions such as transmission of electrical idle, receiver detection, and speed negotiation and control. In summary, the Gen3 PIPE 3.0-like interface block performs the following:

- Dynamic clock selection between Gen1, Gen2, and Gen3 speeds
- Gen3 auto speed negotiation (ASN)
- Controlling the 128B/130B encoder/decoder
- Gen3 Electrical Idle Entry and Exit detections/CDR Control Block
- Dynamic Gen3 and Gen2/Gen1 PCS data rate Auto Speed Negotiation
- Dynamic transceiver PMA data rate and PLL switching

Auto-Speed Negotiation Block

PCIe Gen3 mode enables ASN (auto-speed negotiation) between Gen1 (2.5 Gbps), Gen2 (5.0 Gbps), and Gen3 (8.0 Gbps) signaling data rates. The signaling rate switch is accomplished through frequency scaling and configuration of the PMA and PCS blocks using a fixed 32-bit wide PIPE 3.0-like Interface.

The PMA switches clocks between Gen1, Gen2, and Gen3 data rates in a glitch-free manner. For a non-bonded x1 channel, an ASN module facilitates speed negotiation in that channel. For bonded x2, x4, and x8 channels, the ASN module selects the master channel to control the rate switch. The master channel distributes the speed change request to the other PMA and PCS channels.

Table 6-6: PIPE Gen3 32-Bit PCS Clock Rates

PCIe Gen3 Capability Mode Enabled	Gen1	Gen2	Gen3
Lane data rate	2.5G	5G	8G
PCS clock frequency	250 MHz	500 MHz	250 MHz
FPGA Core IP clock frequency	62.5 MHz	125 MHz	250 MHz
PIPE interface width	32-bit	32-bit	32-bit
Rate[1:0]	00	01	10

The PCIe Gen3 speed negotiation process is initiated by writing a 1 to bit 5 of the Link Control register of the root port, causing a PIPE rate signal change from the hard IP. The ASN then places the PCS in reset, dynamically shuts down the clock paths to disengage the current active state PCS (either Standard PCS or Gen3 PCS). If a switch to or from Gen3 is requested, the ASN automatically selects the correct PCS clock paths and datapath selection in the multiplexers. The ASN block then sends a request to the PMA block to switch the data rate change and waits for a rate change done signal for confirmation. When the PMA completes the rate change and sends confirmation to the ASN block, ASN enables the clock paths to engage the new PCS block and releases the PCS reset. Successful completion of this process is indicated by assertion of the `pipe_phystatus` signal by the ASN block to the hard IP block.

Note: In PHY IP Core for PCI Express configuration, the Core IP must set the values to `pipe_rate[1:0]` to initiate the transceiver datarate switch sequence.

Note: When you switch speeds to either Gen2 or Gen3, hold the LTSSM steady for 700 µs in Recovery.RCVRLOCK. The `rx_is_lockedtodata` signal from the CDR must be stable during this time. The PHY MAC interface should not look at `rxvalid` during this time because its contents may be invalid.

Transmitter Electrical IDLE Generation

The PIPE 3.0-like interface under the control of the hard IP block in Hard IP for PCIe or the user Core IP in PHY IP Core for PCIe may place the transmitter in electrical idle during low power states and the ASN process. Before the transmitter enters electrical idle, the HIP sends an electrical idle order set (EIOS) to the PHY. For Gen1 and Gen2, the order set format is COM, IDL, IDL, IDL. For Gen3, the order set format consists of 16 symbols with value 0x66.

During electrical idle, the transmitter differential and common mode voltage levels are compliant to the PCIe Base Specification 3.0.

Receiver Electrical IDLE Inference

If there is no activity on the link for a period of time or during the ASN process, the Inferring Electrical Idle condition is detected by the receiver PHY. These conditions are specified according to Table 4-11 of the PCI Express Base Specification, Rev 3.0.

Gen3 Power State Management

The PCIe base specification defines low power states for PHY layer devices to minimize power consumption. The Gen3 PCS does not implement these power saving measures, except when placing the transmitter driver in electrical idle state in the low power states. In P2 low power state, the transceivers do not disable the PIPE block clock.

CDR Control Block

The CDR control block controls the PMA CDR to obtain bit and symbol alignment and deskew within the allocated time, and generates status signals for other PCS blocks. The PCIe base specification requires that the receiver L0s power state exit time be a maximum of 4 ms for Gen1, 2 ms for Gen2, and 4 ms for Gen3 signaling rates. The transceivers have an improved CDR control block to accommodate fast lock times when the CDR must relock to the new multiplier/divider settings when entering or exiting Gen3 speeds.

Transceiver Clocking and Channel Placement Guidelines

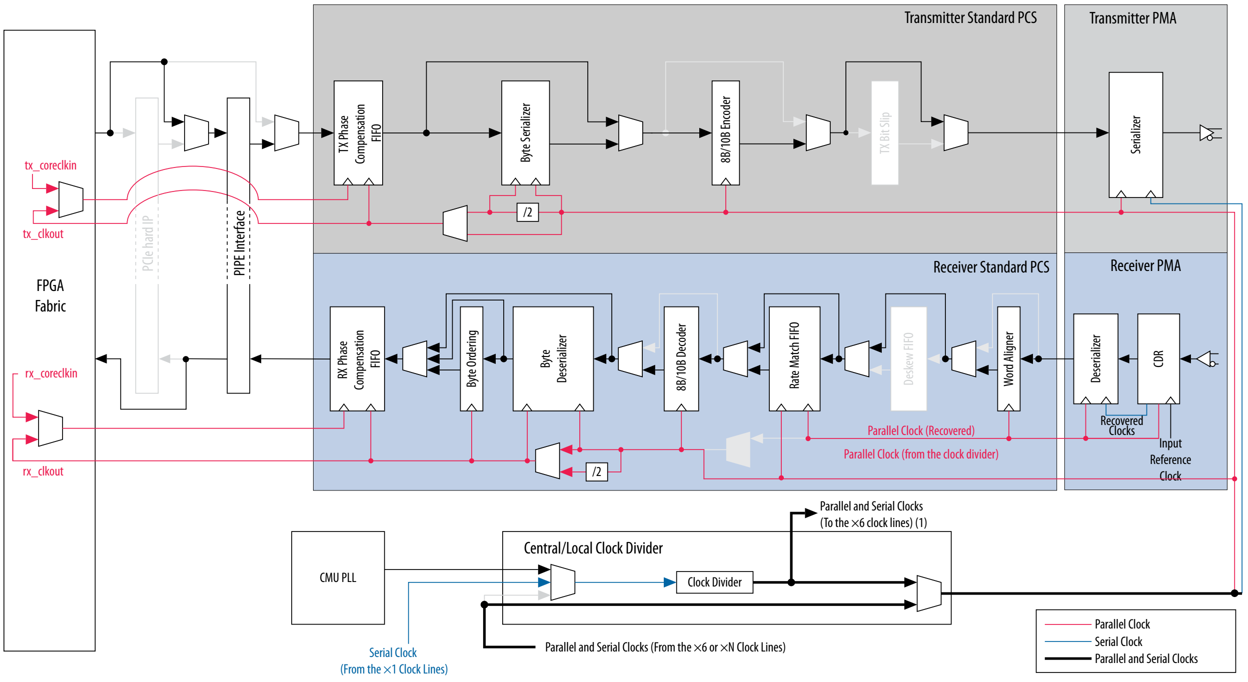
This section describes the transceiver clocking for Gen1 and Gen2 Hard IP and PIPE configurations. The channel placement guidelines are only described for Gen1 and Gen2 PIPE configuration. The channel placement guidelines for Gen1 and Gen2 Hard IP configuration are not included.

Transceiver Clocking for PCIe Gen1 and Gen2

PIPE x1 Configuration

The high-speed serial clock is provided by the CMU PLL in a channel different from that of the data channel. The local clock divider block in the data channel generates a parallel clock from this high-speed clock and distributes both clocks to the PMA and PCS of the data channel.

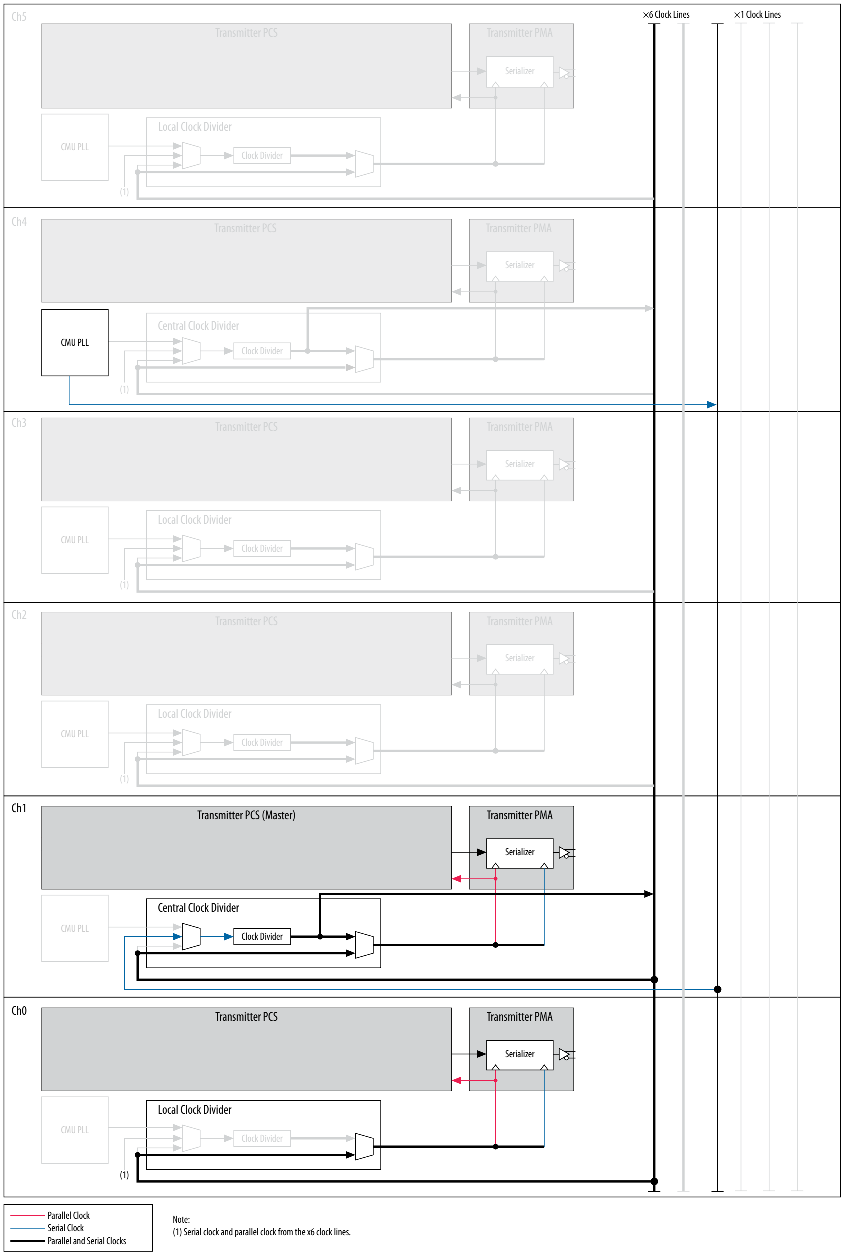
Figure 6-21: Transceiver Clocking in a Gen1/Gen2 PIPE x1 Configuration



PIPE x2 Configuration

In a PIPE x2 bonded configuration, clocking within the PCS is independent for each receiver channel. Clocking is bonded only for transmitter channels, while the control signals are bonded for both transmitter and receiver channels. The Quartus II software automatically places the transmit CMU PLL and master channel in either channel 1 or channel 4 within a transceiver bank

Figure 6-22: Transmitter Clocking in a Gen1/Gen2 PIPE x2 Configuration

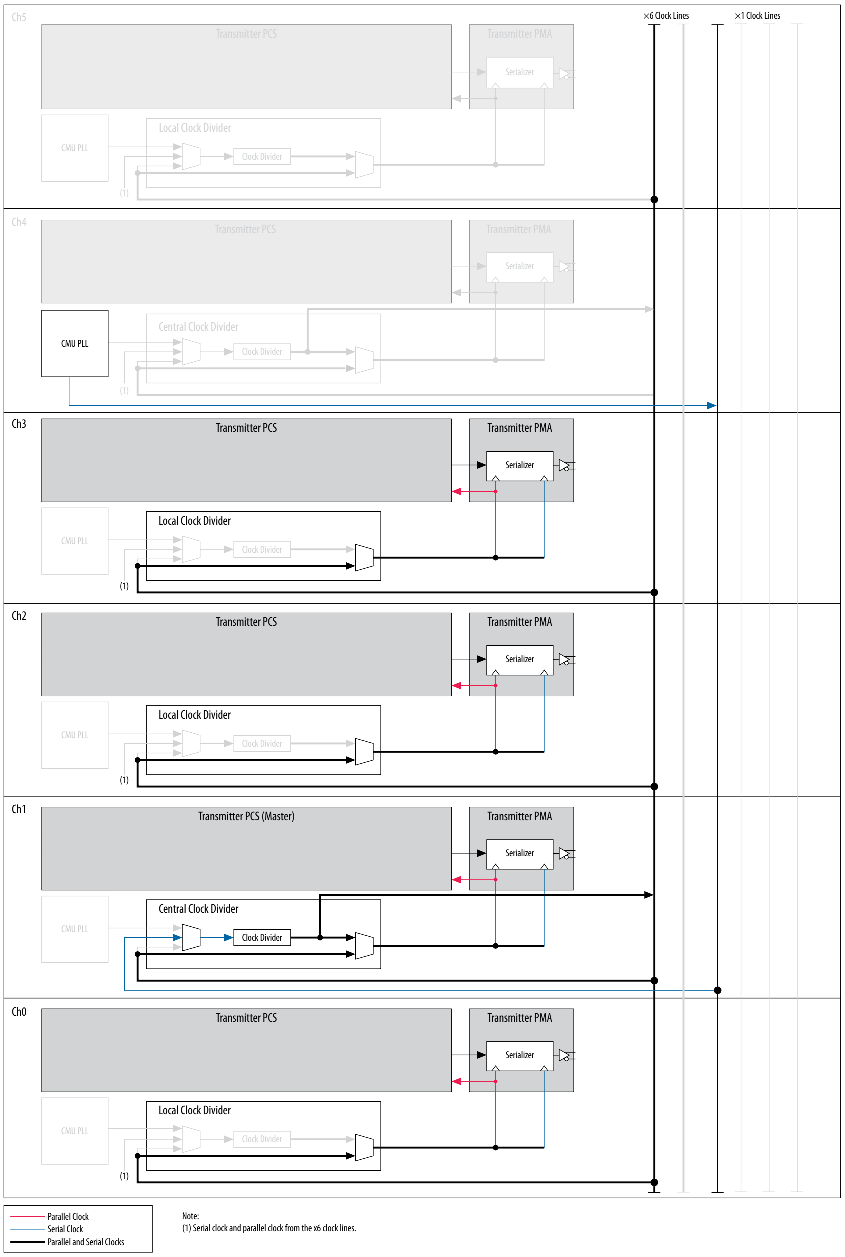


Note:
(1) Serial clock and parallel clock from the x6 clock lines.

PIPE x4 Configuration

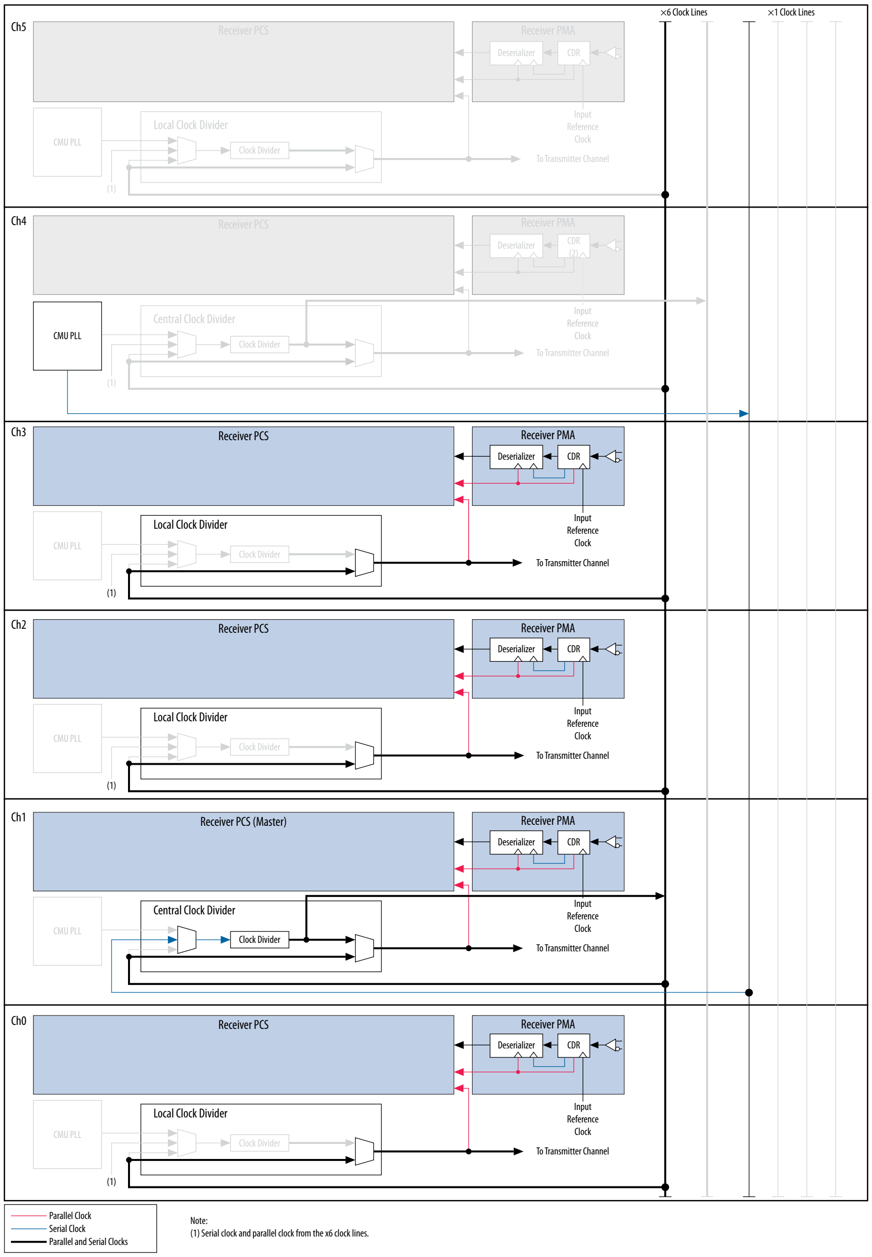
In a PIPE x4 bonded configuration, clocking within the PCS is independent for each receiver channel. Clocking is bonded only for transmitter channels, while the control signals are bonded for both transmitter and receiver channels. The Quartus II software automatically places the transmit CMU PLL and master channel in either channel 1 or channel 4 within a transceiver bank.

Figure 6-23: Transmitter Clocking in a Gen1/Gen2 PIPE x4 Configuration



Note:
(1) Serial clock and parallel clock from the x6 clock lines.

Figure 6-24: Receiver Clocking in a Gen1/Gen2 PIPE x4 Configuration

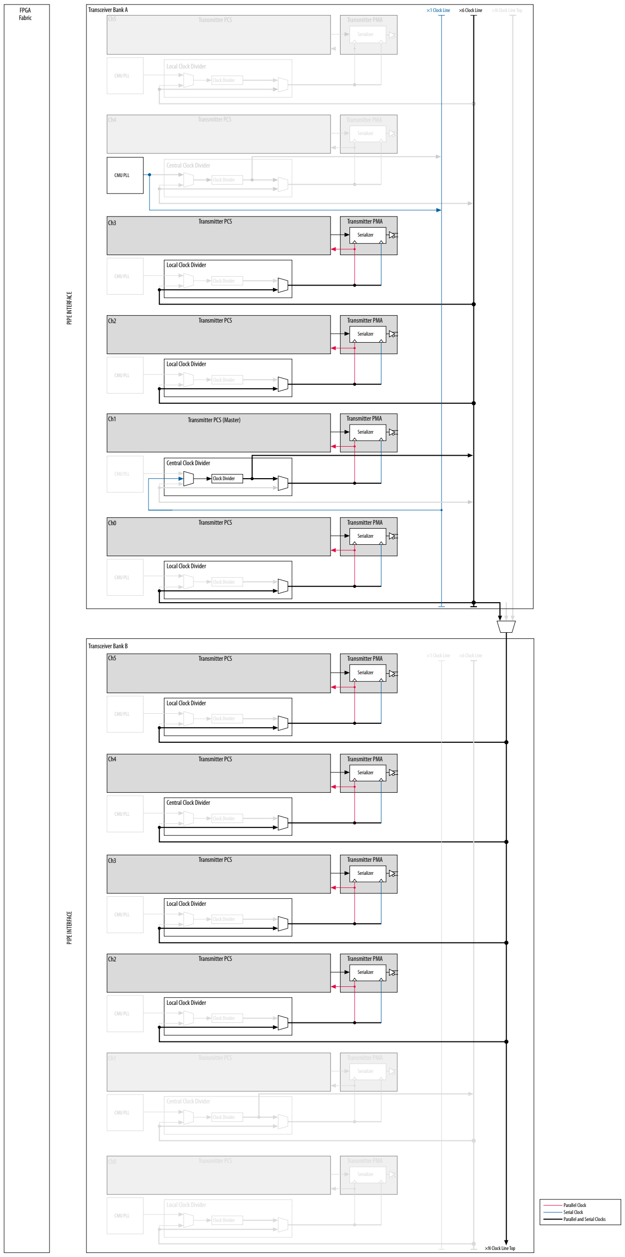


Note:
(1) Serial clock and parallel clock from the x6 clock lines.

PIPE x8 Configuration

In the x8 PCIe bonded configuration, clocking is independent for receiver channels. Clocking and control signals are bonded only for transmitter channels.

Figure 6-25: Transceiver Clocking in a Gen1/Gen2 PIPE x8 Configuration



Transceiver Channel Placement Guidelines for Gen1, Gen2, and Gen3 PIPE Configurations

Note: The channel placement guidelines are only described for Gen1, Gen2, and Gen3 x1, x2, x4, and x8 PIPE configurations. The channel placement guidelines for Gen1, Gen2, and Gen3 Hard IP configuration are not included.

The following table lists the physical placement of PIPE channels in x1, x2, x4, and x8 bonding configurations. The Quartus® II software automatically places the CMU PLL in a channel different from that of the data channels.

Table 6-7: PIPE Configuration Channel Placement

Placement by the Quartus II software may vary with design, thus resulting in higher channel usage.

Configuration	Data Channel Placement	Channel Utilization Using CMU PLL in Gen1 and Gen2	Channel Utilization Using ATX PLL in Gen1 and Gen2	Channel Utilization Using CMU and ATX PLL in Gen3
x1	Any channel	2	1	2
x2	Contiguous channels	3	2	3
x4	Contiguous channels	5	4	5
x8	Contiguous channels	9	8	9

Channel Placement for Gen1, Gen2, and Gen3 x1 PIPE Configuration

For PIPE x1 configurations, the channel can be placed anywhere within a transceiver bank that contains the transmitter PLL. In Gen1 and Gen2 configurations, you can select either the ATX PLL or the CMU PLL as the transmitter PLL. In Gen3 configurations, a CMU PLL is used for Gen1 and Gen2 datarates and an ATX PLL is used for Gen3 datarates.

Channel Placement for Gen1, Gen2, and Gen3 x2 and x4 PIPE Configuration

The following two figures show examples of channel placement for PIPE x2 and x4 configurations. In a PIPE x2 or x4 configuration, the two or four channels must be contiguous and within the same transceiver bank, but they can be placed in any order as long as Logical Lane 1 is placed on the master channel. In Gen1 and Gen2 configurations, you can select either the ATX PLL or the CMU PLL as the transmitter PLL. In Gen3 configurations, a CMU PLL is used for Gen1 and Gen2 datarates and an ATX PLL is used for Gen3 datarates. The CMU PLL and/or ATX PLL must be within the same transceiver bank as the master channel.

In the figures, channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock. Channels shaded in gray are data channels. The Quartus II software automatically selects one of the following within a transceiver bank:

- The CMU PLL in either channel 1 or channel 4.
- The upper or lower ATX PLL if the ATX PLL is selected as the transmitter PLL within the transceiver bank containing the master channel.

Gen3 channel placement requires both a CMU and an ATX PLL in the same transceiver bank as the master channel.

Figure 6-26: Example of PIPE x2 Gen1, Gen2, and Gen3 Channel Placement Using an ATX PLL, a CMU PLL, or Both

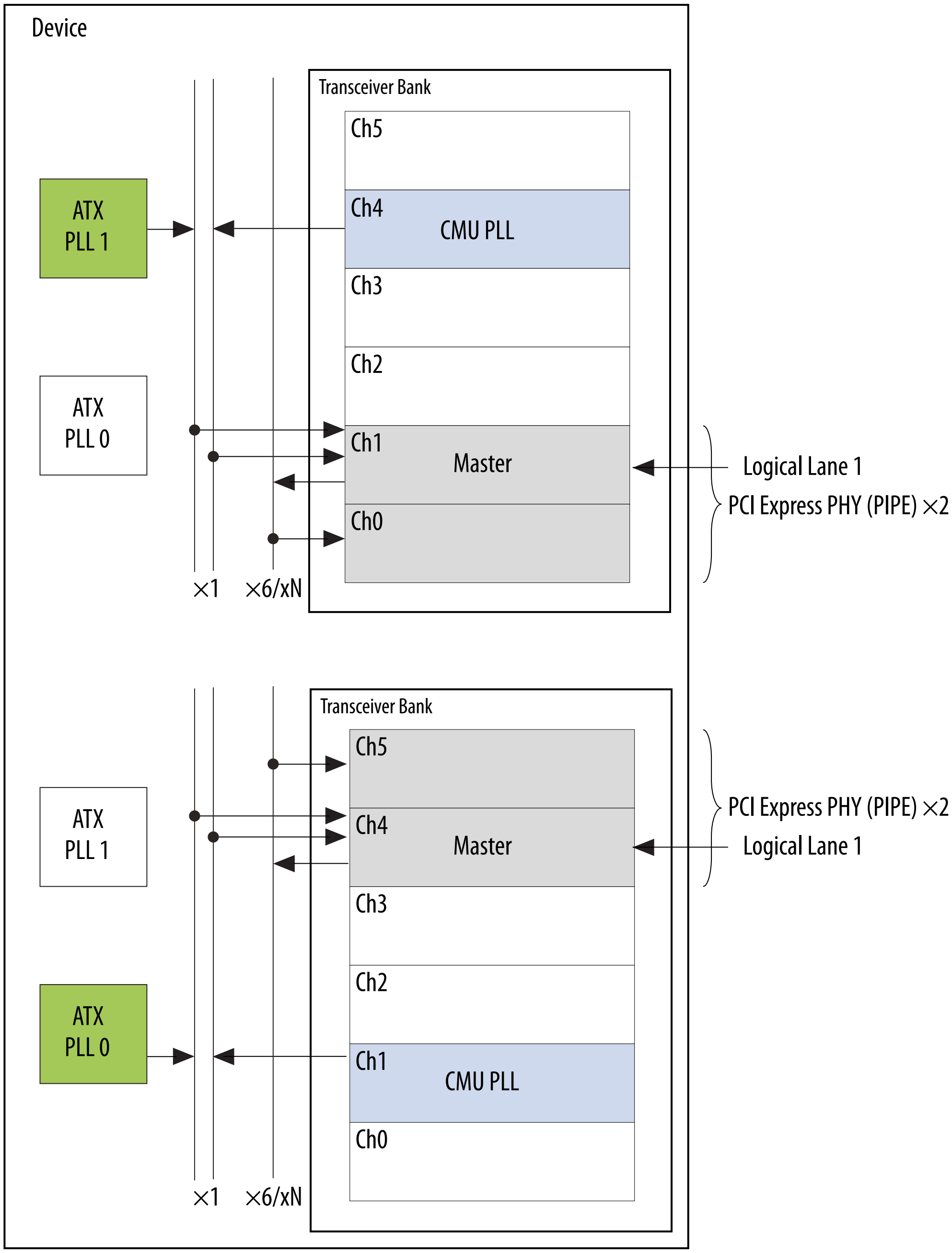
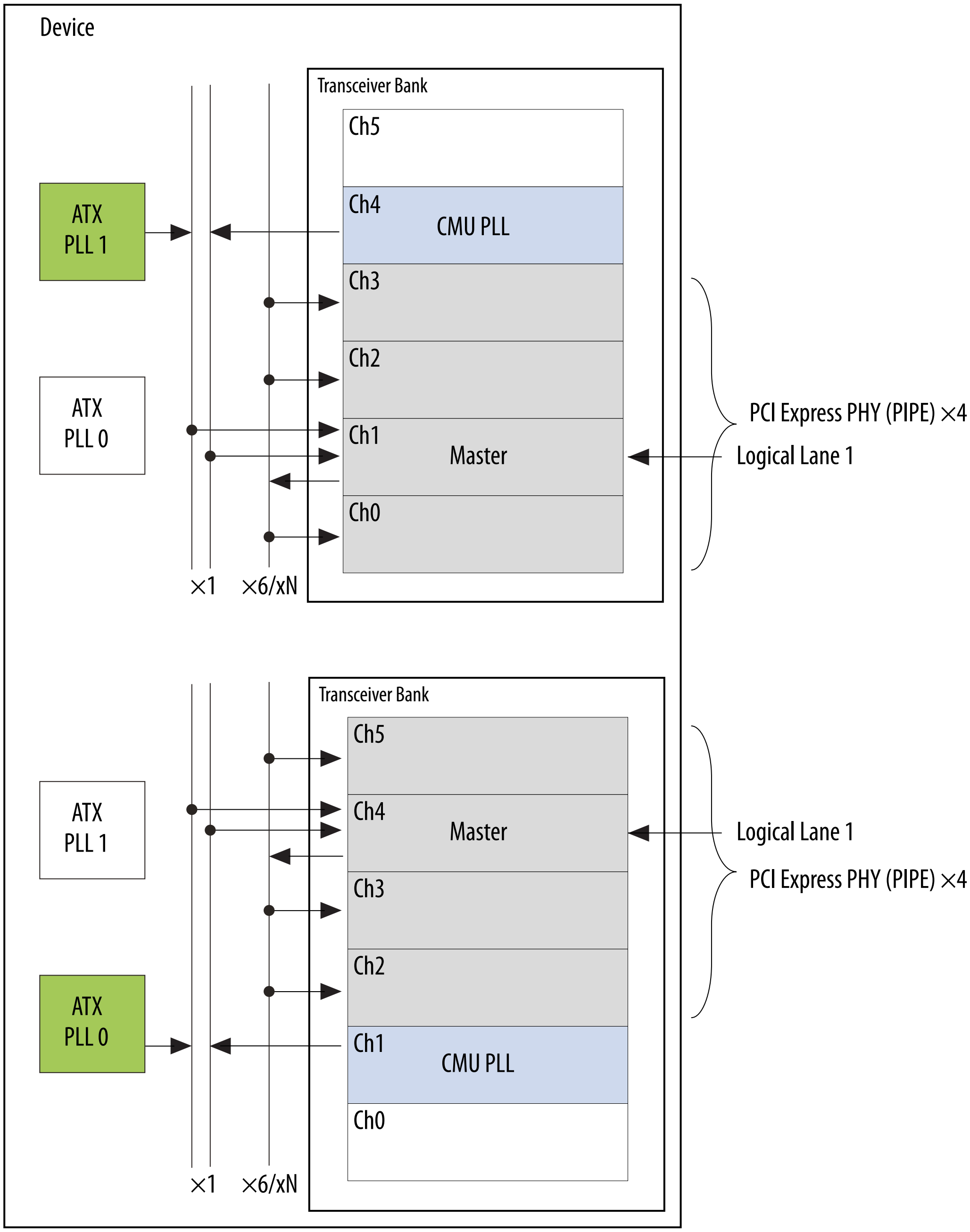


Figure 6-27: Example of PIPE x4 Gen1, Gen2, and Gen3 Channel Placement Using an ATX PLL, a CMU PLL, or Both

Channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock. Channels shaded in gray are data channels. The Quartus II software automatically selects the CMU PLL in either channel 1 or channel 4 within a transceiver bank. Gen3 channel placement requires an additional ATX PLL in the same transceiver bank as the master channel.



Channel Placement for Gen1, Gen2, and Gen3 x8 PIPE Configuration

In a PIPE x8 configuration, the eight channels must be contiguous, but they can be placed in any order as long as Logical Lane 0 is placed on the master channel.

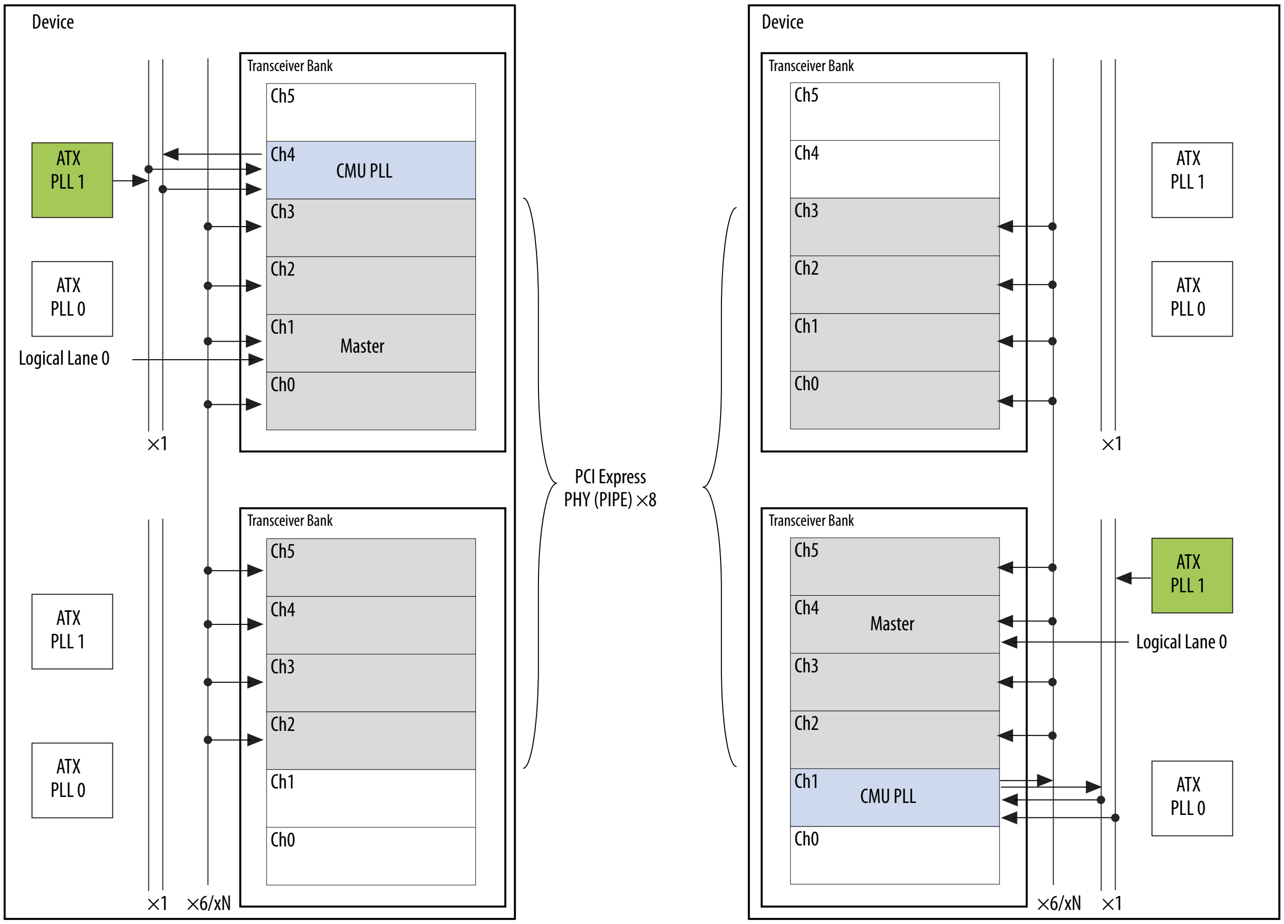
The Quartus II software automatically selects one of the following within a transceiver bank:

- The CMU PLL in either channel 1 or channel 4.
- The upper or lower ATX PLL if the ATX PLL is selected as the transmitter PLL within the transceiver bank containing the master channel.

In Gen1 and Gen2 configurations, you can select either the ATX PLL or the CMU PLL as the transmitter PLL. In Gen3 configurations, a CMU PLL is used for Gen1 and Gen2 datarates and an ATX PLL is used for Gen3 datarates. The CMU PLL and/or ATX PLL must be within the same transceiver bank.

Figure 6-28: Example of PIPE x8 Gen1, Gen2, and Gen3 Channel Placement Using an ATX PLL, a CMU PLL, or Both

Channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock. Channels shaded in gray are data channels. Gen3 channel placement requires both a CMU and ATX PLL in the same transceiver bank as the master channel.



Related Information

For channel placement guidelines for PCIe hard IP configuration using the Hard IP for PCI Express, refer to the Arria V Hard IP for PCI Express User Guide.

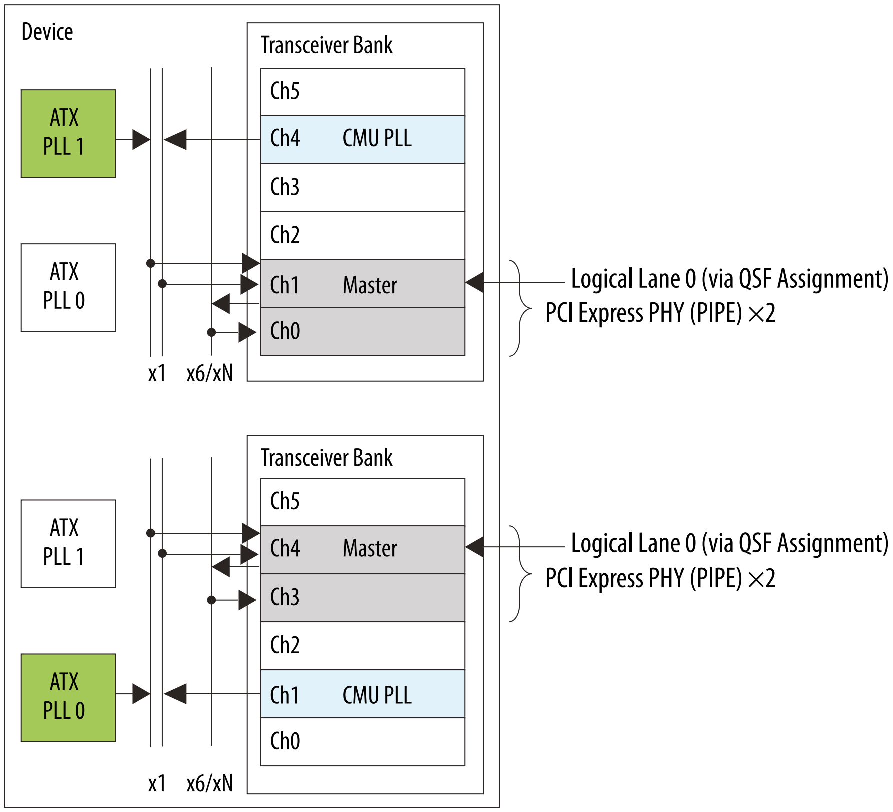
Advanced Channel Placement Guidelines for PIPE Configurations

Advanced channel placement options for PIPE configurations are enabled through Quartus Settings File (QSF) assignments. A QSF assignment allows you to override the master channel assignment. By using a QSF assignment, master channels can be assigned any logical channel number instead of the default Quartus II logical lane assignment. Any PIPE channel placement can also be made compatible with the HIP configuration channel placement.

In the following figures, channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock. Channels shaded in gray are data channels. An ATX PLL shaded in green can be substituted for the CMU PLL for Gen1 and Gen2 configurations only. Gen3 channel placement requires both the CMU PLL for Gen1/Gen2 datarates and the ATX PLL for Gen3 datarates to be located in the same transceiver bank as the master channel. The Quartus II software automatically selects the CMU PLL in either channel 1 or channel 4 and/or the upper or lower ATX PLL within a transceiver bank.

Advanced Channel Placement for PIPE x2 Gen1, Gen2, and Gen3 Configurations

Figure 6-29: PIPE x2 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL



Advanced Channel Placement for PIPE x4 Gen1, Gen2, and Gen3 Configurations

Figure 6-30: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL in the Same Transceiver Bank

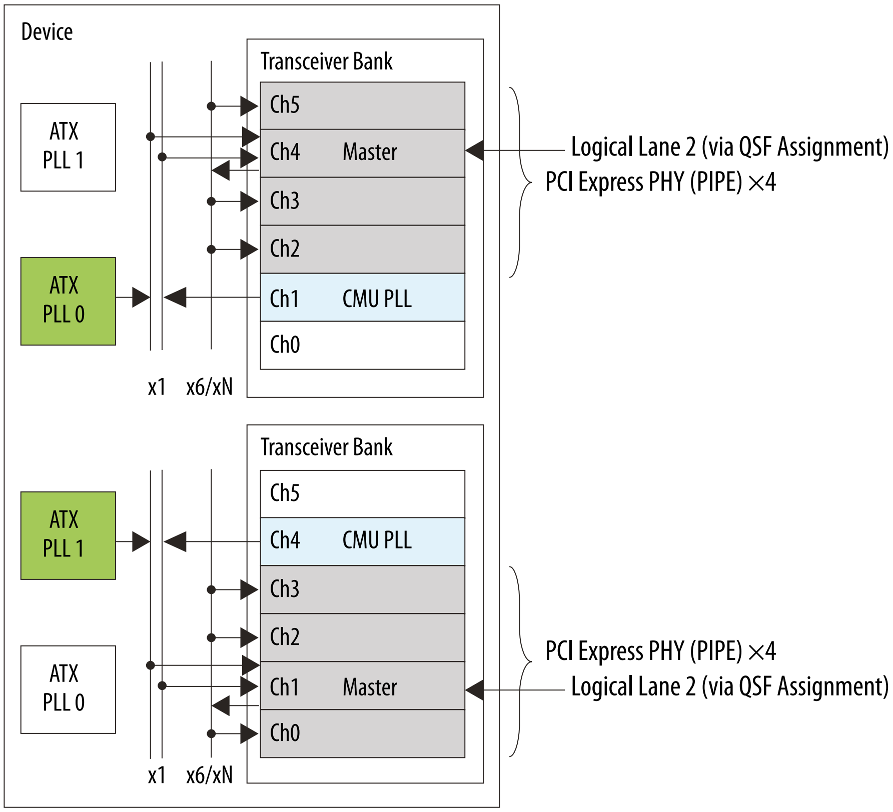


Figure 6-31: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL Across Two Transceiver Banks – example 1

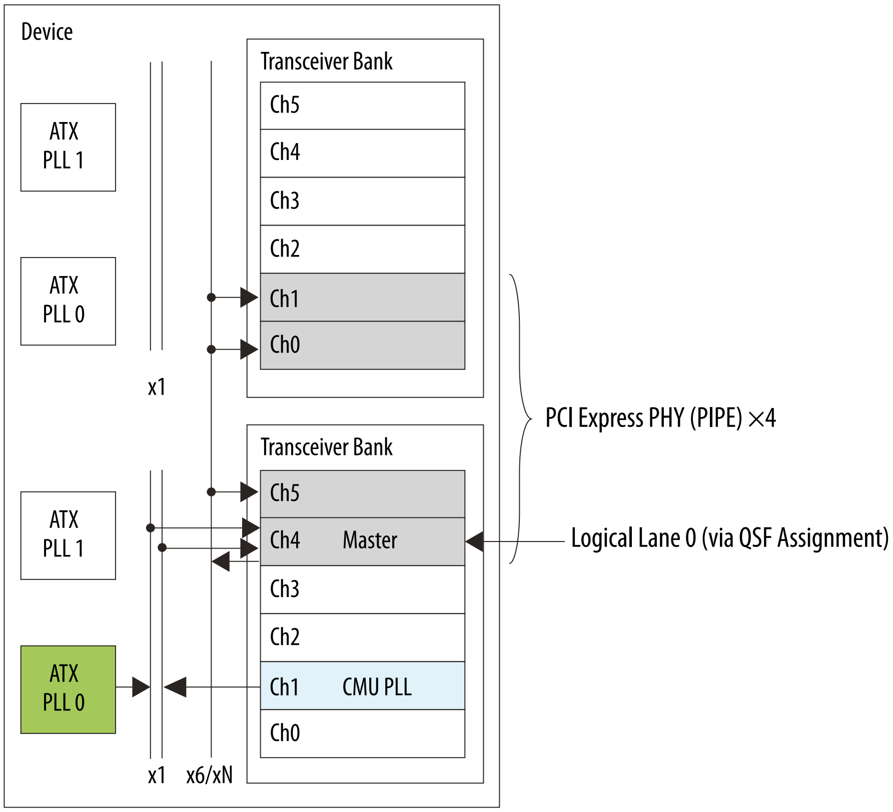
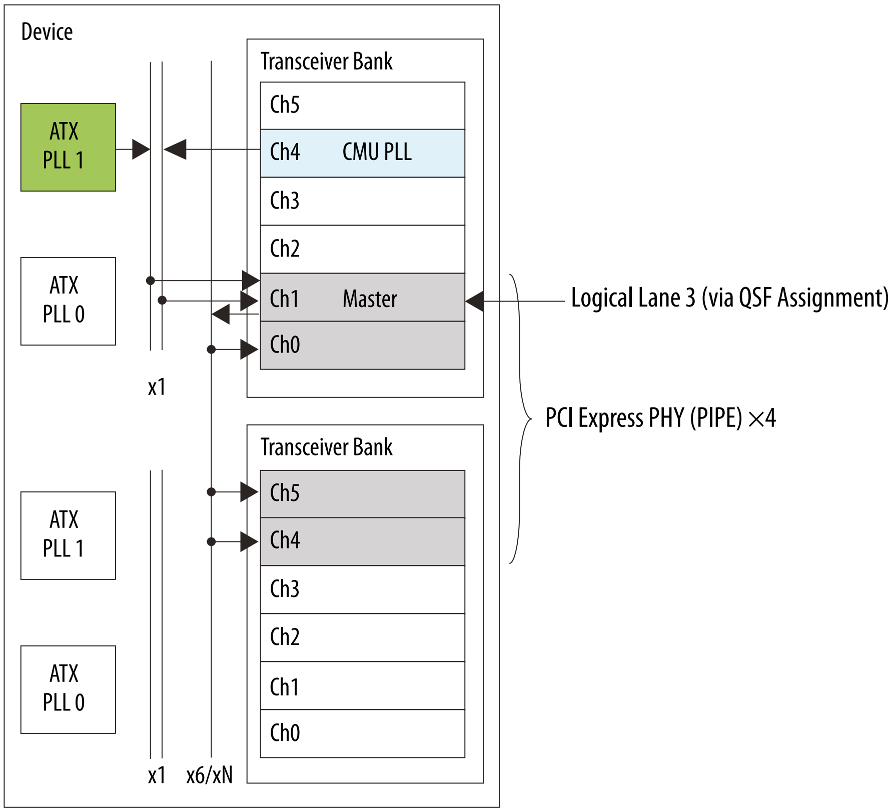


Figure 6-32: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL Across Two Transceiver Banks – example 2



Advanced Channel Placement for PIPE x8 Gen1, Gen2, and Gen3 Configurations

For PCIe x8 advanced channel placement where the master channel resides between the contiguous data channel assignments, a second QSF assignment is required that allows the master channel to be placed between data channels.

For a HIP-compatible PCIe x8 channel placement, the master channel must be assigned logical channel 4 in the lower transceiver bank and the second QSF assignment for the reserve channel that allow master channel placement between contiguous data channel assignments are required.

Figure 6-33: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement That is Compatible with HIP x8 Channel Placement

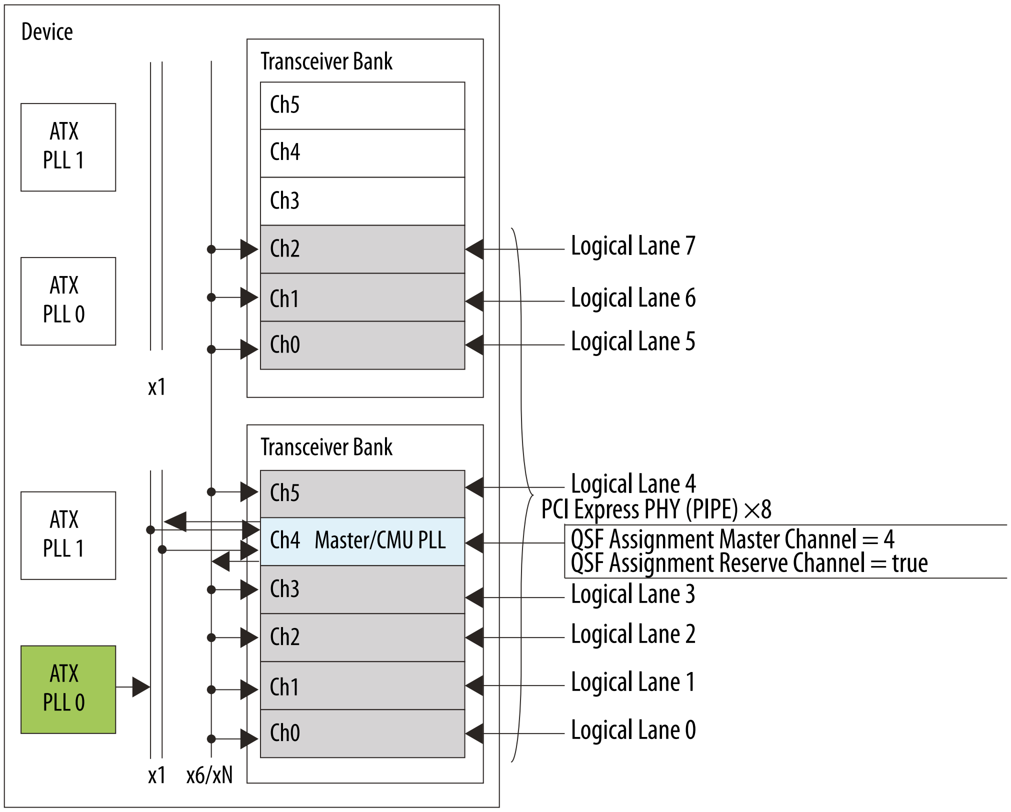
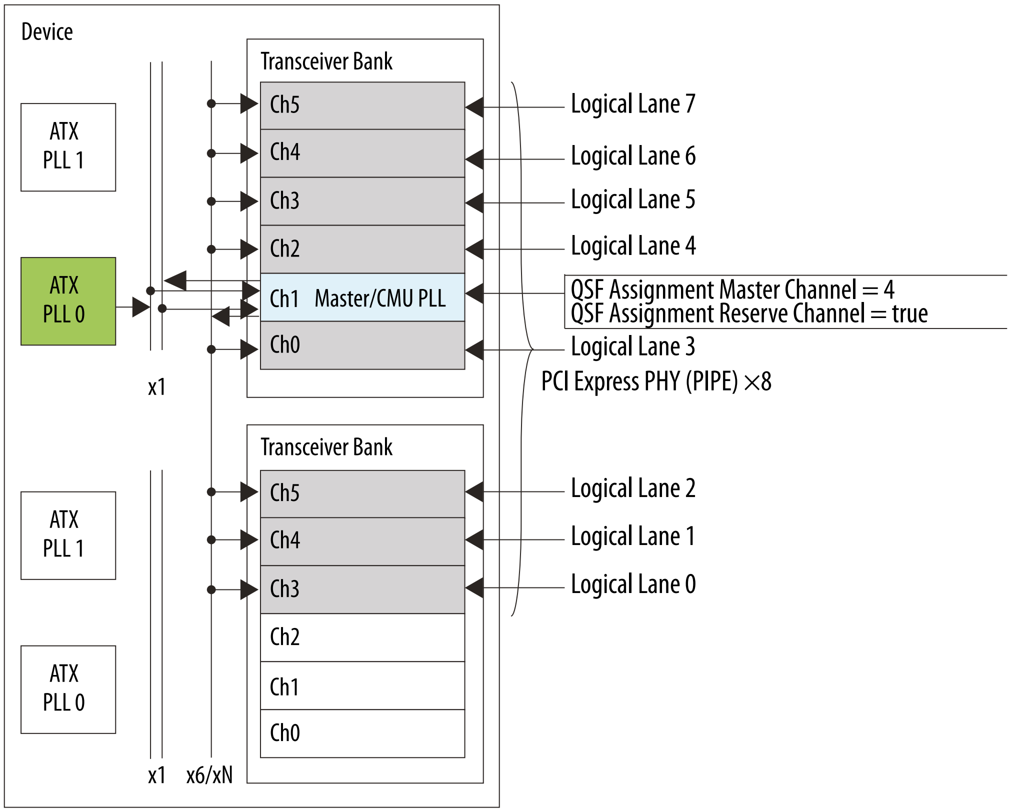


Figure 6-34: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement That is Not Compatible with HIP x8 Channel Placement



The following figures show PIPE x8 Gen1, Gen2, and Gen3 advanced channel placement that requires only a master channel QSF assignment.

Figure 6-35: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 1

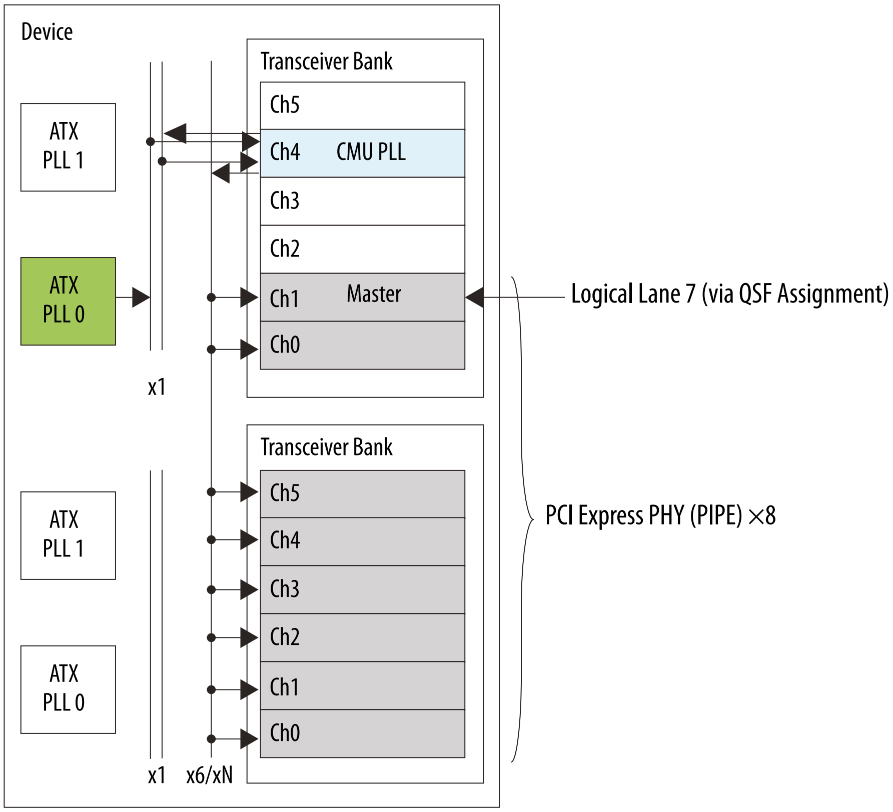


Figure 6-36: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 2

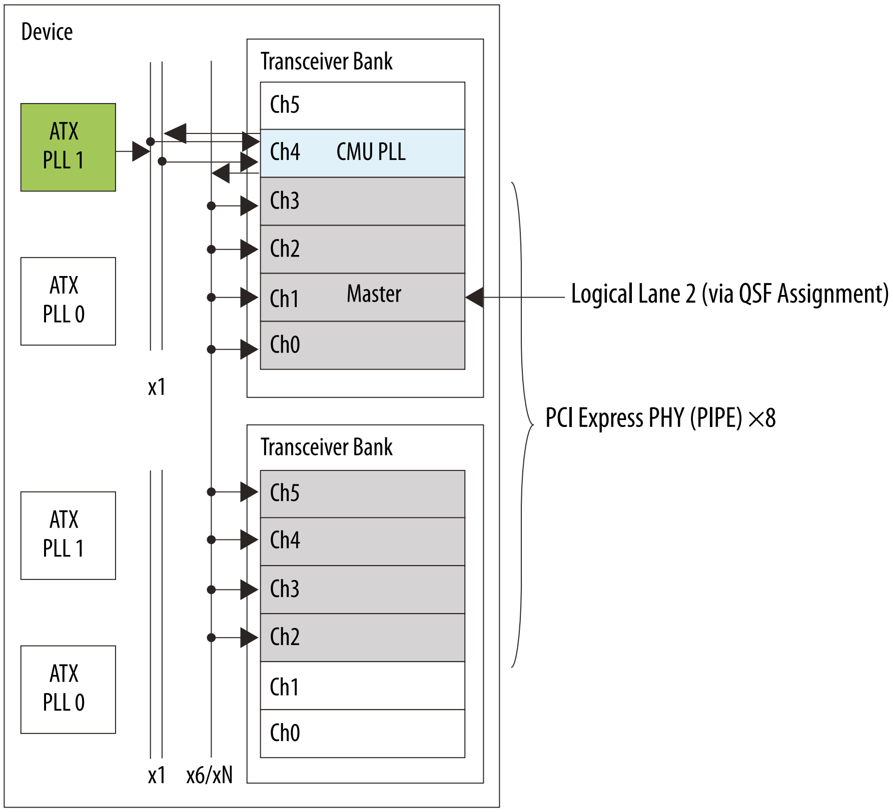
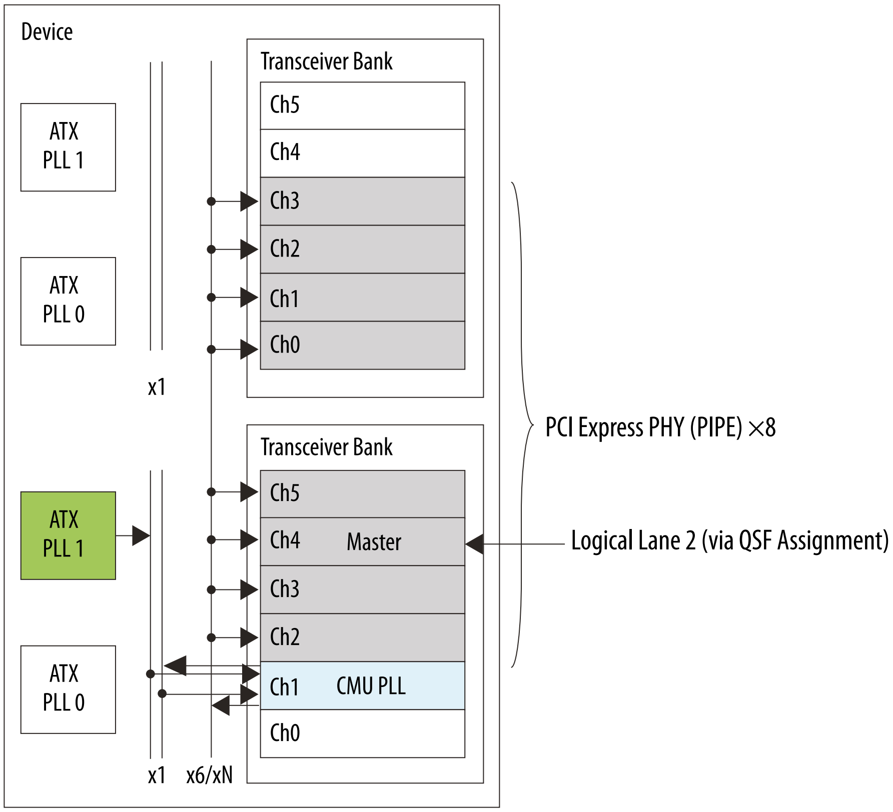


Figure 6-37: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 3



Transceiver Clocking for PCIe Gen3

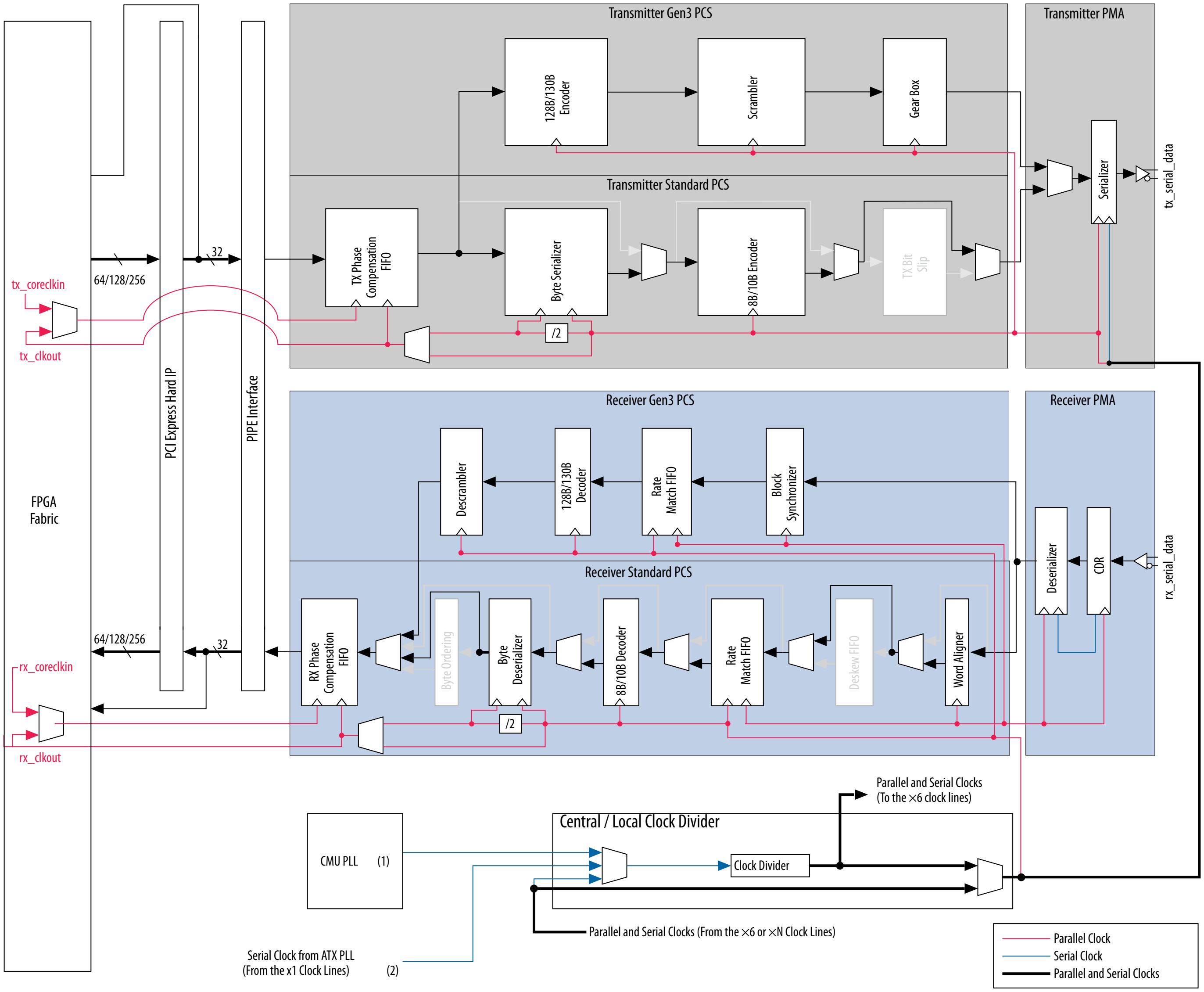
This section describes the transceiver clocking topology for both the PCIe Gen3 Hard IP and PIPE configuration.

In a PCIe x1, x2, x4, and x8 Gen3 Mode, both a channel PLL (CMU PLL) from transceiver physical channel 1 or 4 of the transceiver bank and either the top or bottom ATX PLL are used to generate the high-speed serial clock and support ASN. The CMU PLL supports Gen1 and Gen2 data rates while the ATX PLL supports Gen3 data rates. To enable rapid switching between Gen1, Gen2, and Gen3 data rates, a multiplexer selects either the free running CMU PLL for Gen1 and Gen2 data rates or the free running ATX PLL for Gen3 data rates. PLL reconfiguration is not used to support ASN.

Gen3 x1 Configuration

Figure 6-38: Transceiver Clocking in a Gen1/Gen2/Gen3 PCIe x1 Hard IP and PIPE Configuration

For Gen1 and Gen2, use the CMU PLL. For Gen3, use the ATX PLL.



For **PCIe x1 Gen3** using Hard IP configuration, the CMU PLL (transceiver physical channel 1) and the bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock for the transmitter datapath clock and the rate matcher side of the FIFO in the receiver datapath if rate matching is enabled for the data channel. Two transceiver channels are needed to implement PCIe x1 Gen3, one for the data channel and one for the CMU PLL. The local clock divider block in the data channel generates a parallel clock from this high-speed serial clock and distributes both clocks to the PMA and PCS of the data channel.

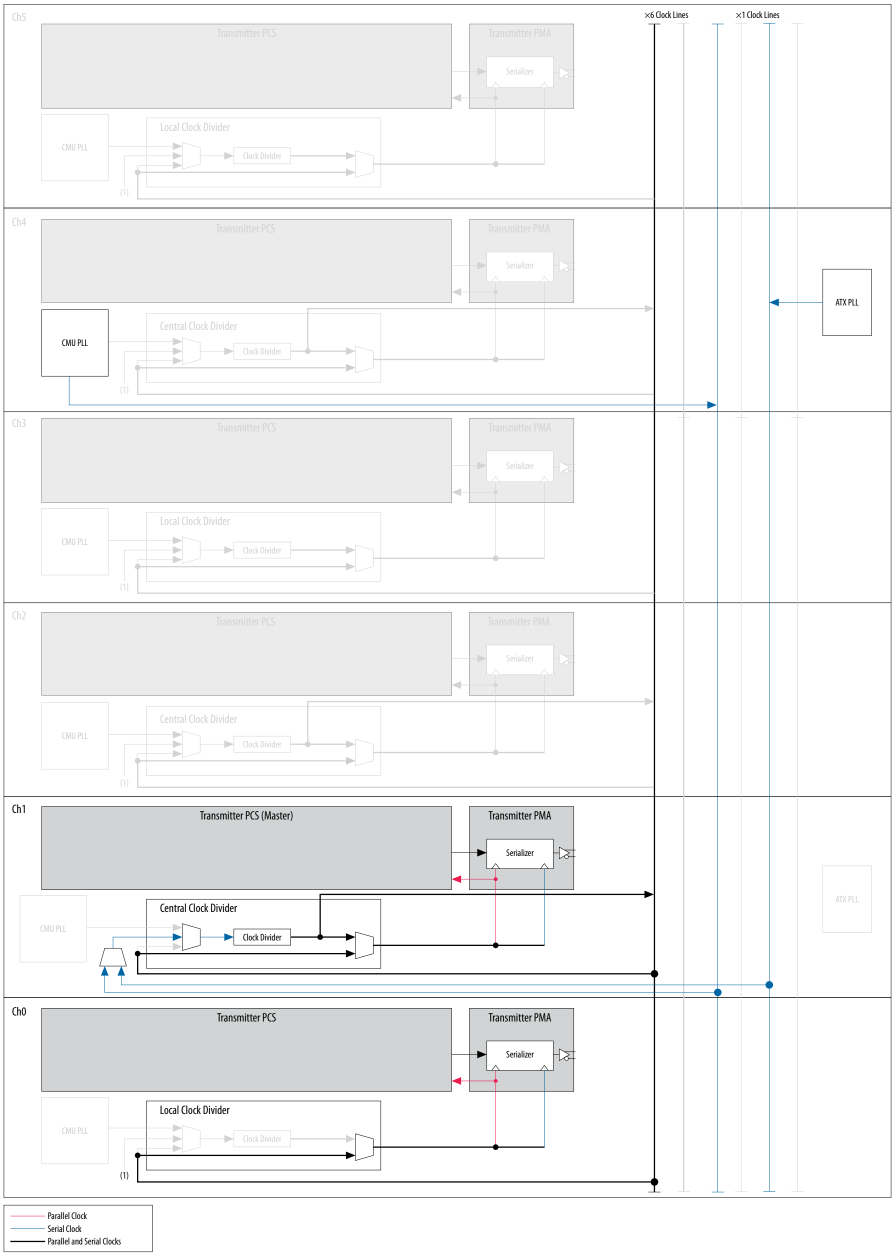
For **PCIe x1 Gen3** using PIPE configuration, the CMU PLL (transceiver physical channel 1 or 4) and the top or bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock for the transmitter datapath clock and the rate matcher side of the FIFO in the receiver datapath if rate matching is enabled for the data channel. Two transceiver channels are needed to implement PCIe x1

Gen3, one for the data channel and one for the CMU PLL. The local clock divider block in the data channel generates a parallel clock from this high-speed serial clock and distributes both clocks to the PMA and PCS of the data channel.

Gen3 x2 Configuration

Figure 6-39: Transmitter Clocking in a Gen1/Gen2/Gen3 PCIe x2 Hard IP and PIPE Configuration

Unlike the Hard IP configuration, the PIPE configuration has the additional flexibility of using the top four transceiver channels in a transceiver bank or spanning the four lanes across two banks.



For **PCIe x2 Gen3** using Hard IP configuration, the CMU PLL (transceiver physical channel 4) and the top ATX PLL of the transceiver bank are configured to generate the high-speed serial clock. A total of three transceiver channels are required to implement PCIe x2 Gen3, including two data channels and one channel for the CMU PLL. The Quartus II software automatically selects channel 1 in the transceiver bank as the master channel. Channel 1 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the two data channels. The local clock divider block in each data channel generates the parallel clock from the high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel.

For **PCIe x2 Gen3** using PIPE configuration, the CMU PLL (transceiver physical channel 1 or 4) and the top or bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock. A total of three transceiver channels are required to implement PCIe x2 Gen3, including two data channels and one channel for the CMU PLL. The Quartus II software automatically selects either channel 1 or 4 in the transceiver bank as the master channel. Channel 1 or 4 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the two data channels. The local clock divider block in each data channel generates the parallel clock from the high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel.

Gen3 x4 Configuration

Figure 6-40: Transmitter Clocking in a Gen1/Gen2/Gen3 PCIe x4 Hard IP and PIPE Configuration

Unlike the Hard IP configuration, the PIPE configuration has the additional flexibility of using the top four transceiver channels in a transceiver bank or spanning the four lanes across two banks.

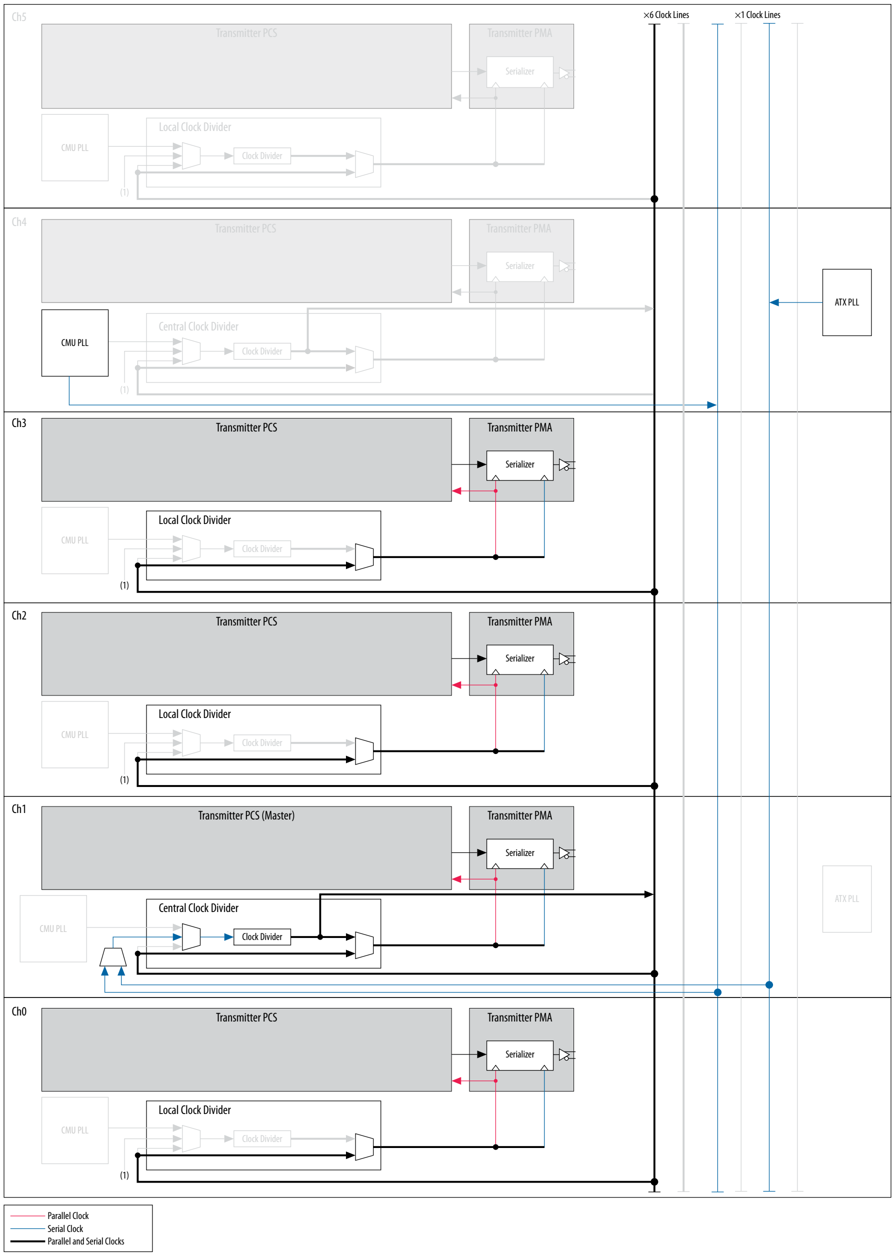
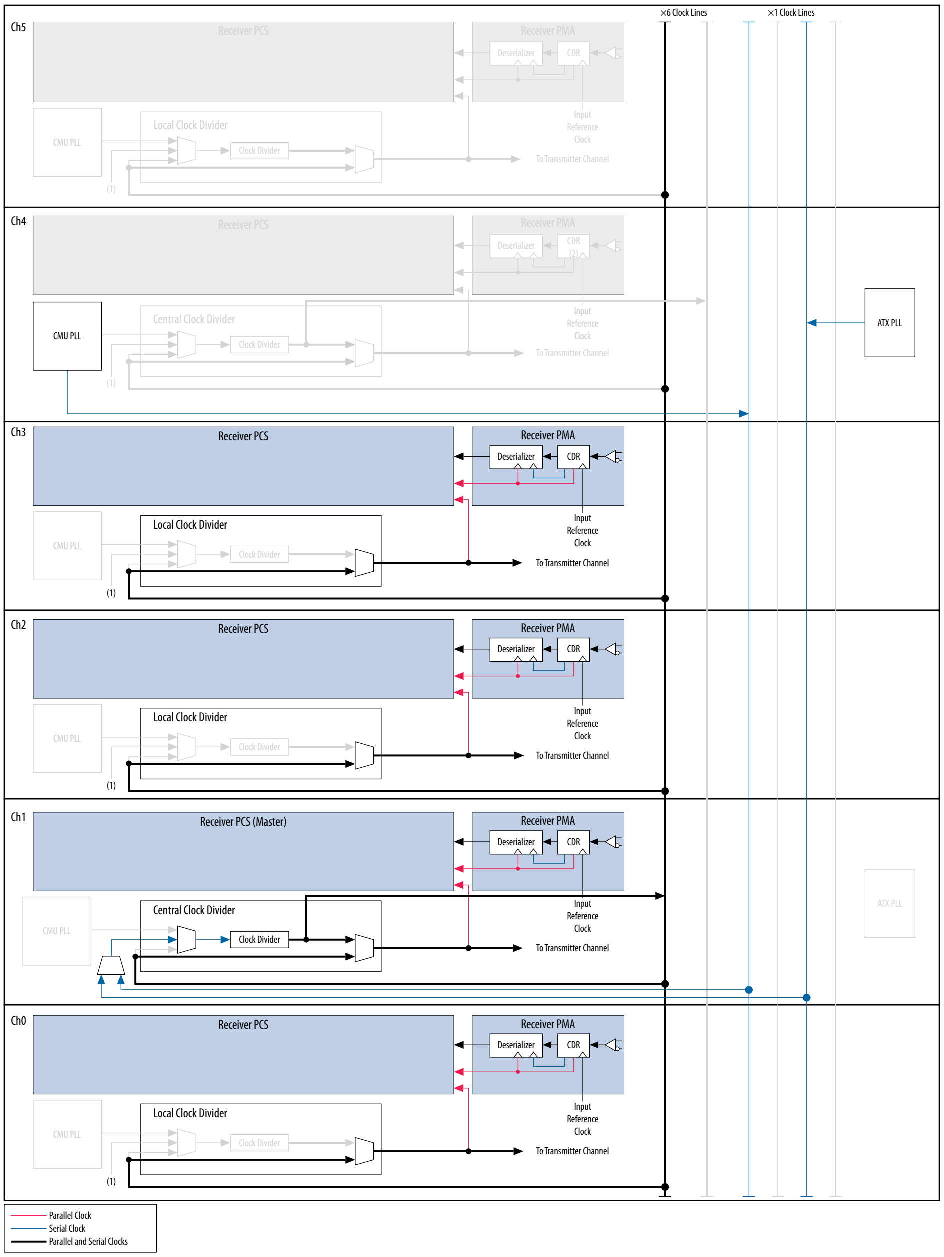


Figure 6-41: Receiver Clocking in a Gen1/Gen2/Gen3 PCIe x4 Hard IP and PIPE Configuration



For **PCIe x4 Gen3** using Hard IP configuration, the CMU PLL (transceiver physical channel 4) and the top ATX PLL of the transceiver bank are configured to generate the high-speed serial clock. A total of five transceiver channels are required to implement PCIe x4 Gen3, including four data channels and one channel for the CMU PLL. The Quartus II software automatically selects channel 1 in the transceiver bank as the master channel. Channel 1 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the four data channels. The local clock divider block in each data channel generates the parallel clock from the high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel.

For **PCIe x4 Gen3** using PIPE configuration, the CMU PLL (transceiver physical channel 1 or 4) and the top or bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock. A total of five transceiver channels are required to implement PCIe x4 Gen3, including four data channels and one channel for the CMU PLL. The Quartus II software automatically selects either channel 1 or 4 in the transceiver bank as the master channel. Channel 1 or 4 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the four data channels. The local clock divider block in each data channel generates the parallel clock from the high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel.

Gen3 x8 Configuration

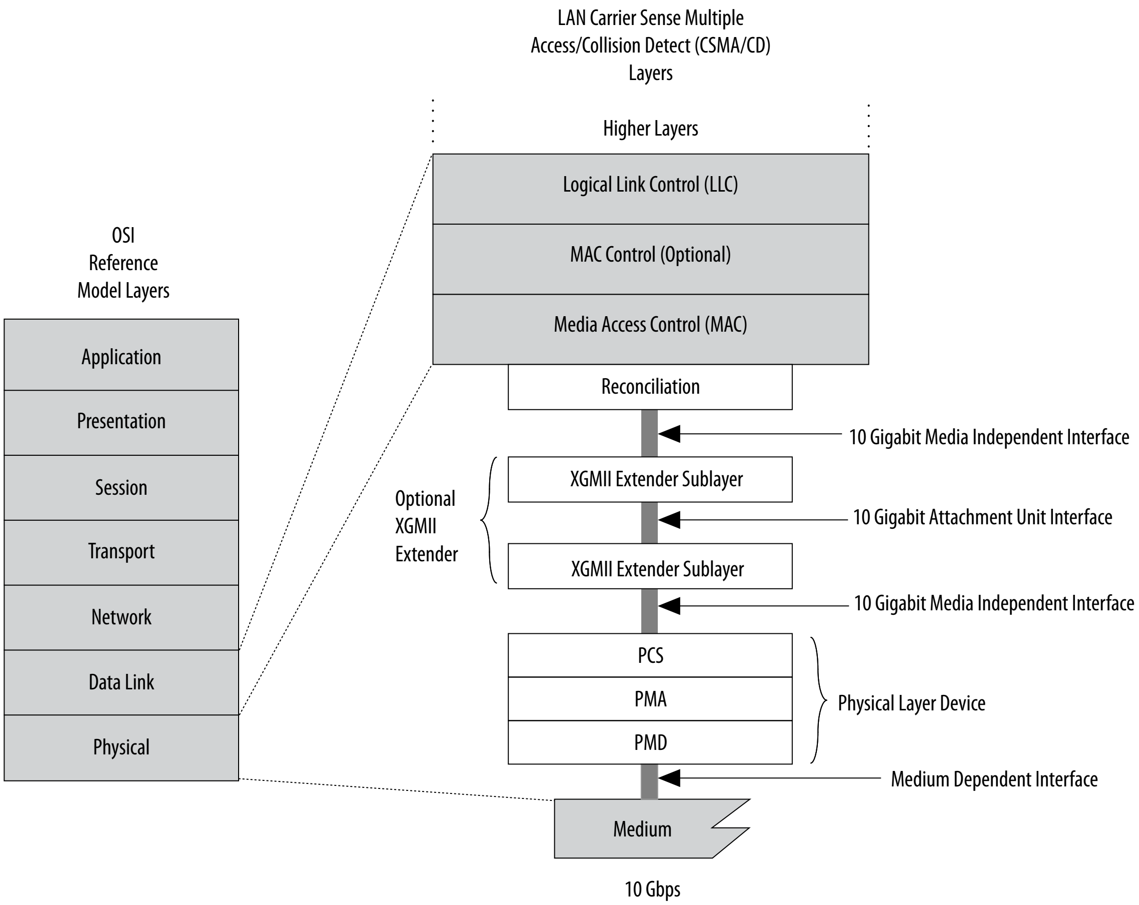
For **PCIe x8 Gen3**, the CMU PLL (transceiver physical channel 4) and the top or bottom ATX PLL of the lower transceiver bank are configured to generate the high-speed serial clock. A total of nine transceiver channels are required to implement PCIe x8 Gen3, including eight data channels and one channel for the CMU PLL. The Quartus II software automatically selects channel 4 in the transceiver bank as the master channel. Channel 4 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the eight data channels. The local clock divider blocks in each data channel generates the parallel clock from this high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel. The master channel in the x8 case is not a data channel.

XAUI

To implement a XAUI link, instantiate the **XAUI PHY IP** core in the IP Catalog, under **Ethernet** in the Interfaces menu. The XAUI PHY IP core implements the XAUI PCS in soft logic.

XAUI is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in the IEEE 802.3ae-2002 specification. The XAUI PHY uses the XGMII interface to connect to the IEEE802.3 MAC and Reconciliation Sublayer (RS). The IEEE 802.3ae-2002 specification requires the XAUI PHY link to support a 10 Gbps data rate at the XGMII interface and four lanes each at 3.125 Gbps at the PMD interface.

Figure 6-42: XAUI and XGMII Layers


LAN Carrier Sense Multiple Access/Collision Detect (CSMA/CD) Layers
Higher Layers
Logical Link Control (LLC)
MAC Control (Optional)
Media Access Control (MAC)
Reconciliation
10 Gigabit Media Independent Interface
OSI Reference Model Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical
Optional XGMII Extender
XGMII Extender Sublayer
10 Gigabit Attachment Unit Interface
XGMII Extender Sublayer
10 Gigabit Media Independent Interface
PCS
PMA
PMD
Physical Layer Device
Medium Dependent Interface
Medium
10 Gbps

Related Information

Refer to the "XAUI PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide.

Transceiver Datapath in a XAUI Configuration

The XAUI PCS is implemented in soft logic inside the FPGA core when using the XAUI PHY IP core. You must ensure that your channel placement is compatible with the soft PCS implementation.

Figure 6-43: XAUI Datapath Configuration

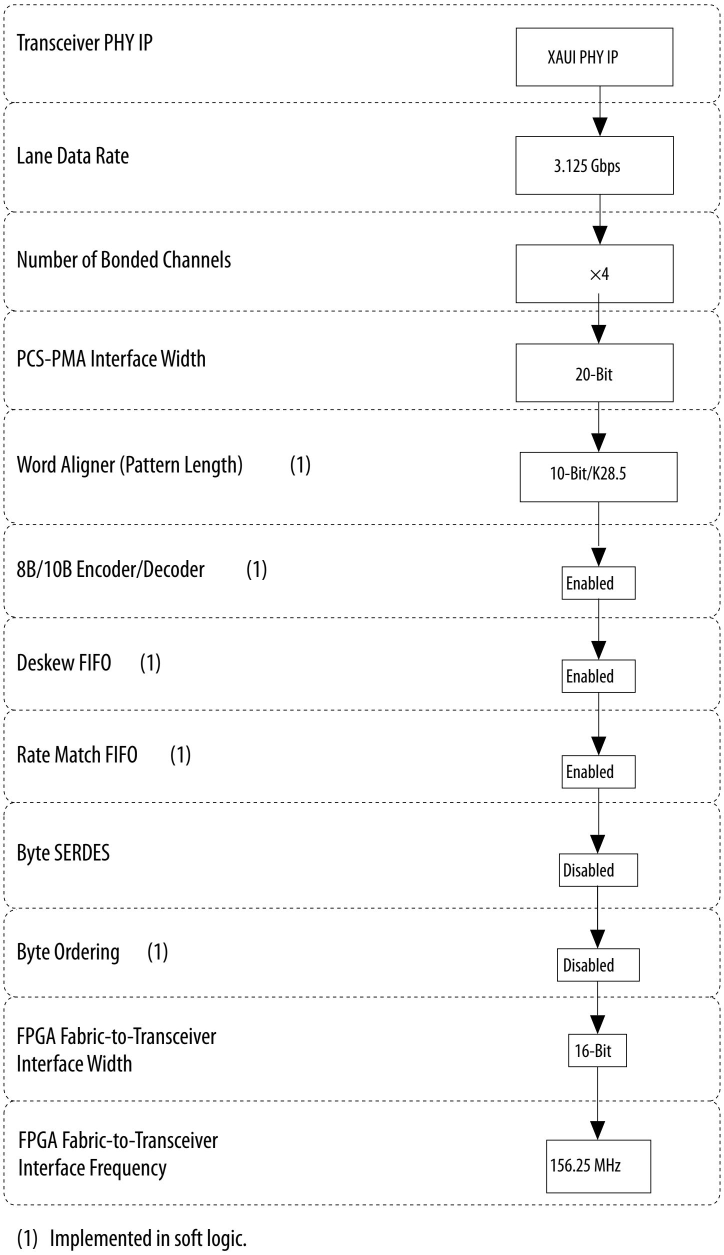
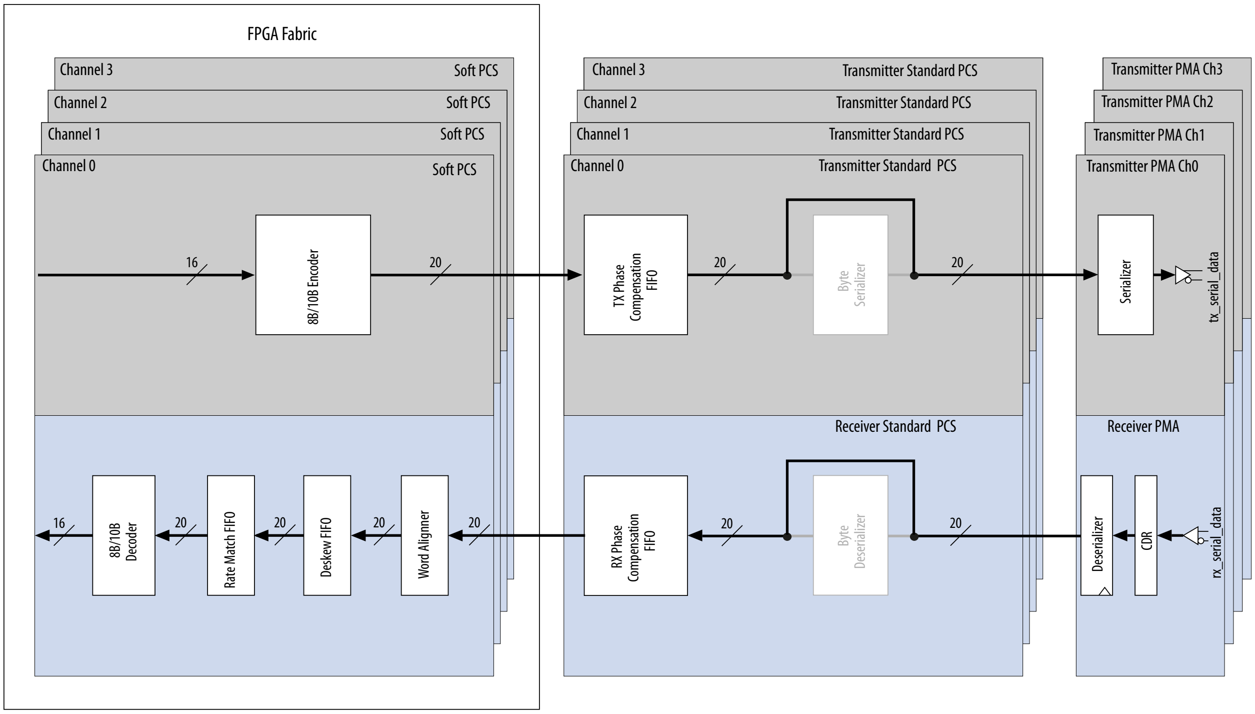


Figure 6-44: Transceiver Channel Datapath for XAUI Configuration

Standard PCS in a low latency configuration is used in this configuration. Additionally, a portion of the PCS is implemented in soft logic.



Supported Features

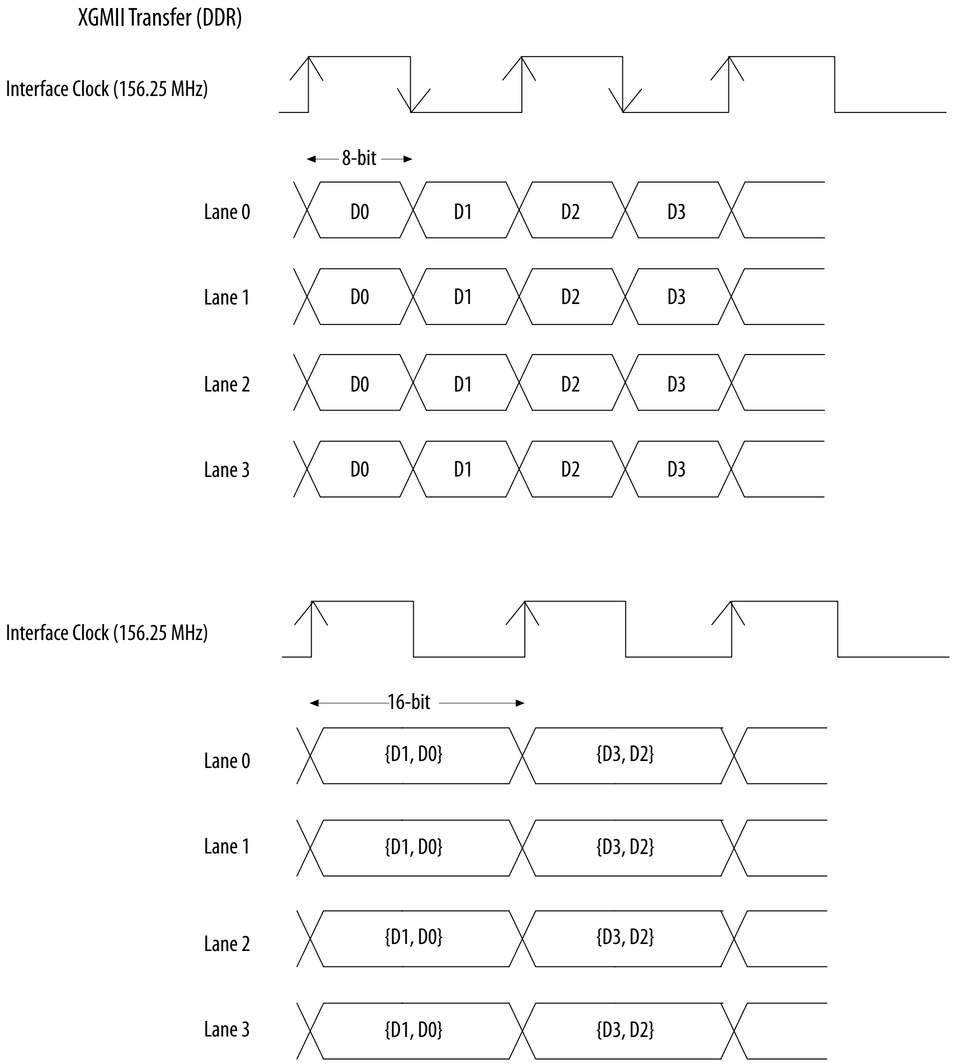
Arria V GZ transceivers support the following features in a XAUI configuration.

64-Bit SDR Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the XAUI PCS and the Ethernet MAC/RS. The specification requires each of the four XAUI lanes to transfer 8-bit data and 1-bit wide control code at both the positive and negative edge (DDR) of the 156.25 MHz interface clock.

Arria V GZ transceivers in a XAUI configuration do not support the XGMII interface to the MAC/RS as defined in IEEE 802.3-2008 specification. Instead, they allow the transferring of 16-bit data and 2-bit control code on each of the four XAUI lanes, only at the positive edge (SDR) of the 156.25 MHz interface clock.

Figure 6-45: Implementation of the XGMII Specification in Arria V GZ Devices



8B/10B Encoding/Decoding

Each of the four lanes in a XAUI configuration support an independent 8B/10B encoder/decoder as specified in Clause 48 of the IEEE802.3-2008 specification. 8B/10B encoding limits the maximum number of consecutive 1s and 0s in the serial data stream to five, thereby ensuring DC balance as well as enough transitions for the receiver CDR to maintain a lock to the incoming data.

The XAUI PHY IP core provides status signals to indicate running disparity as well as the 8B/10B code group error.

Transmitter and Receiver State Machines

In a XAUI configuration, the Arria V GZ transceivers implement the transmitter and receiver state diagrams shown in Figure 48-6 and Figure 48-9 of the IEEE802.3-2008 specification.

In addition to encoding the XGMII data to PCS code groups, in conformance with the 10GBASE-X PCS, the transmitter state diagram performs functions such as converting Idle ||I|| ordered sets into Sync ||K||, Align ||A||, and Skip ||R|| ordered sets.

In addition to decoding the PCS code groups to XGMII data, in conformance with the 10GBASE-X PCS, the receiver state diagram performs functions such as converting Sync ||K||, Align ||A||, and Skip ||R|| ordered sets to Idle ||I|| ordered sets.

Synchronization

The word aligner block in the receiver PCS of each of the four XAUI lanes implements the receiver synchronization state diagram shown in Figure 48-7 of the IEEE802.3-2008 specification.

The XAUI PHY IP core provides a status signal per lane to indicate if the word aligner is synchronized to a valid word boundary.

Deskew

The lane aligner block in the receiver PCS implements the receiver deskew state diagram shown in Figure 48-8 of the IEEE 802.3-2008 specification.

The lane aligner starts the deskew process only after the word aligner block in each of the four XAUI lanes indicates successful synchronization to a valid word boundary.

The XAUI PHY IP core provides a status signal to indicate successful lane deskew in the receiver PCS.

Clock Compensation

The rate match FIFO in the receiver PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. It does so by inserting and deleting Skip ||R|| columns, depending on the ppm difference.

The clock compensation operation begins after:

- The word aligner in all four XAUI lanes indicates successful synchronization to a valid word boundary.
- The lane aligner indicates a successful lane deskew.

The rate match FIFO provides status signals to indicate the insertion and deletion of the Skip ||R|| column for clock rate compensation.

Transceiver Clocking and Channel Placement Guidelines

Transceiver Clocking

Figure 6-46: Transceiver Clocking Diagram for XAUI Configuration

One of the two channel PLLs configured as a CMU PLL in a transceiver bank generates the transmitter serial and parallel clocks for the four XAUI channels. The x6 clock line carries the transmitter clocks to the PMA and PCS of each of the four channels.

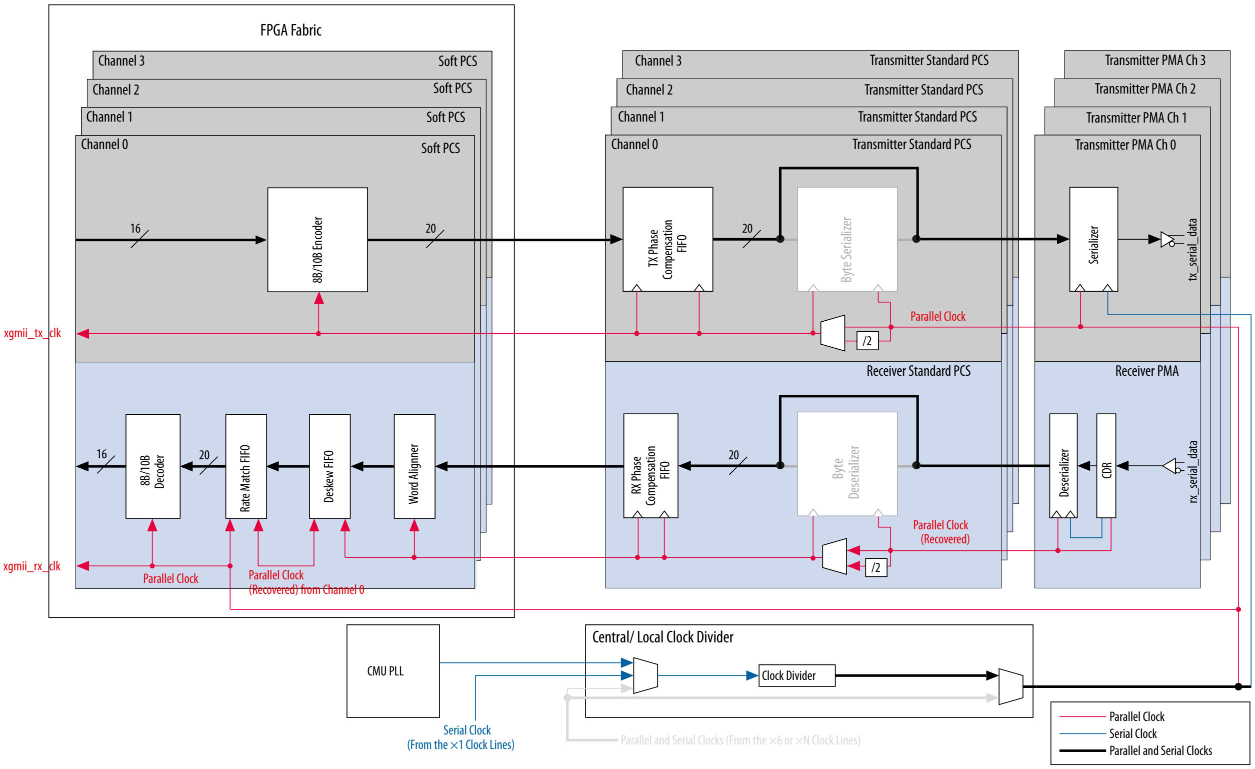


Table 6-8: Input Reference Clock Frequency and Interface Speed Specifications for XAUI Configurations

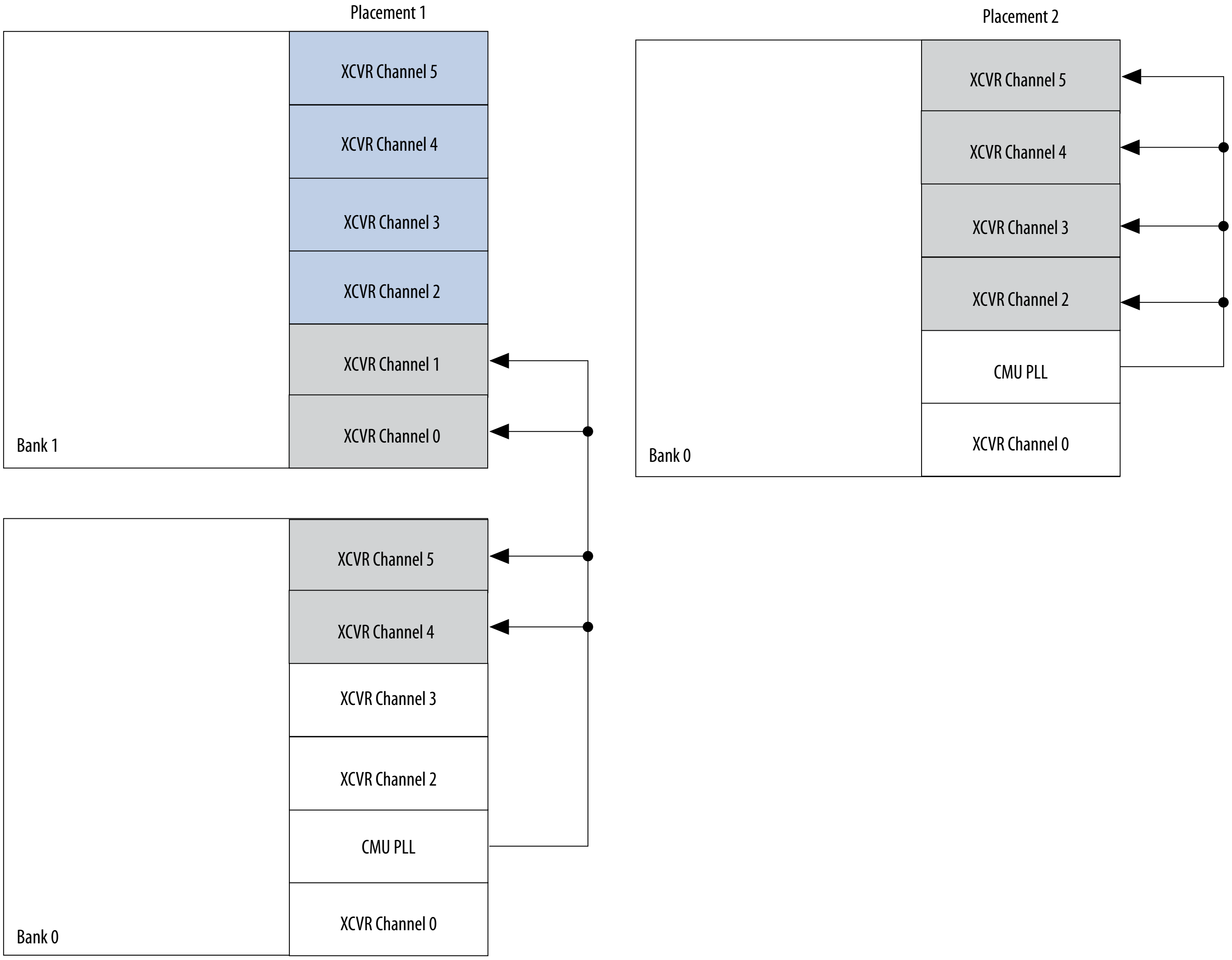
Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
156.25	16-bit data, 2-bit control	156.25

Transceiver Channel Placement Guidelines

In the soft PCS implementation of the XAUI configuration, all four channels must be placed continuously. The channels may all be placed in one bank or they may span two banks.

Figure 6-47: Transceiver Channel Placement Guidelines in a XAUI Configuration

Use one of the two allowed channel placements when using either the CMU PLL or the ATX PLL to drive the XAUI link. The Quartus II software implements the XAUI PCS in soft logic.



Related Information

To implement the QSF assignment workaround using the Assignment Editor, refer to the "XAUI PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide.

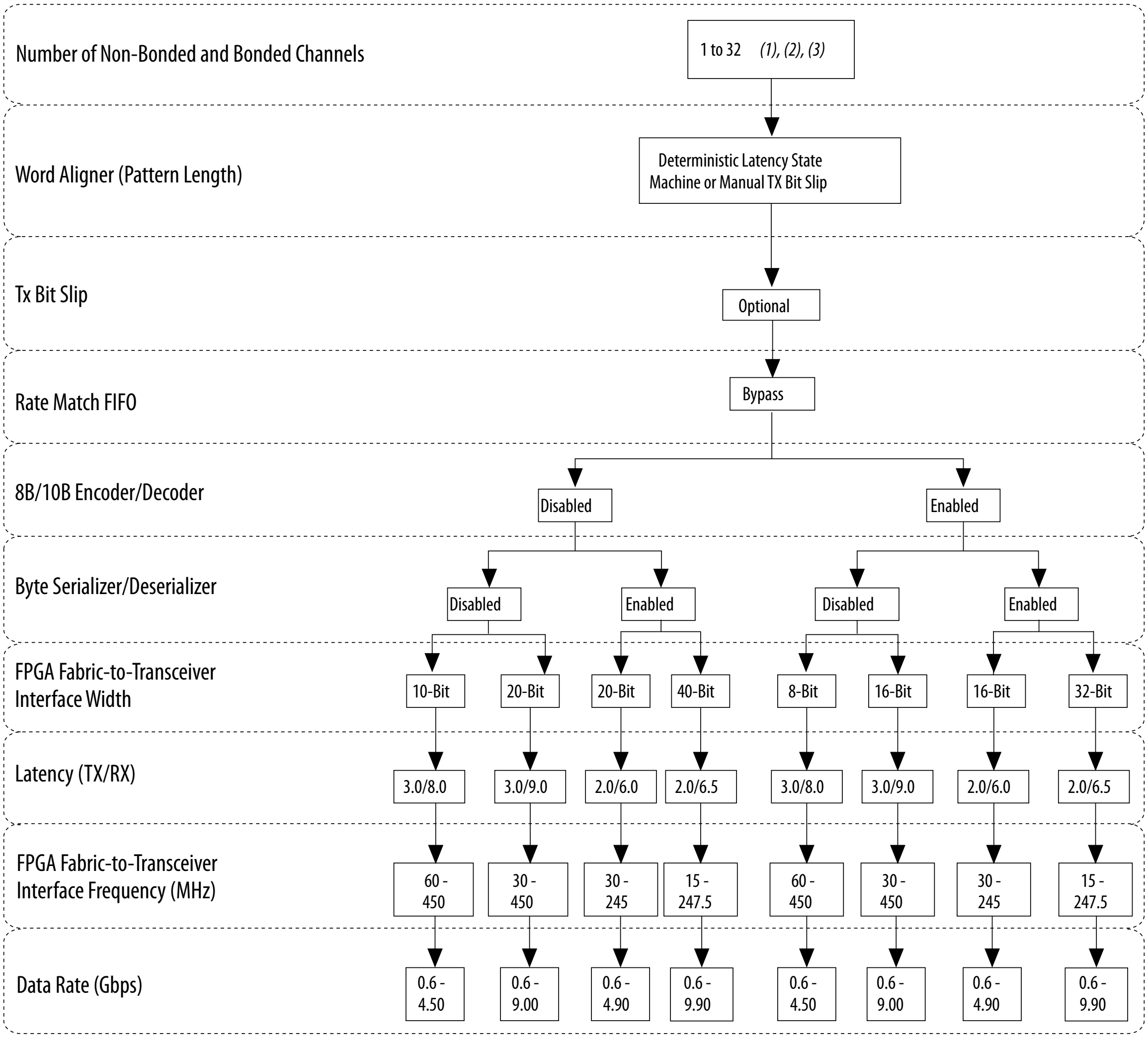
CPRI and OBSAI—Deterministic Latency Protocols

Arria V GZ devices have a deterministic latency option available for use in high-speed serial interfaces such as the Common Public Radio Interface (CPRI) and OBSAI Reference Point 3 (OBSAI RP3). Both CPRI and OBSAI RP3 protocols place stringent requirements on the amount of latency variation that is permissible through a link that implements these protocols.

Transceiver Datapath Configuration

Arria V GZ devices have a number of options available for the deterministic latency datapath configuration.

Figure 6-48: Deterministic Latency Datapath Configuration



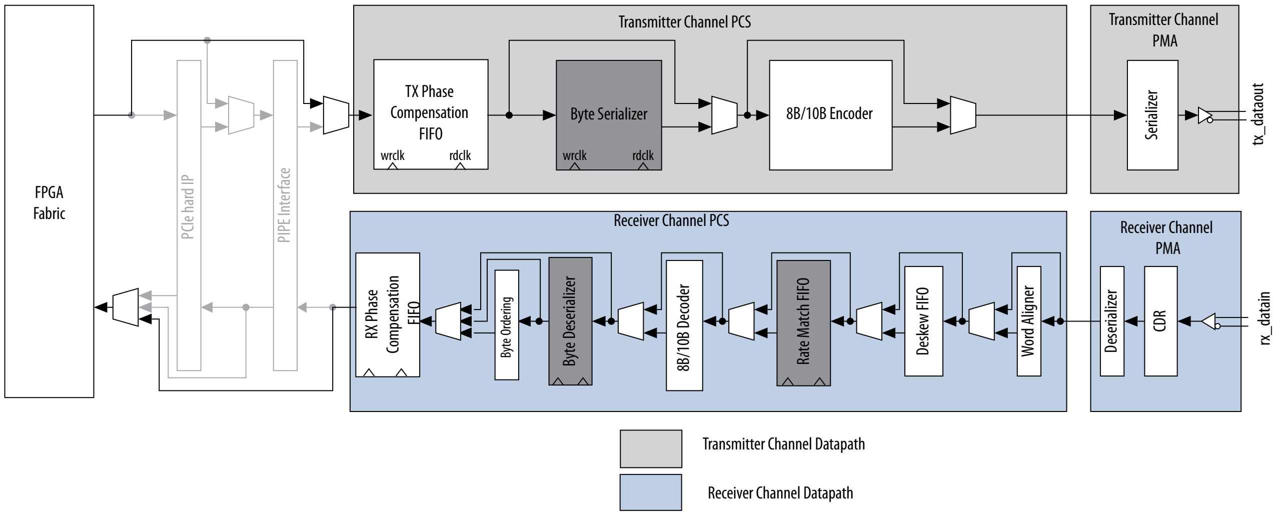
Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) The TX-client feedback path to the transmit PLL is only supported in a non-bonded single lane instance.

Figure 6-49: Transceiver Datapath in Deterministic Latency Mode



Phase Compensation FIFO in Register Mode

To remove the latency uncertainty through the receiver's phase compensation FIFO, the receiver and transmitter phase compensation FIFOs are always set to register mode. In register mode, the phase compensation FIFO acts as a register and thereby removes the uncertainty in latency. The latency through the phase compensation FIFO in register mode is one clock cycle.

The following options are available:

- Single-width mode with 8-bit channel width and 8B/10B encoder enabled or 10-bit channel width with 8B/10B disabled
- Double-width mode with 16-bit channel width and 8B/10B encoder enabled or 20-bit channel width with 8B/10B disabled

Channel PLL Feedback

To implement the deterministic latency functional mode, the phase relationship between the low-speed parallel clock and channel PLL input reference clock must be deterministic. A feedback path is enabled to ensure a deterministic relationship between the low-speed parallel clock and channel PLL input reference clock.

To achieve deterministic latency through the transceiver, the reference clock to the channel PLL must be the same as the low-speed parallel clock. For example, if you need to implement a data rate of 1.2288 Gbps for the CPRI protocol, which places stringent requirements on the amount of latency variation, you must choose a reference clock of 122.88 MHz to allow the usage of a feedback path from the channel PLL. This feedback path reduces the variations in latency.

When you select this option, provide an input reference clock to the channel PLL that is of the same frequency as the low-speed parallel clock.

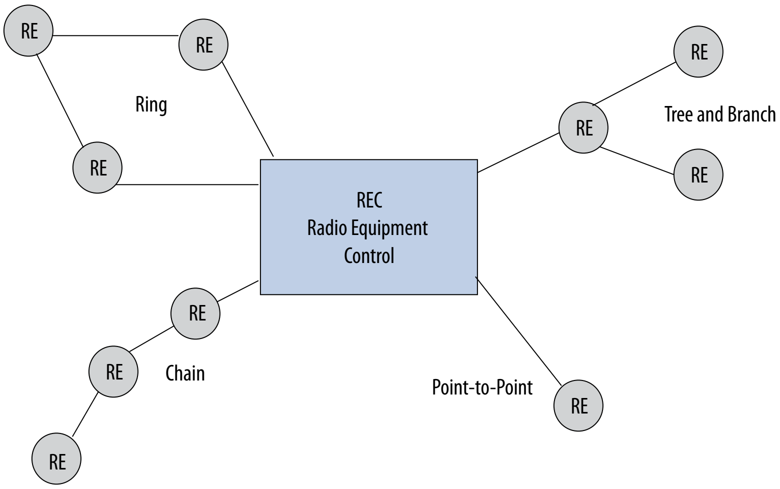
CPRI and OBSAI

Use the deterministic latency functional mode to implement protocols such as CPRI and OBSAI.

The CPRI interface defines a digital point-to-point interface between the Radio Equipment Control (REC) and the Radio Equipment (RE), allowing flexibility in either co-locating the REC and the RE, or a remote location of the RE.

Figure 6-50: CPRI Topologies

In most cases, CPRI links are between REC and RE modules or between two RE modules in a chain configuration.



If the destination for the high-speed serial data that leaves the REC is the first RE, it is a single-hop connection. If the serial data from the REC must traverse through multiple REs before reaching the destination RE, it is a multi-hop connection.

Remotely locating the RF transceiver from the main base station introduces a complexity with overall system delay. The CPRI specification requires that the accuracy of measurement of roundtrip delay on single-hop and multi-hop connections be within ±16.276 ns to properly estimate the cable delay.

For a single-hop system, this allows a variation in roundtrip delay of up to ±16.276 ns. However, for multi-hop systems, the allowed delay variation is divided among the number of hops in the connection—typically, equal to ±16.276 ns/(the number of hops) but not always equally divided among the hops.

Deterministic latency on a CPRI link also enables highly accurate triangulation of the location of the caller.

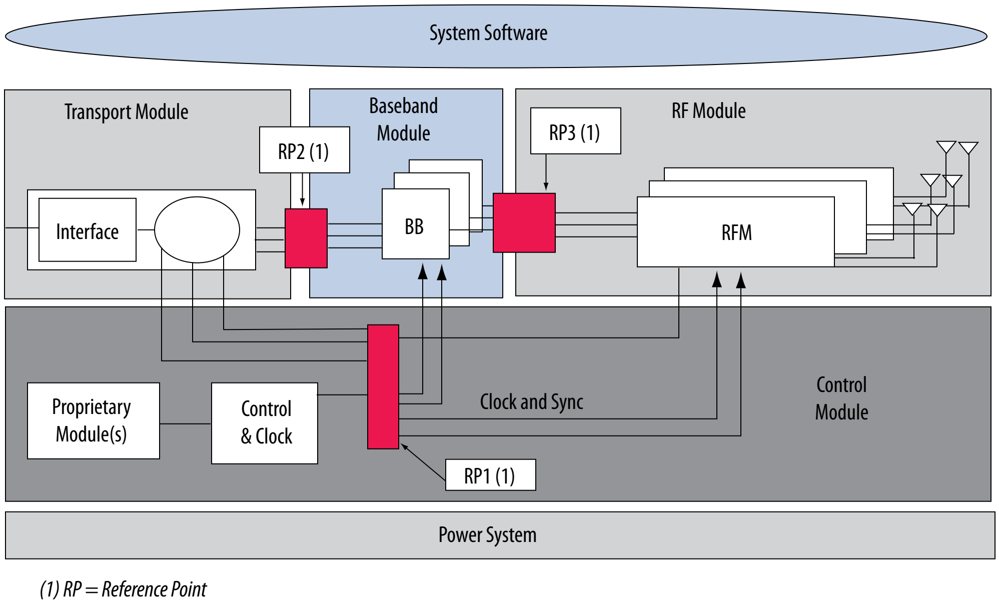
OBSAI was established by several OEMs to develop a set of specifications that can be used for configuring and connecting common modules into base transceiver stations (BTS).

The BTS has four main modules:

- Radio frequency (RF)
- Baseband
- Control
- Transport

In a typical BTS, the radio frequency module (RFM) receives signals using portable devices and converts the signals to digital data. The baseband module processes the encoded signal and brings it back to the baseband before transmitting it to the terrestrial network using the transport module. A control module maintains the coordination between these three functions.

Figure 6-51: Example of the OBSAI BTS Architecture



Using the deterministic latency option, you can implement the CPRI data rates in the following modes:

- Single-width mode—with 8/10-bit channel width
- Double-width mode—with 16/20-bit channel width

Table 6-9: Sample Channel Width Options for Supported Serial Data Rates

Serial Data Rate (Mbps)	Channel Width (FPGA-PCS Fabric)			
	Single Width		Double-Width	
	8-Bit	16-Bit	16-Bit	32-Bit
614.4	Yes	Yes	—	—
1228.8	Yes	Yes	Yes	Yes
2457.6	—	Yes	Yes	Yes
3072	—	Yes	Yes	Yes
4915.2	—	—	—	Yes
6144	—	—	—	Yes
9800 [39][40]	—	—	—	Yes

[39] The Arria V GZ Standard PCS can support up to 9.9 Gbps datarates in Deterministic Latency configuration or up to 9.8 Gbps in Custom and Low Latency configurations.

[40] Applicable to C3 and I3L speed grades only.

Related Information

For more information, refer to the Deterministic Latency PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide.

Transceiver Configurations

Arria V GZ transceivers offer both standard PCS and 10G PCS configurations. These configurations allow you to modify, enable, or disable blocks based on your protocol requirements. This flexibility allows you to implement various protocols through the Custom, Low Latency, and Native PHY IPs.

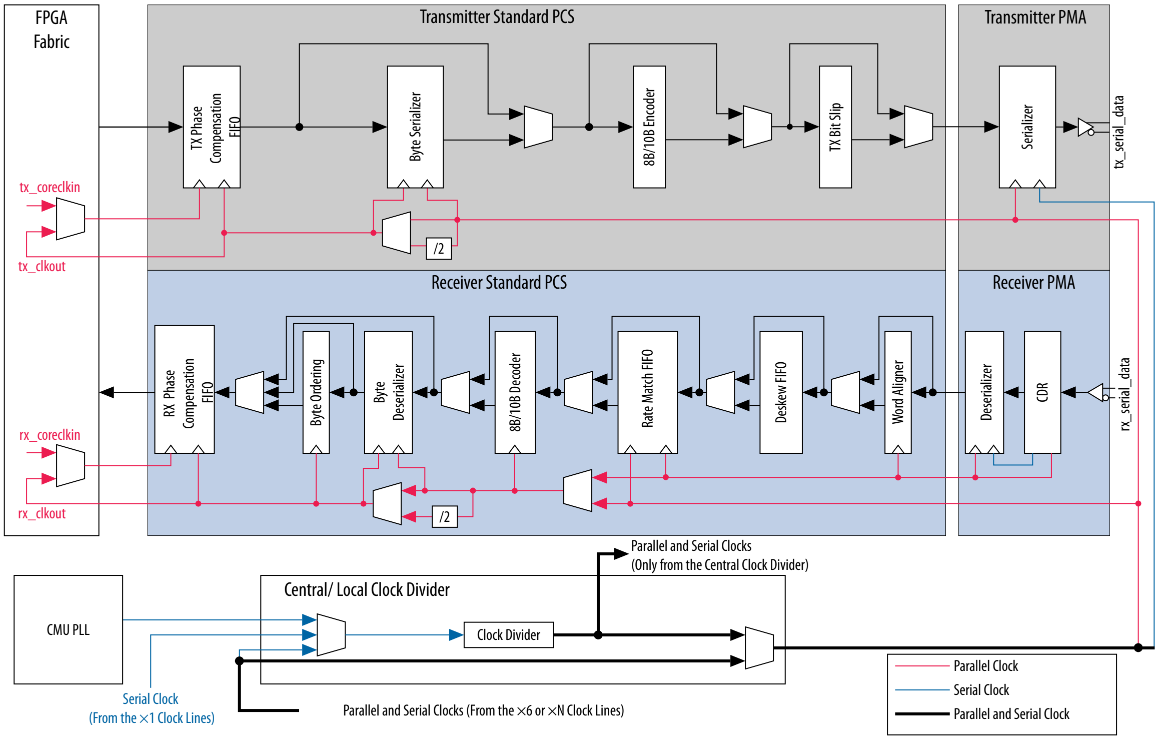
Standard PCS Configurations—Custom Datapath

Use the Custom PHY IP to enable the standard PCS in custom datapath. To implement a Custom PHY link, instantiate the **Custom PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. Define your custom datapath configurations by selecting the blocks to use and the appropriate data width.

The custom datapath consists of the following blocks:

- 8B/10B encoder and decoder
- Word aligner
- Deskew FIFO
- Rate match FIFO (clock rate compensation FIFO)
- Byte ordering block
- Phase compensation FIFO
- Byte serializer and deserializer
- Transmit bit slip

Figure 6-52: Standard PCS Custom Datapath and Clocking



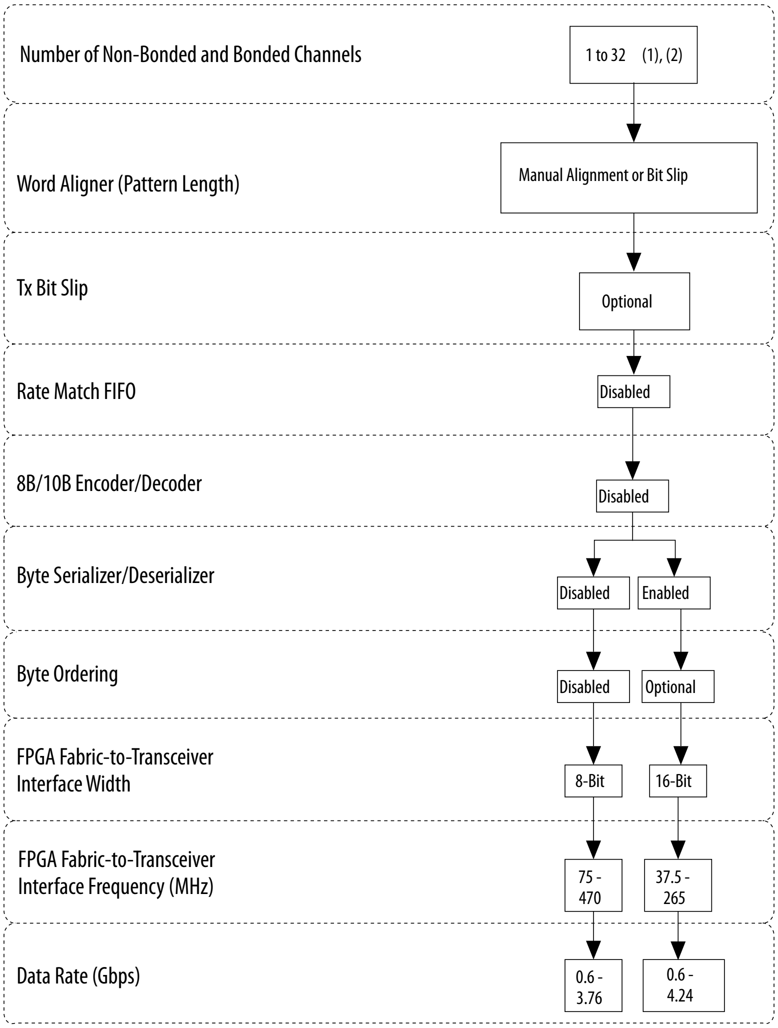
You can divide the custom datapath into two configurations based on the FPGA fabric-transceiver interface width and the PMA-PCS interface width (serialization factor):

- **Custom 8/10-bit-width**—the PCS-PMA interface width is in 8-bit or 10-bit mode for lower data rates.
- **Custom 16/20-bit-width**—the PCS-PMA interface width is in 16-bit or 20-bit mode for higher data rates.

Table 6-10: PCS-PMA Interface Widths and Supported Data Rates

PCS-PMA Interface Width	Supported Data Rate Range PMA
Custom 8-bit width	600 Mbps to 4.24 Gbps
Custom 10-bit width	600 Mbps to 5.30 Gbps
Custom 16-bit width	600 Mbps to 7.84 Gbps
Custom 20-bit width	600 Mbps to 9.80 Gbps

Figure 6-53: Standard PCS Custom 8-Bit PMA-PCS Interface Width

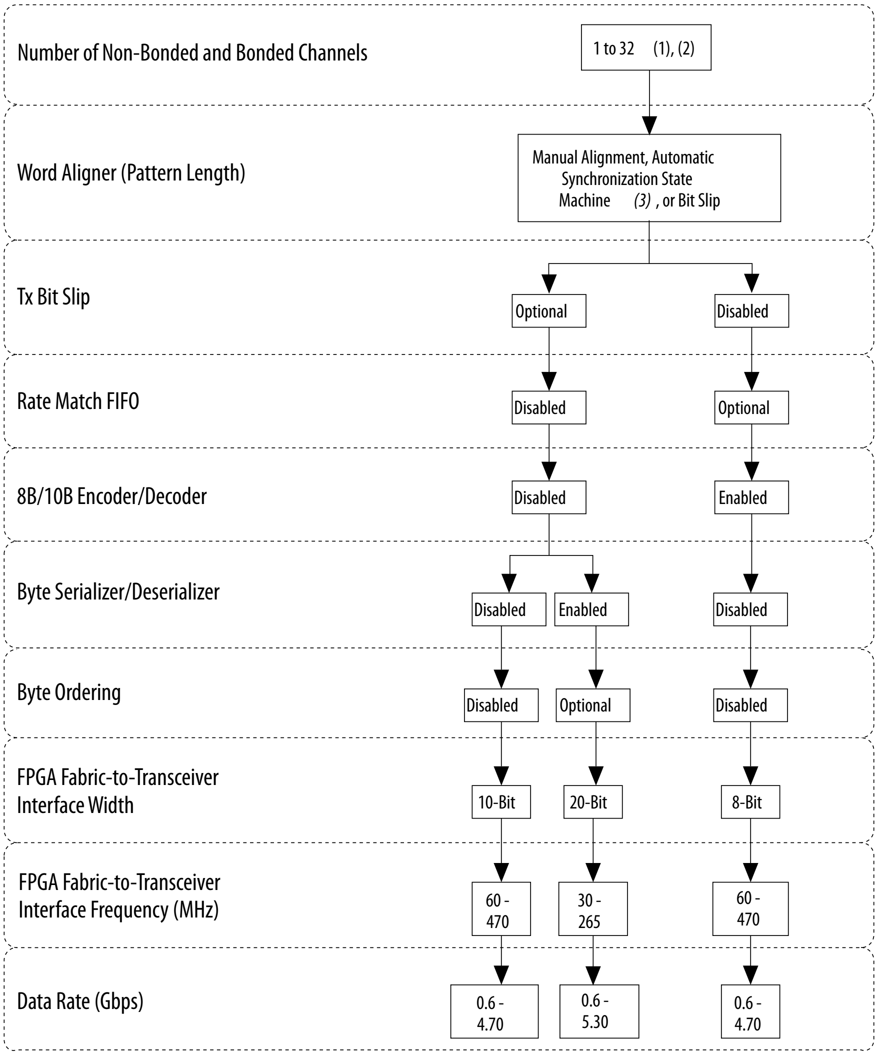

Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Manual Alignment or Bit Slip
Tx Bit Slip
Optional
Rate Match FIFO
Disabled
8B/10B Encoder/Decoder
Disabled
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Disabled
Optional
FPGA Fabric-to-Transceiver Interface Width
8-Bit
16-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
75 - 470
37.5 - 265
Data Rate (Gbps)
0.6 - 3.76
0.6 - 4.24

Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 6-54: Standard PCS Custom 10-Bit PMA-PCS Interface Width


Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Manual Alignment, Automatic Synchronization State Machine (3) , or Bit Slip
Tx Bit Slip
Optional
Disabled
Rate Match FIFO
Disabled
Optional
8B/10B Encoder/Decoder
Disabled
Enabled
Byte Serializer/Deserializer
Disabled
Enabled
Disabled
Byte Ordering
Disabled
Optional
Disabled
FPGA Fabric-to-Transceiver Interface Width
10-Bit
20-Bit
8-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
60 - 470
30 - 265
60 - 470
Data Rate (Gbps)
0.6 - 4.70
0.6 - 5.30
0.6 - 4.70

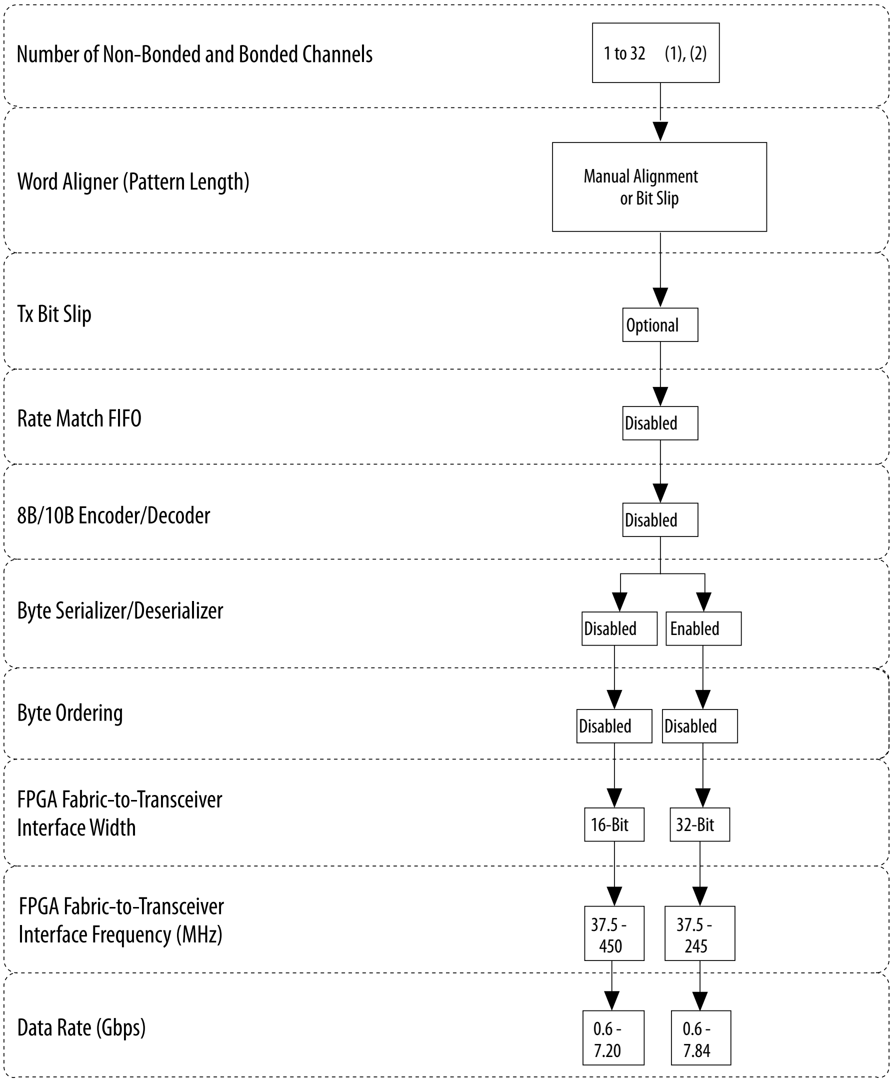
Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) Automatic Synchronization State Machine requires enabling the 8B/10B Encoder/Decoder.

Figure 6-55: Standard PCS Custom 16-Bit PMA-PCS Interface Width

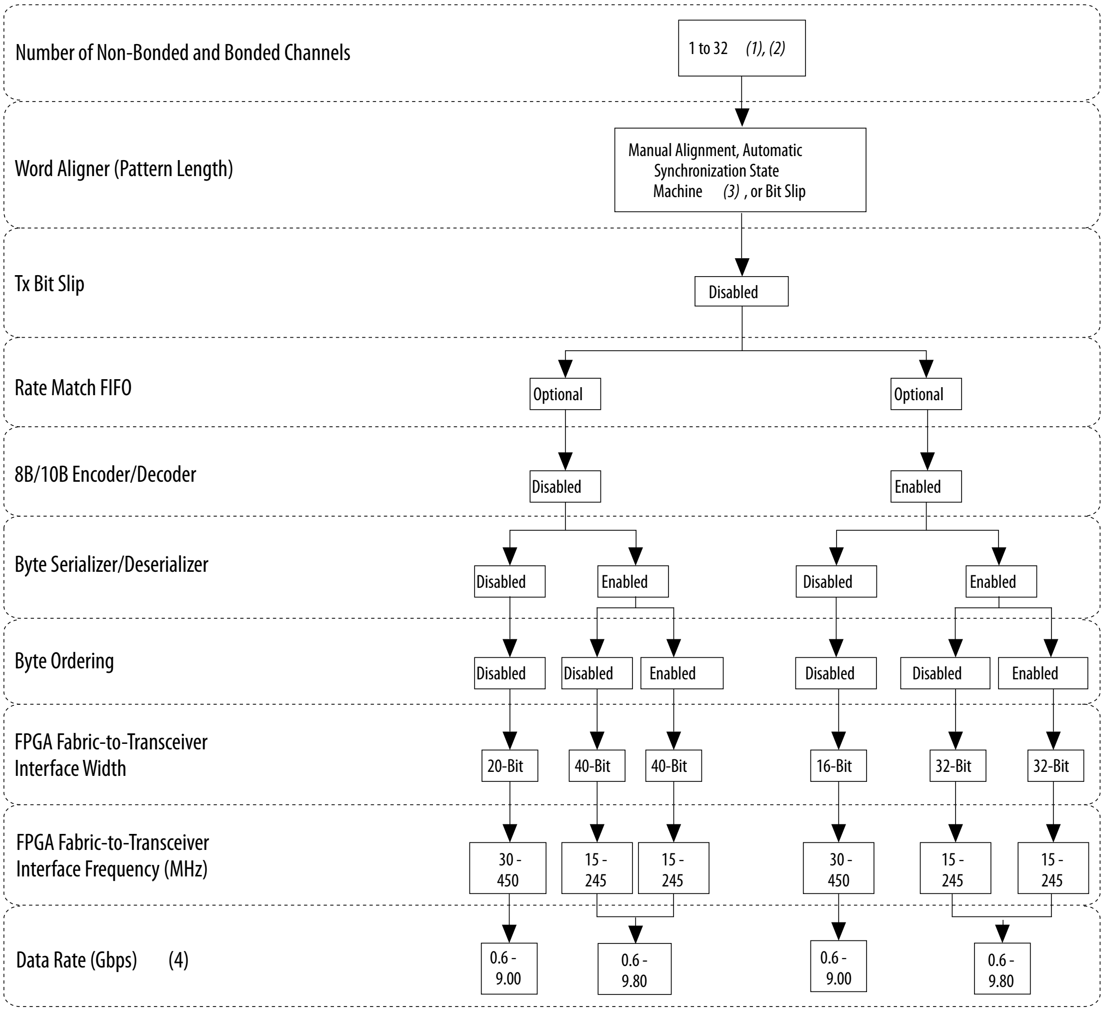

Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Manual Alignment or Bit Slip
Tx Bit Slip
Optional
Rate Match FIFO
Disabled
8B/10B Encoder/Decoder
Disabled
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Disabled
Disabled
FPGA Fabric-to-Transceiver Interface Width
16-Bit
32-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
37.5 - 450
37.5 - 245
Data Rate (Gbps)
0.6 - 7.20
0.6 - 7.84

Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 6-56: Standard PCS Custom 20-Bit PMA-PCS Interface Width



Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) Automatic Synchronization State Machine requires enabling the 8B/10B Encoder/Decoder.

(4) The maximum data rate specification is valid only for the -2 (fastest) speed grade devices. For data rate specifications for other speed grades, refer to the device datasheet for that device.

Related Information

- Refer to the “PCS Architecture” section in the Transceiver Architecture in Arria V Devices chapter
- For information about the maximum data rate for a certain speed grade, refer to the Arria V Device Datasheet
- Refer to the "Custom PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide

Standard PCS Configurations—Low Latency Datapath

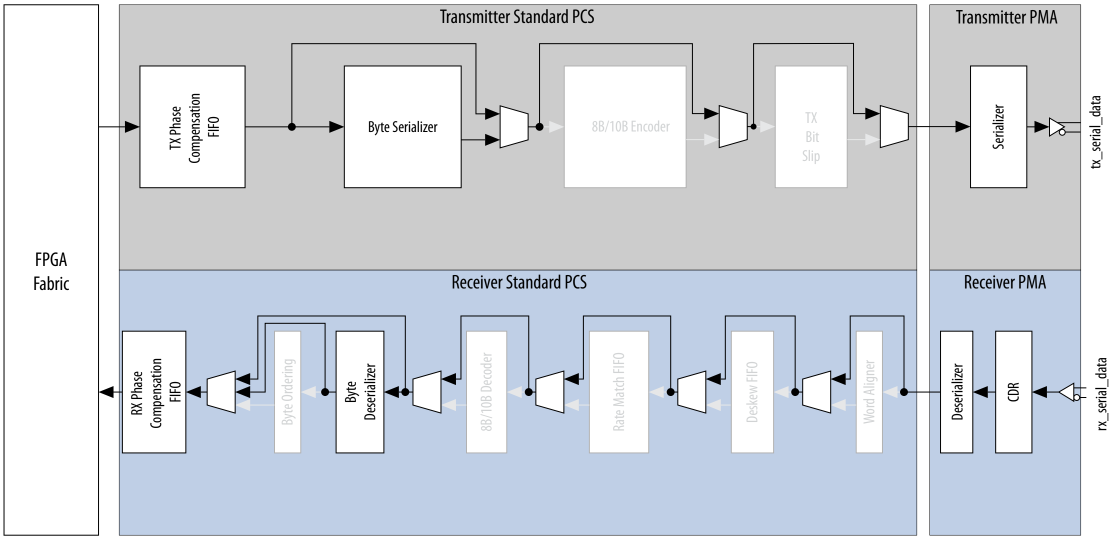
A low latency datapath bypasses much of the standard PCS, allowing more design control in the FPGA fabric. Use the Low Latency PHY IP to enable the standard PCS in a low latency datapath.

To implement a Low Latency PHY link, instantiate the **Low Latency PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. In the Low Latency GUI under the **General** tab, select **Standard** on the **Datapath type** field.

The standard PCS can be used in a low latency datapath that contains only the following blocks:

- Phase compensation FIFO
- Byte serializer and deserializer

Figure 6-57: Standard PCS Low Latency Datapath



You can divide the low latency datapath into two configurations based on the FPGA fabric-transceiver interface width and the PMA-PCS interface width (serialization factor):

- **Low latency 8/10-bit-width**—the PCS-PMA interface width is in 8-bit or 10-bit mode for lower data rates.
- **Low latency 16/20-bit-width**—the PCS-PMA interface width is in 16-bit or 20-bit mode for higher data rates.

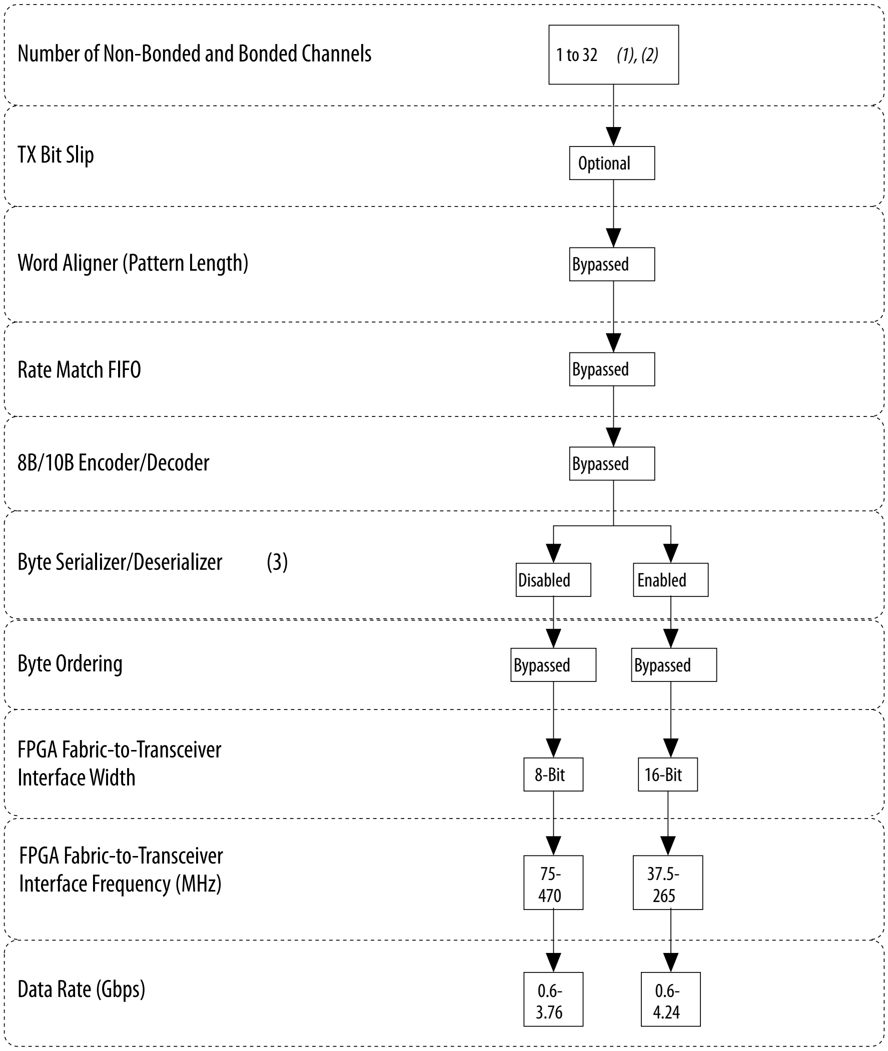
Table 6-11: PCS-PMA Interface Widths and Data Rates

Low Latency PHY IP Core	Supported Data Rate Range PMA
Low Latency 8-bit width	600 Mbps to 4.24 Gbps
Low Latency 10-bit width	600 Mbps to 5.30 Gbps
Low Latency 16-bit width	600 Mbps to 7.84 Gbps
Low Latency 20-bit width	600 Mbps to 9.80 Gbps

In the low latency datapath, the TX and RX phase compensation FIFOs are always enabled. Depending on the targeted data rate, you may bypass the byte serializer and deserializer blocks.

Figure 6-58: Standard PCS Low Latency 8-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS low latency 8-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies are for the fastest devices.



Notes:

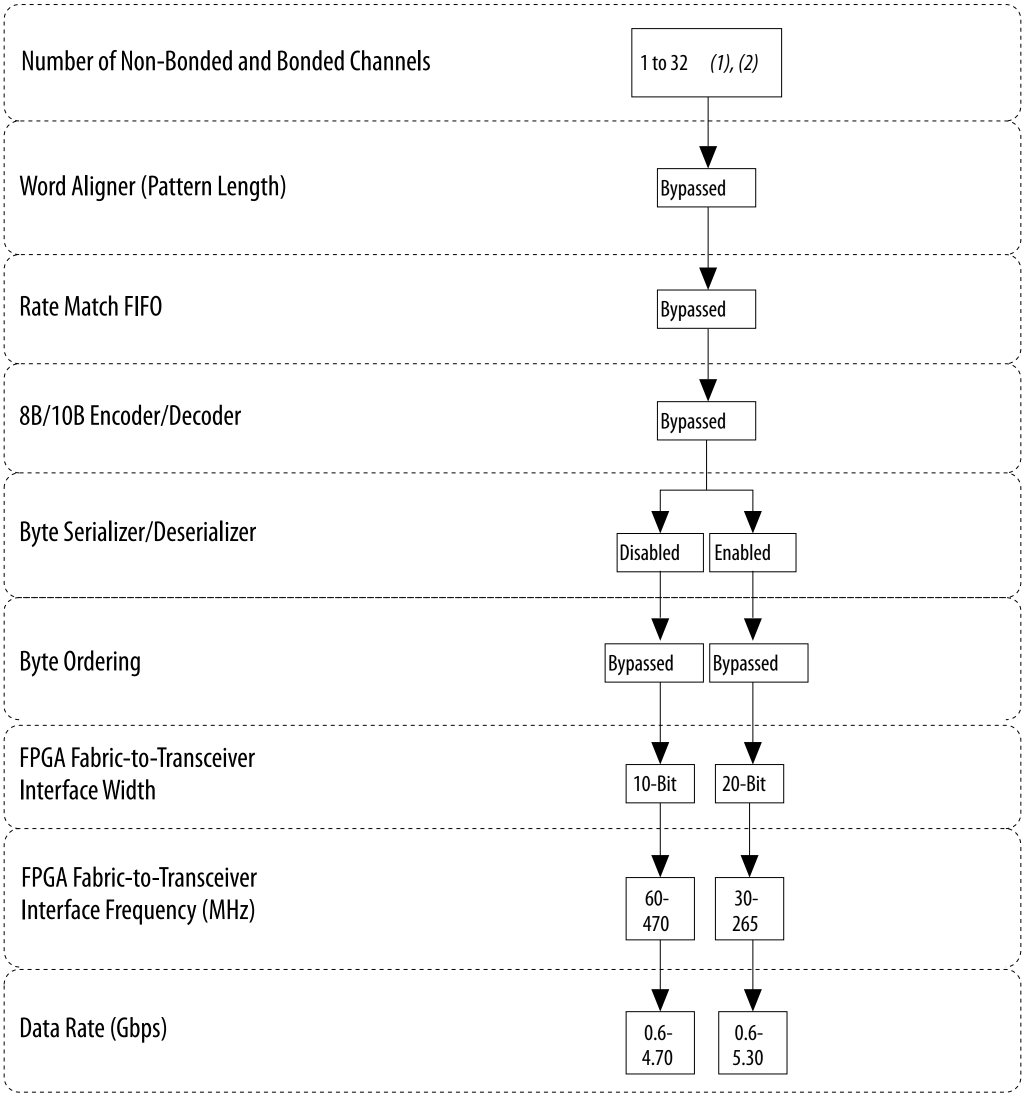
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) The Quartus II software selects whether the byte serializer/deserializer is enabled or disabled based on the datapath width.

Figure 6-59: Standard PCS Low Latency 10-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS low latency 10-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies are for the fastest devices.


Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Bypassed
Rate Match FIFO
Bypassed
8B/10B Encoder/Decoder
Bypassed
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Bypassed
Bypassed
FPGA Fabric-to-Transceiver Interface Width
10-Bit
20-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
60-470
30-265
Data Rate (Gbps)
0.6-4.70
0.6-5.30

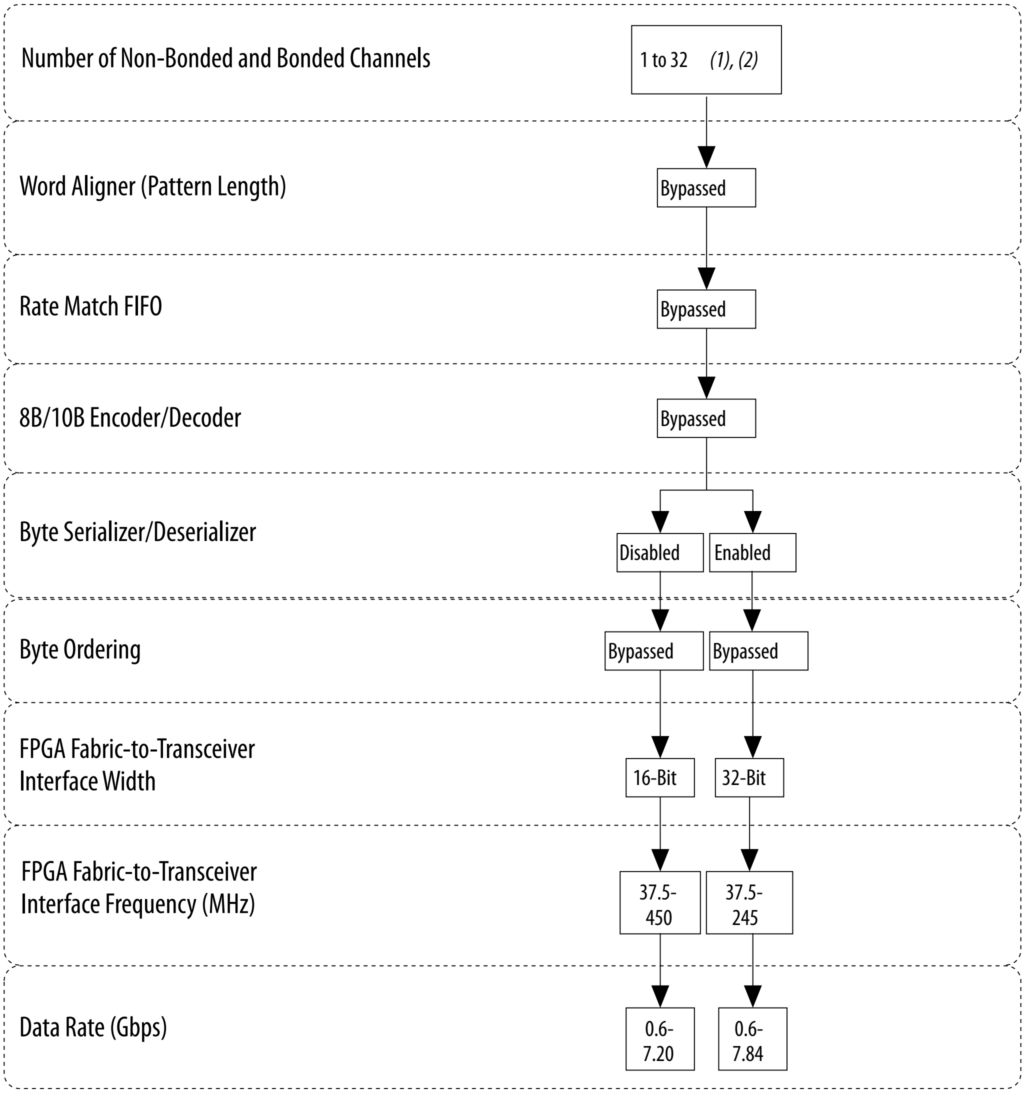
Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 6-60: Standard PCS Low Latency 16-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS low latency 16-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies are for the fastest devices.


Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Bypassed
Rate Match FIFO
Bypassed
8B/10B Encoder/Decoder
Bypassed
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Bypassed
Bypassed
FPGA Fabric-to-Transceiver Interface Width
16-Bit
32-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
37.5-450
37.5-245
Data Rate (Gbps)
0.6-7.20
0.6-7.84

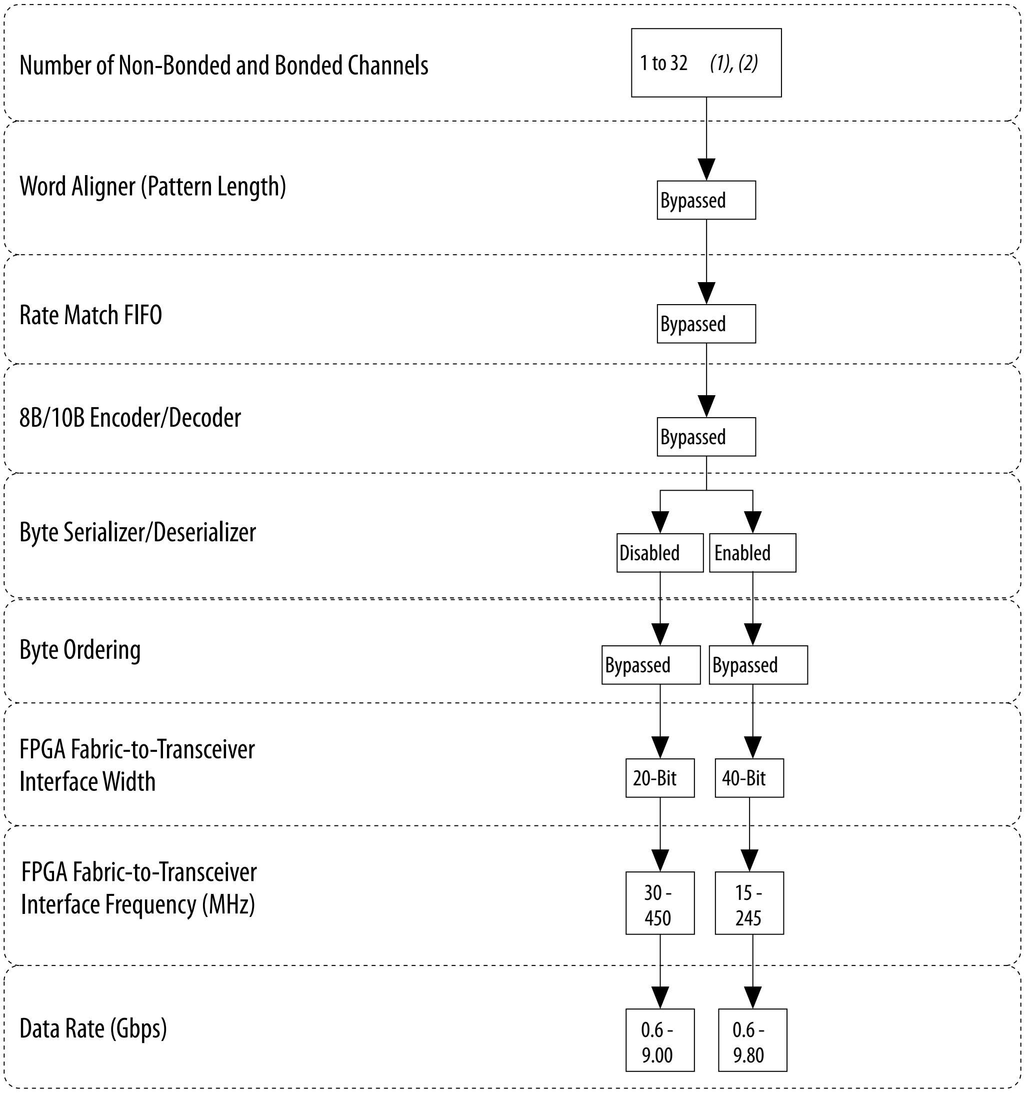
Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 6-61: Standard PCS Low Latency 20-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS low latency 20-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies are for the fastest devices.


Number of Non-Bonded and Bonded Channels
1 to 32 (1), (2)
Word Aligner (Pattern Length)
Bypassed
Rate Match FIFO
Bypassed
8B/10B Encoder/Decoder
Bypassed
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Bypassed
Bypassed
FPGA Fabric-to-Transceiver Interface Width
20-Bit
40-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
30 - 450
15 - 245
Data Rate (Gbps)
0.6 - 9.00
0.6 - 9.80

Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Related Information

- Refer to the "PCS Architecture" section in the Transceiver Architecture in Arria V Devices chapter
- For information about the maximum data rate for a certain speed grade, refer to the Arria V Device Datasheet

- **Refer to the "Low Latency PHY IP Core" chapter in the Altera Transceiver PHY IP Core User Guide**

Transceiver Channel Placement Guidelines

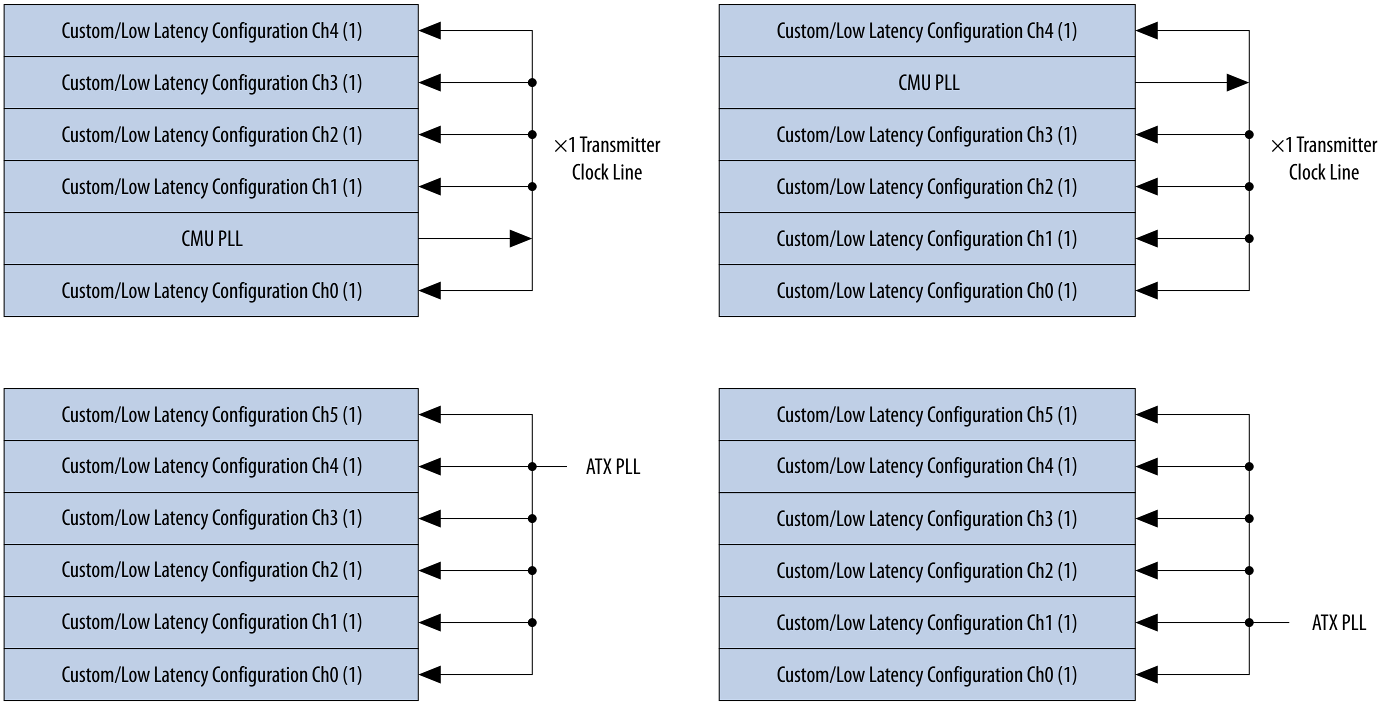
You can use CMU PLLs or ATX PLLs in non-bonded and bonded configurations.

Arria V GZ devices allow the placement of up to five channels when a CMU PLL is used or up to six channels when an ATX PLL is used in a non-bonded configuration within the same transceiver bank:

- Custom PHY IP with standard PCS datapath configuration
- Low Latency PHY IP with Standard PCS or 10G PCS (same data rate) in low latency datapath configuration

Figure 6-62: Non-Bonded Channel Placement Guidelines with Standard and 10G PCS in Custom and Low Latency Datapath Configurations

All channels are assumed to contain a transmitter and receiver.

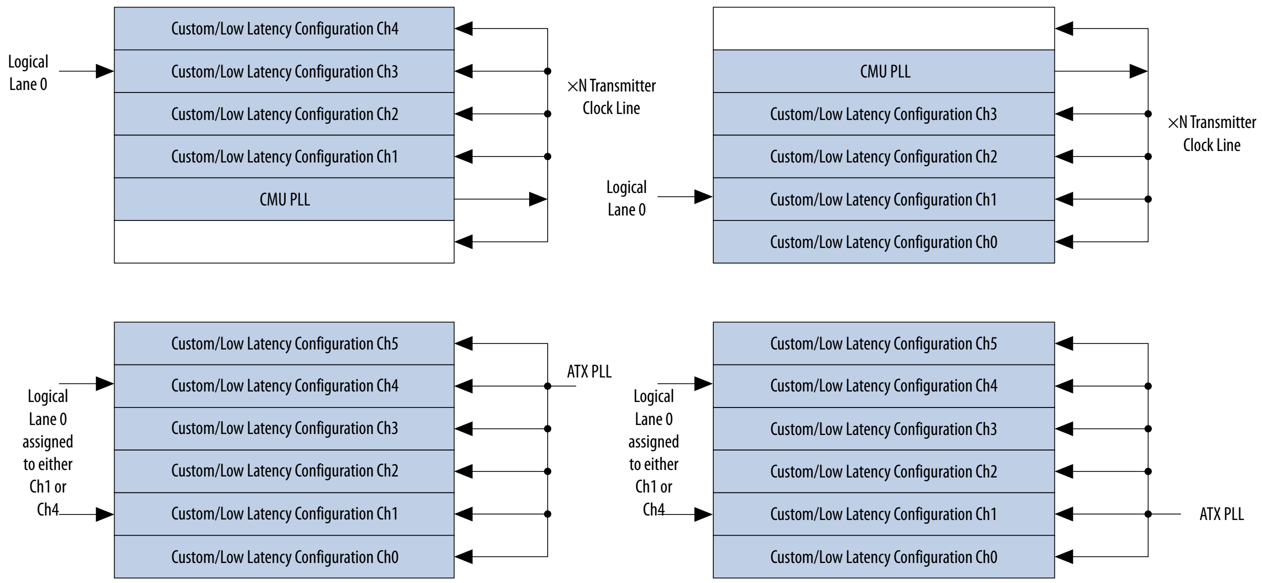

Custom/Low Latency Configuration Ch4 (1)
Custom/Low Latency Configuration Ch3 (1)
Custom/Low Latency Configuration Ch2 (1)
Custom/Low Latency Configuration Ch1 (1)
CMU PLL
Custom/Low Latency Configuration Ch0 (1)
×1 Transmitter Clock Line
Custom/Low Latency Configuration Ch4 (1)
CMU PLL
Custom/Low Latency Configuration Ch3 (1)
Custom/Low Latency Configuration Ch2 (1)
Custom/Low Latency Configuration Ch1 (1)
Custom/Low Latency Configuration Ch0 (1)
×1 Transmitter Clock Line
Custom/Low Latency Configuration Ch5 (1)
Custom/Low Latency Configuration Ch4 (1)
Custom/Low Latency Configuration Ch3 (1)
Custom/Low Latency Configuration Ch2 (1)
Custom/Low Latency Configuration Ch1 (1)
Custom/Low Latency Configuration Ch0 (1)
ATX PLL
Custom/Low Latency Configuration Ch5 (1)
Custom/Low Latency Configuration Ch4 (1)
Custom/Low Latency Configuration Ch3 (1)
Custom/Low Latency Configuration Ch2 (1)
Custom/Low Latency Configuration Ch1 (1)
Custom/Low Latency Configuration Ch0 (1)
ATX PLL

Arria V GZ devices allow the placement of up to four channels when a CMU PLL is used or up to six channels when an ATX PLL is used in a bonded configuration within the same transceiver bank:

- Custom PHY IP with standard PCS datapath configuration
- Low Latency PHY IP with Standard PCS or 10G PCS (same data rate) in low latency datapath configuration

The xN bonding method requires Logical Lane 0 be placed at either transceiver physical channel 1 or 4 within a transceiver bank. The PLL feedback compensation bonding method does not have a Logical Lane 0 assignment requirement and must be used when more than one transceiver bank is needed. However, PLL feedback compensation bonding requires the use of one PLL per transceiver bank.

Figure 6-63: Bonded Channel Placement Guidelines with Standard and 10G PCS in Custom and Low Latency Datapath Configurations


Logical Lane 0
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
CMU PLL
×N Transmitter Clock Line
CMU PLL
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Logical Lane 0
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
×N Transmitter Clock Line
Custom/Low Latency Configuration Ch5
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
Logical Lane 0 assigned to either Ch1 or Ch4
ATX PLL
Logical Lane 0 assigned to either Ch1 or Ch4
Custom/Low Latency Configuration Ch5
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
ATX PLL

10G PCS Configurations

The Low Latency PHY IP can also configure 10G PCS in the low latency datapath.

To implement a Low Latency PHY link with the 10G PCS, instantiate the **Low Latency PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. In the Low Latency GUI under the **General** tab, select **10G** on the **Datapath type** field.

A Low Latency PHY IP core with the 10G PCS is available for 32-bit, 40-bit, 50-bit, 64-bit, or 66-bit PCS data width configurations.

Figure 6-64: 10G PCS Low Latency Configuration Datapath

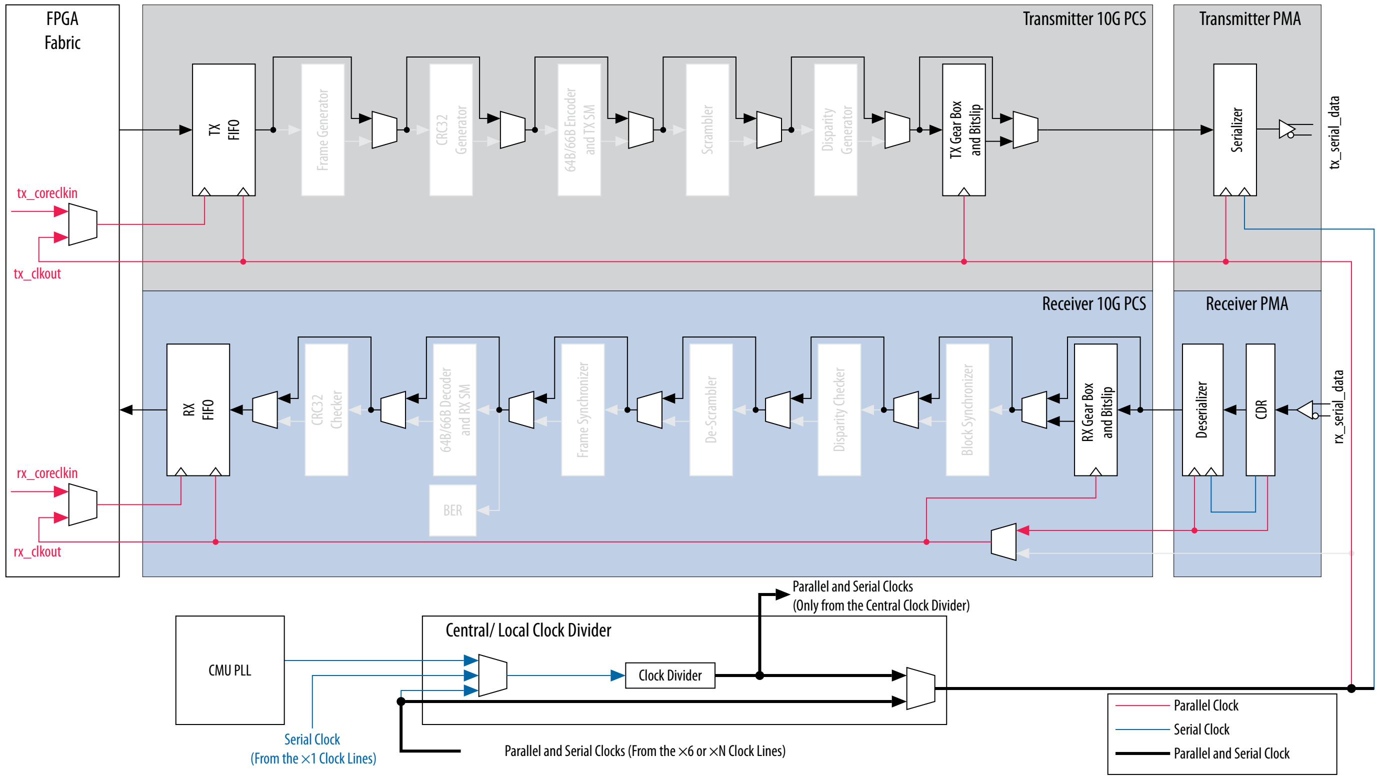
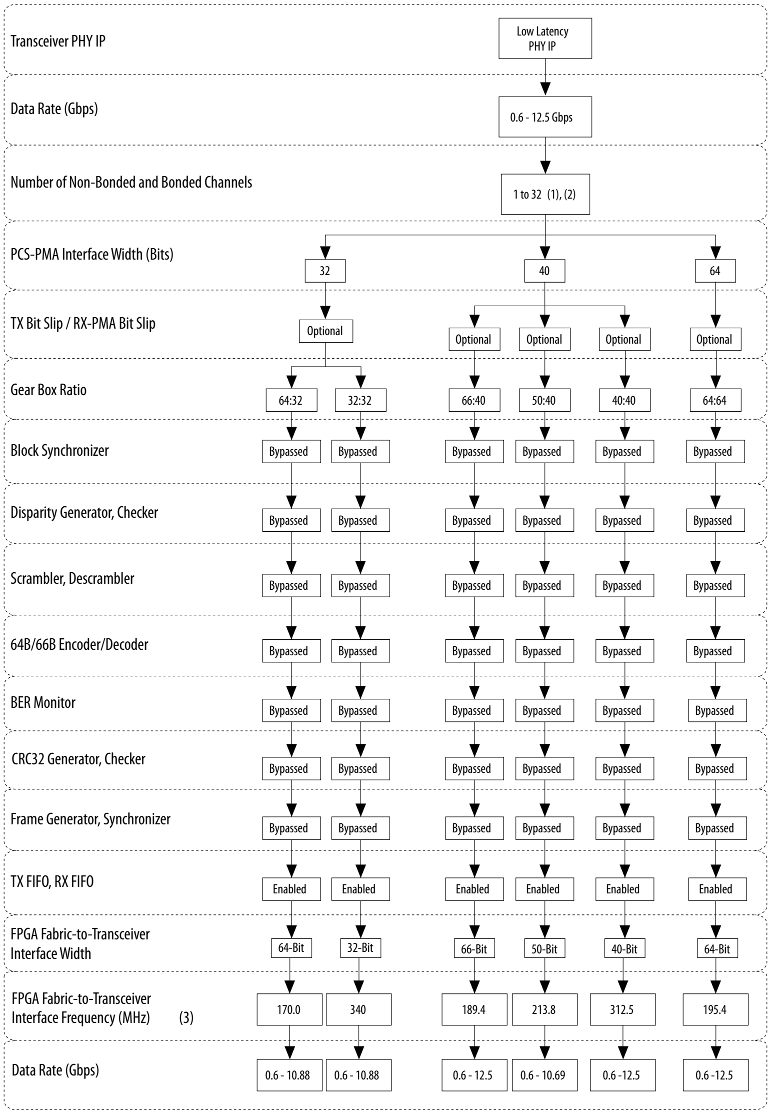


Figure 6-65: Options for 10G PCS Low Latency Configuration

The blocks shown as “Disabled” are not used but incur latency. The blocks shown as “Bypassed” are not used and do not incur any latency. The FPGA fabric-to-transceiver interface maximum frequency is for the fastest speed grade devices.



Notes:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) You must generate an rx_coreclkin with the specified frequency whenever the gear box is enabled.

The Quartus II software supports both non-bonded configuration and bonded configurations up to 32 lanes in the link when the 10G PCS in low latency datapath configuration is enabled. If you create multiple non-bonded channels with the 10G PCS in low latency mode, a common parallel clock (used in the bonded lane or channel configuration) is not generated by the central clock divider block. Each transmitter channel takes the high-speed clock, generated by the channel PLL, and locally divides it to generate the parallel clock.

Related Information

- For the limits of all speed grades, refer to the "Transceiver Performance Specifications" section in the Arria V Device Datasheet
- Transceiver Clocking in Arria V Devices
- Refer to the Low Latency PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide

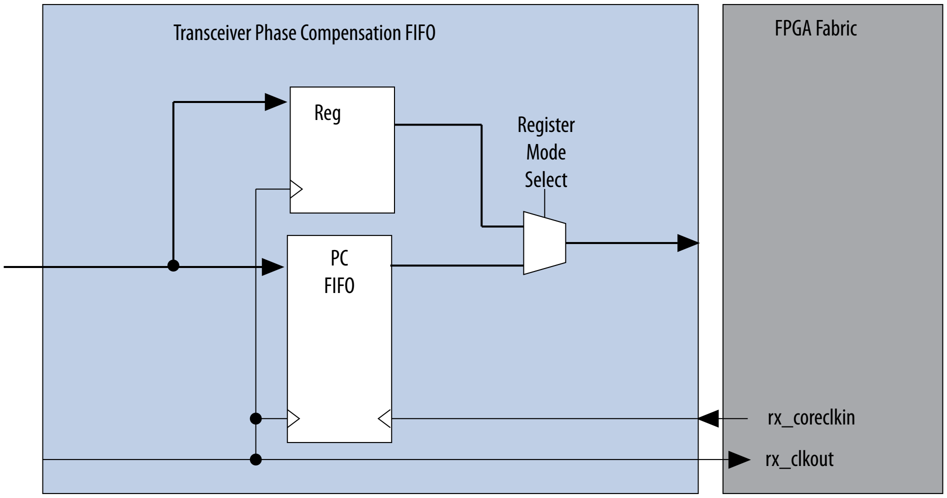
10G PCS Datapath Functionality

Various 10G PCS blocks are available when you implement the 10G PCS in low latency mode.

Transmitter and Receiver FIFO

The FIFOs can be configured in phase compensation or registered mode for the RX path. In phase compensation mode, the FIFO compensates the phase differences in the clock between the read and write side of the FIFO. The clocking scheme for the write side of the transmitter (TX) and receiver (RX) FIFOs depends on whether the gear box is enabled and on its ratio (40:66, 40:50, or 32:64). The clocking scheme is described in Clocking on page 1-82.

Figure 6-66: Phase Compensation FIFO in RX Path



Gear Box

The gear box translates the datapath width differences between the PCS and the physical medium attachment (PMA) interfaces. The gear box contains handshake control logic and FIFOs to implement the data-width translation. For the supported gear box ratio, refer to figure "Options for 10G PCS Low Latency Configuration".

TX Bit Slip Feature

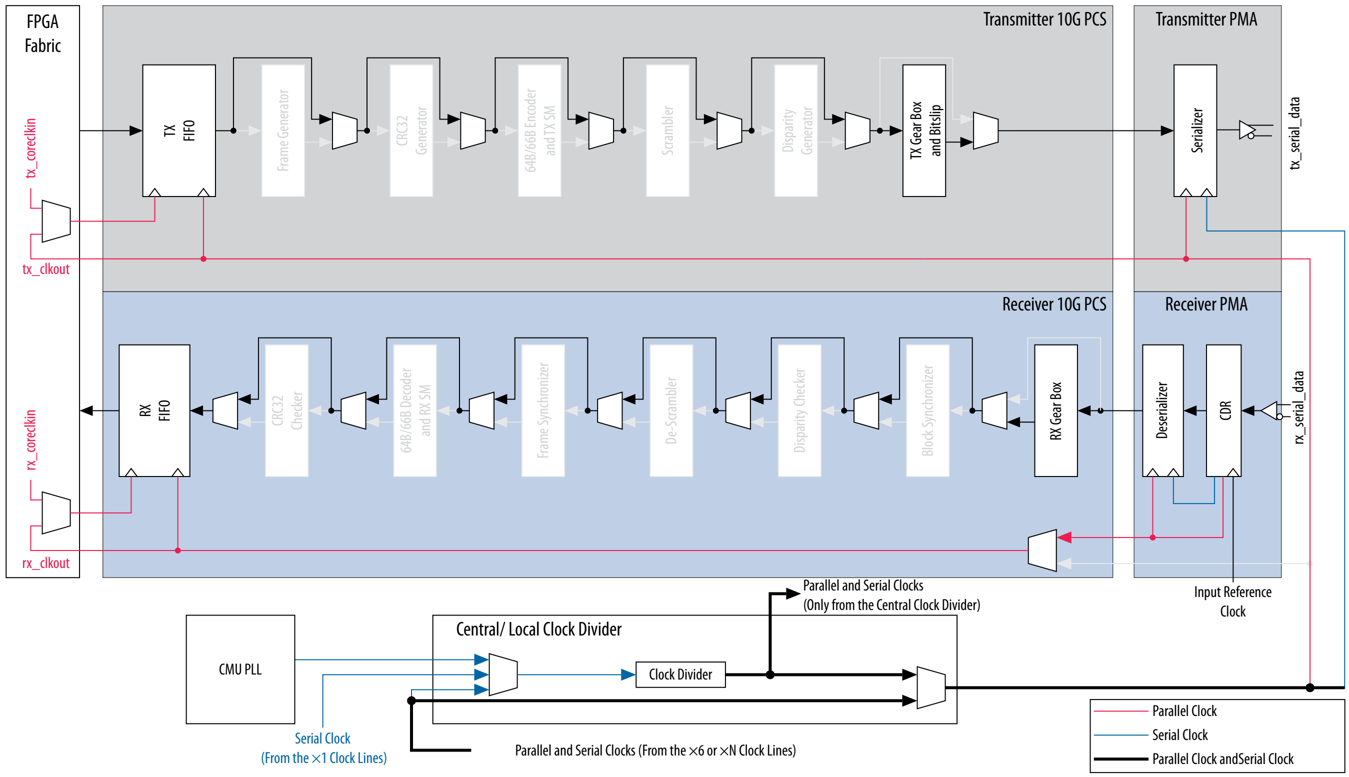
The bit slip feature allows you to slip the transmitter side bits before they are sent to the gear box. The number of bits slipped is equal to the FPGA fabric-to-transceiver interface width minus 1. For example, if the FPGA fabric-to-transceiver interface width is 64 bits, a maximum of 63 bits can be slipped. That is, `bit[63]` from the first word and `bit[62:0]` are concatenated to form a 64 bit word (`bit[62:0]` from the second word, `bit[63]` from the first word LSB). The 7-bit input control signal is available to the FPGA fabric. For a 63-bit shift mentioned above, set the value of the input control to 7'b0011111.

Clocking

The transceiver datapath clocking scheme depends on the gear box ratio.

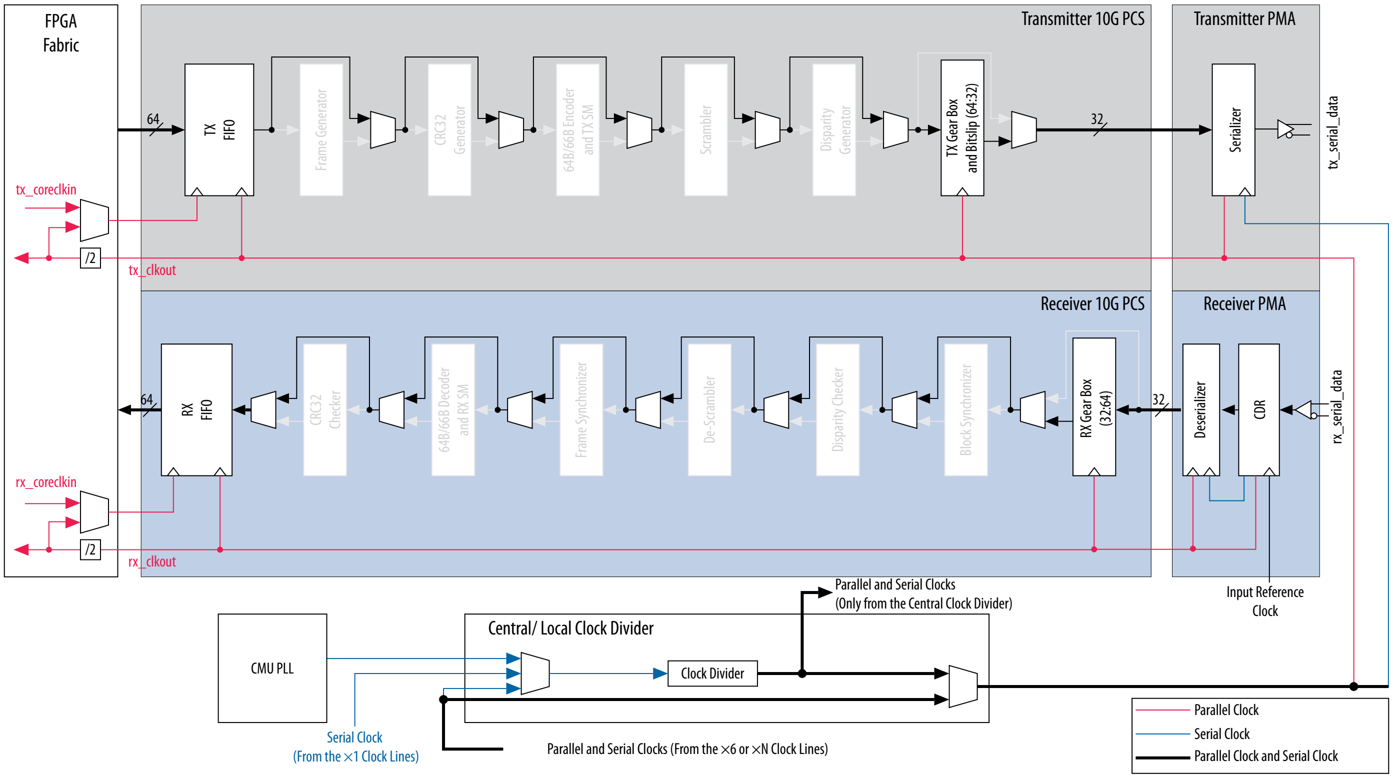
When the gear box ratio is 64:64, 40:40, or 32:32, there is no frequency difference between the read and write side of the TX and RX FIFO clocks because the gear box is the same ratio. The Quartus II software automatically connects the clocks to the read and write side of the TX FIFO and RX FIFO. In this configuration, the data from the TX FIFO is still fed to the gear box before being sent to the serializer. The gear box cannot be bypassed or disabled.

Figure 6-67: 10G PCS Low Latency Datapath with Gear Box in 64:64, 40:40, and 32:32 Ratio



When the gear box ratio is 64:32. The FPGA fabric interface width (64 bits) is exactly twice the internal transceiver datapath width. You can divide the `tx_clkout` and `rx_clkout` in the FPGA fabric by two, and use them to clock the write side of TX FIFO and the read side of RX FIFO, respectively. Select the `tx_coreclkin` and the `rx_coreclkin` ports in the Low Latency PHY IP core and connect the divided clock to these ports.

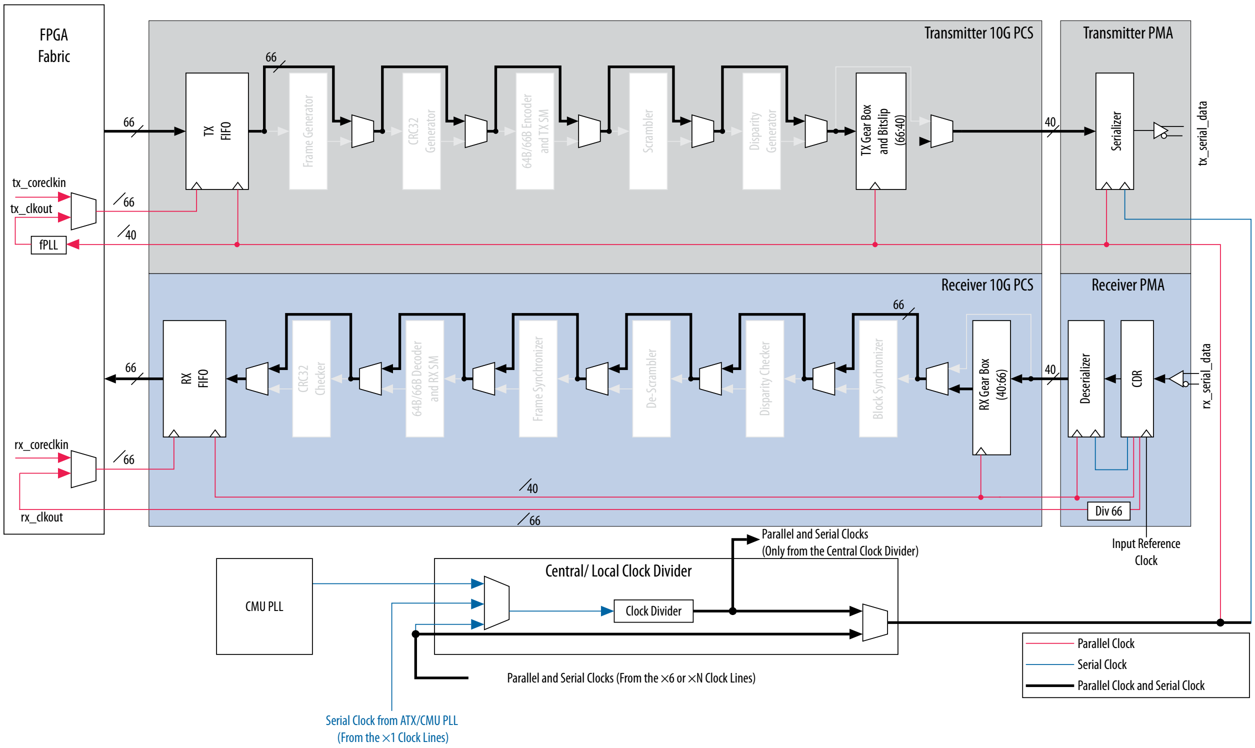
Figure 6-68: 10G PCS Low Latency Datapath with the Gear Box Ratio of 64:32



When the gear box ratio is 66:40, the `rx_clkout` parallel clock provided is a recovered clock coming from the CDR with a divided-by-66 output frequency.

The `tx_clkout` parallel clock is generated from the transmit PLL feeding a fractional PLL that is automatically instantiated from the FPGA core with a divided-by-66 output frequency.

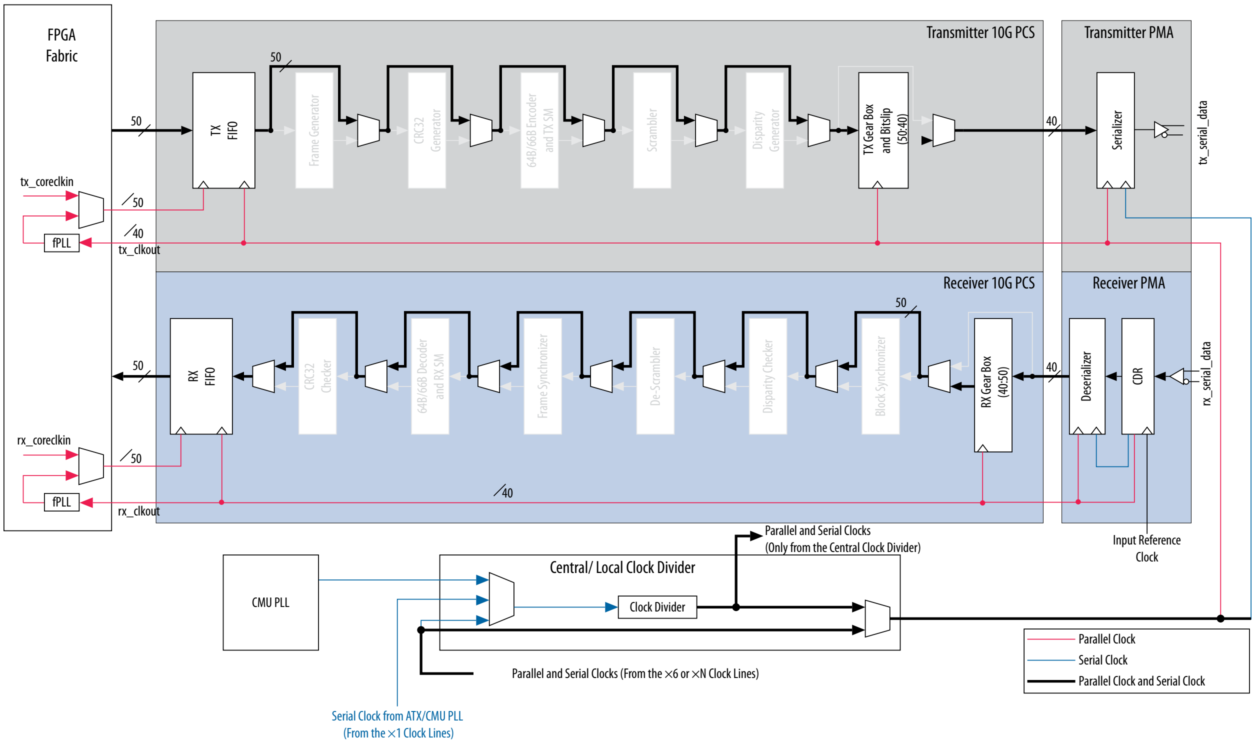
Figure 6-69: 10G PCS Low Latency Datapath with the Gear Box Ratio of 66:40



When the gear box ratio is not an integral multiple of the FPGA fabric interface width (for example, 50:40), you must instantiate a fractional PLL to provide the appropriate clock frequency to the write side of the TX FIFO. Set the division factor in the fractional PLL so that its output frequency is equal to the transmitter or lane data rate divided by 50 for the 50:40 gear box ratio. The clock source that provides the input reference clock to the fractional PLL and the CMU or ATX transmit PLL must be the same because the TX FIFO operates as a phase compensation FIFO, unlike a clock compensation or rate match FIFO. Therefore, the clock requires a zero ppm between the read and write operations.

For the receiver side, enable the `rx_coreclkin` port and connect a second fractional PLL output to the `rx_coreclkin` port. The RX FIFO operates as a phase compensation FIFO. Therefore, the read and write side of the RX FIFO must have a zero ppm difference.

Figure 6-70: 10G PCS Low Latency Datapath with the Gear Box Ratio of 50:40



Note:

(1) The clock source that provides the input reference clock to the fractional PLL (fPLL) and the CMU or ATX PLL (the CMU or ATX PLL generates the high-speed clock for the serializer) must be the same. The transmitter and the receiver FIFOs compensates only for phase differences. Therefore, the same clock source ensures zero ppm between the read and write clocks of the FIFOs.

Using the coreclkin Ports

The `tx_coreclkin` and `rx_coreclkin` ports offer the flexibility to use the `tx_clkout` and `rx_clkout` from one channel to clock the TX and RX FIFOs multiple channels for source synchronous links or if the upstream transmitters are all clocked by the same clock source. The `tx_coreclkin` and `rx_coreclkin` ports require a zero ppm difference between the `tx_clkout` and `rx_clkout` ports, respectively, with a divided-by-50 input frequency.

Related Information

For more information, refer to the “Selecting a Transmitter Datapath Interface Clock" and “Selecting a Receiver Datapath Interface Clock” sections in the Transceiver Clocking in Arria V Devices chapter

Merging Instances

You can merge transmitter and receiver instances with the different 10G PCS datapath configurations in the same 10 Gbps physical channel.

For example, the Quartus II software allows you to create the two following instances and place them in the same physical transceiver channel:

- Transmitter only instance with a 40-bit FPGA fabric interface
- Receiver only instance with a 64-bit FPGA fabric interface

However, you cannot merge a transmitter instance and receiver instance (1 channel instance) using different PCS blocks (10G PCS and standard PCS) within the same physical transceiver channel.

Transceiver Channel Placement Guidelines

Arria V GZ devices allow the placement of up to four or five channels when a CMU PLL is used or up to six channels when an ATX PLL is used with Custom and Low Latency datapath configurations with Standard PCS and 10G PCS (same data rate) within the same transceiver bank.

Related Information

Transceiver Channel Placement Guidelines on page 6-77
You can use CMU PLLs or ATX PLLs in non-bonded and bonded configurations.

Native PHY IP Configuration

The Native PHY IP is a full exposure of the transceiver hardware features with little abstraction of the physical hardware layer.

Access to both the Standard PCS and 10G PCS hardware, as well as PMA Direct modes can be enabled with full user control over the transceiver interfaces, parameters, and ports. Enable the Standard PCS and 10G PCS or PMA Direct mode to design for multi-datarate protocols, speed negotiation, and support multiple PCS datapath natively on the transceiver link.

The Transceiver Reconfiguration Controller is used to dynamically switch between the Standard PCS and 10G PCS datapaths. In addition, the Reconfiguration Controller is required for calibration, remote loopback enablement, PLL reference clock switching, channel PCS and PLL reconfiguration and switching, and to dynamically adjust PMA transmit pre-emphasis, receiver CDR, CTLE, and DFE advance settings.

Dynamic switching to and from PMA Direct mode is not supported.

Not all hardware combinations are legal or supported, so the user must have sufficient prior knowledge of the transceiver hardware, PLLs, and clocking architecture to determine valid PCS hardware setting, parameters, and combinations. All serial transceiver protocols can be supported by the Native PHY IP.

Note: Altera recommends all new serial protocol designs use the Native PHY IP with the exception of XAUI and PCI Express. A default preset is provided for ASI, SDI, SRIO, CPRI, GIGE, Interlaken, SAS, SATA, and other protocol configurations as well as Low Latency configurations for the Standard PCS and 10G PCS similar to the Low Latency PHY IP implementation. Users can also select the default preset for guidance and then modify the configurations for custom applications and have the ability to save the modified preset.

The transmit CMU or ATX Phase-Locked Loop (PLL) selection is embedded in the PHY IP. In addition, the fractional PLL (fPLL) can now be used as a transmit PLL for lane datarates up to 3.125Gbps. User must select the appropriate PLL for balancing datarate and jitter performance trade-off requirements. Unlike the other PHY IPs, the Native PHY IP does not have an Avalon Memory-Mapped (Avalon-MM) interface as the intent is to have direct access to the port interfaces. As a result, there are no embedded registers. In addition, the reset controller is also not embedded in the Native PHY IP. Altera recommends that the Transceiver PHY Reset Controller IP is used to implement the reset sequence and to make PLL sharing and merging effortless.

To implement a Native PHY link, instantiate the **Arria V Transceiver Native PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. Select options to generate valid custom transceiver configurations or select the default preset for by double-clicking in the window menu.

Related Information

- For information using the x1, xN, and feedback compensation clock bonding requirements, use case limitations, advantages, and functionality, refer to the Transceiver Clocking in Arria V Devices.
- For more information about the functions and capabilities of the Reconfiguration Controller, refer to Dynamic Reconfiguration in Arria V Devices.

Protocols and Transceiver PHY IP Support

Table 6-12: Protocols and PHY IP Features Support

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
PCIe Gen3 x1, x2, x4, x8	PHY IP Core for PCIe (PIPE) [41]	Standard and Gen3	Yes	Embedded
PCIe Gen2 x1, x2, x4, x8	PHY IP Core for PCIe (PIPE) [41]	Standard	Yes	Embedded
PCIe Gen1 x1, x2, x4, x8	PHY IP Core for PCIe (PIPE) [41]	Standard	Yes	Embedded
10GBASE-R	10GBASE-R	10G	Yes	Embedded
	Native PHY	10G	No	External Reset IP
10G/40G Ethernet	Native PHY	10G	No	External Reset IP
1G/10Gb Ethernet	1G/10GbE and 10GBASE-KR	Standard and 10G	Yes	Embedded
1G/10Gb Ethernet with 1588	1G/10GbE and 10GBASE-KR	Standard and 10G	Yes	Embedded
10G Ethernet with 1588	Native PHY	10G	No	External Reset IP
10GBASE-KR and 1000BASE-X	1G/10GbE and 10GBASE-KR	Standard and 10G	Yes	Embedded
1000BASE-X and SGMII Gigabit Ethernet	Custom PHY Standard	Standard	Yes	Embedded or External Reset IP
XAUI	XAUI PHY IP	Standard Soft-PCS	Yes	Embedded
SPAUI	Low Latency PHY	Standard and 10G	Yes	Embedded or External Reset IP
	Native PHY	Standard and 10G	No	External Reset IP

[41] Hard IP for PCI Express is also available as a Intel FPGA IP core function.

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
DDR XAUI	Low Latency PHY	Standard and 10G	Yes	Embedded or External Reset IP
	Native PHY	Standard and 10G	No	External Reset IP
Interlaken (CEI-6G/ 11G)	Interlaken PHY	10G	Yes	Embedded
	Native PHY [42]	10G	No	External Reset IP
OTU-3 (40G) via OIF SFI-5.2/SFI-5.1	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
OTU-2 (10G) via OIF SFI-5.1s	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
OTU-1 (2.7G)	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
SONET/SDH STS-768/ STM-256 (40G) via OIF SFI-5.2	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
SONET/SDH STS-768/ STM-256 (40G) via OIF SFI-5.2/SFI-5.1	Native PHY	Standard and 10G	No	External Reset IP
SONET/SDH STS-192/ STM-64 (10G) via SFP +/SFF-8431/ CEI-11G	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
SONET/SDH STS-192/ STM-64 (10G) via OIF SFI-5.1s/SxI-5/ SFI-4.2	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
SONET STS-96 (5G) via OIF SFI-5.1s	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
SONET/SDH STS-48/ STM-16 (2.5G) via SFP/TFI-5.1	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP

[42] A Soft-PCS bonding IP is required.

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
SONET/SDH STS-12/ STM-4 (0.622G) via SFP/TFI-5.1	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
Intel QPI	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	PMA-Direct	No	External Reset IP
10G SDI	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
SD-SDI/HD-SDI/ 3G-SDI	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
10G GPON/EPON	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
GPON/EPON	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
10G Fibre Channel	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
8G/4G Fibre Channel	Low Latency PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
FDR/FDR-10 Infiniband x1, x4, x12	Low Latency PHY	10G	Yes	Embedded or External Reset IP
	Native PHY	10G	No	External Reset IP
SDR/DDR/QDR Infiniband x1, x4, x12	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
CPRI 4.2/OBSAI RP3 v4.2	Deterministic PHY	Standard	Yes	Embedded
	Native PHY	Standard	No	External Reset IP

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
SRIO 2.2/1.3 [43]	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
SATA 3.0/2.0/1.0 and SAS 2.0/1.0	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
HiGig+/2+	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
JESD204A	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
ASI	Custom PHY	Standard	Yes	Embedded or External Reset IP
SPI 5 (50G)	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
Custom and other protocols	Native PHY	Standard, 10G, and PMA-Direct	No	External Reset IP

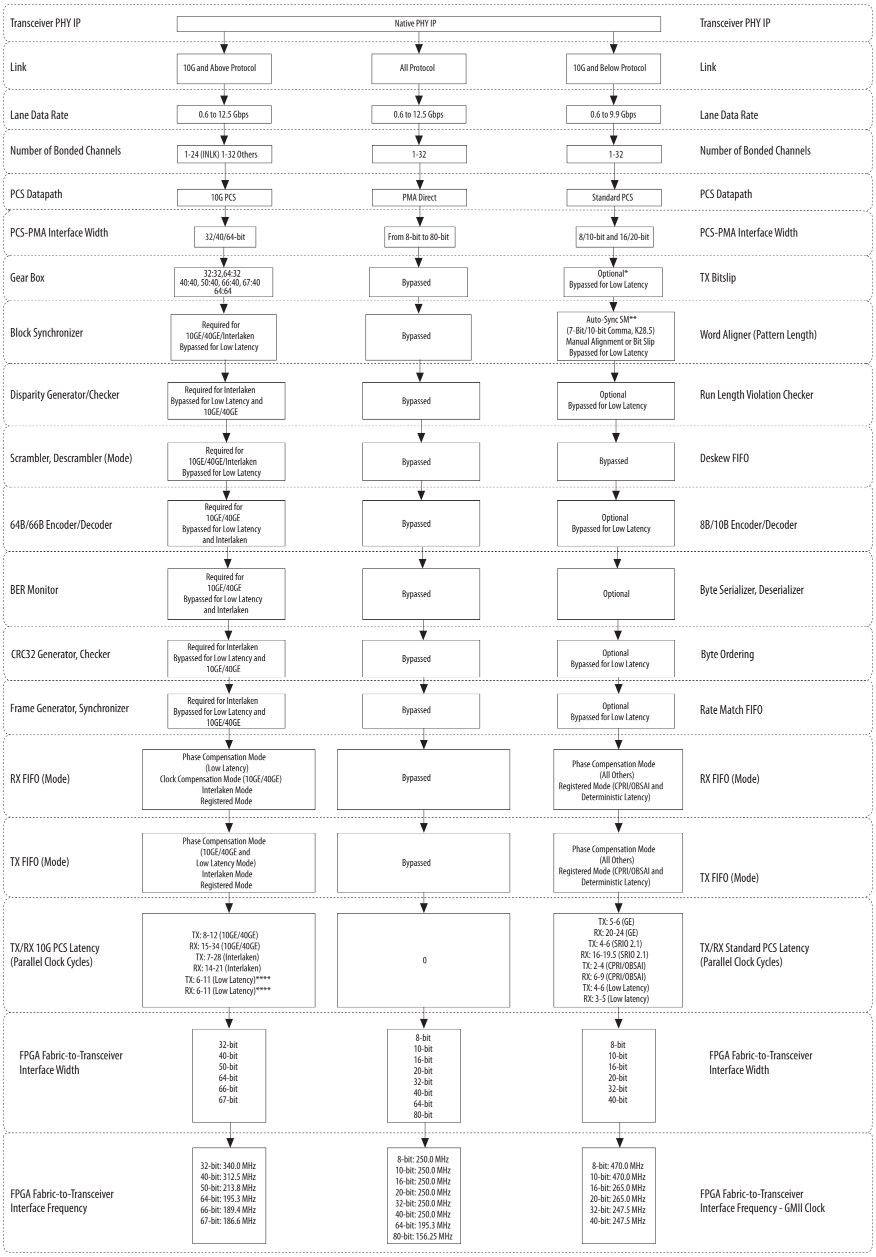
Native PHY Transceiver Datapath Configuration

The following figure shows the transceiver Standard PCS blocks, 10G PCS blocks, and their settings in addition to PMA Direct Mode available in a Native PHY IP configuration.

[43] Nx Multi-Alignment Deskew State Machine must be implemented in the core.

Figure 6-71: Transceiver Blocks in a Native PHY IP Configuration

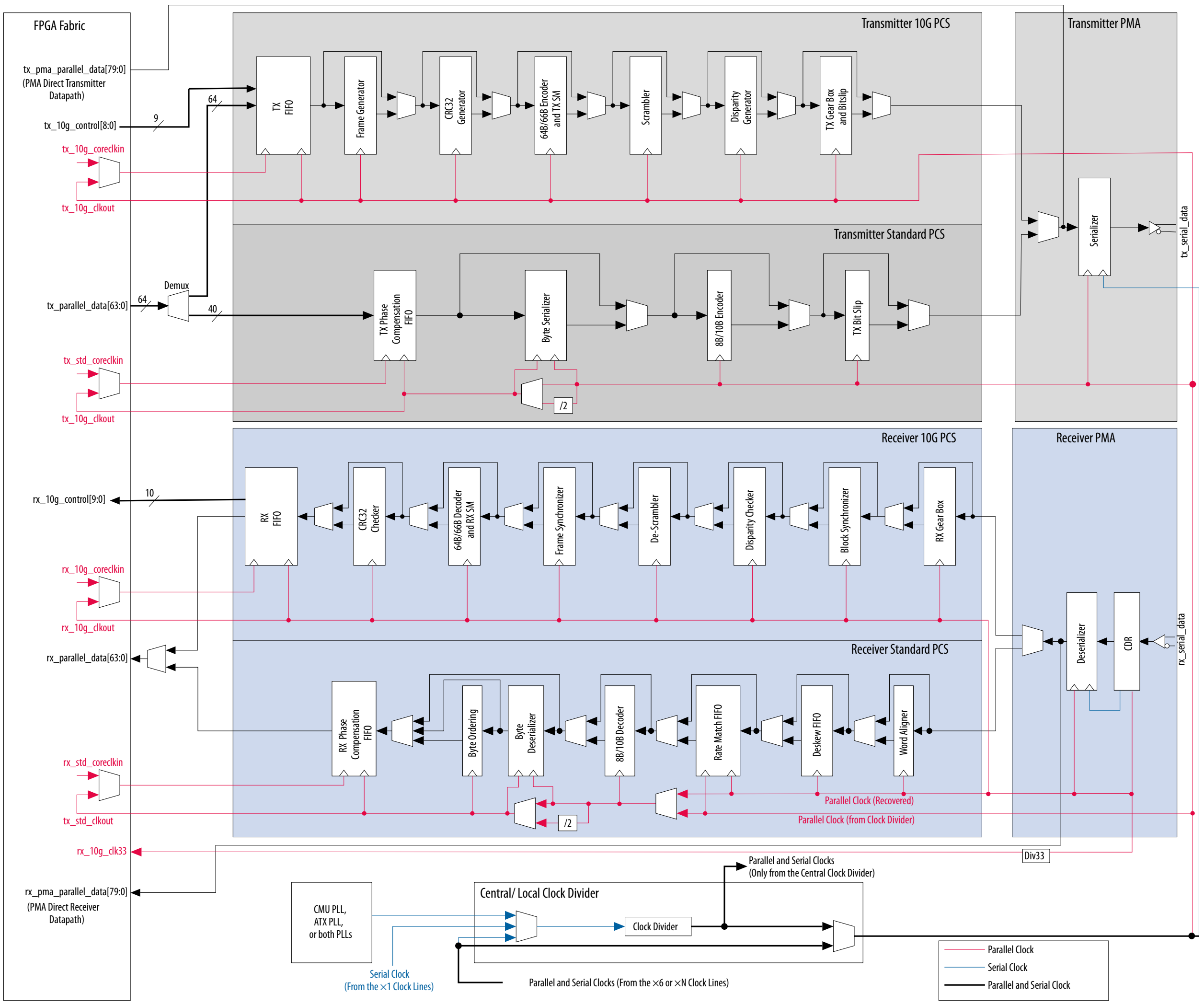
The Optional PCS blocks that are "Disabled" are not used, but incur latency. The Optional PCS blocks selected as "Bypassed" are not used and do not incur latency.



Transceiver PHY IP	Native PHY IP			Transceiver PHY IP
Link	10G and Above Protocol	All Protocol	10G and Below Protocol	Link
Lane Data Rate	0.6 to 12.5 Gbps	0.6 to 12.5 Gbps	0.6 to 9.9 Gbps	Lane Data Rate
Number of Bonded Channels	1-24 (INLK) 1-32 Others	1-32	1-32	Number of Bonded Channels
PCS Datapath	10G PCS	PMA Direct	Standard PCS	PCS Datapath
PCS-PMA Interface Width	32/40/64-bit	From 8-bit to 80-bit	8/10-bit and 16/20-bit	PCS-PMA Interface Width
Gear Box	32:32,64:32 40:40, 50:40, 66:40, 67:40 64:64	Bypassed	Optional* Bypassed for Low Latency	TX Bitslip
Block Synchronizer	Required for 10GE/40GE/Interlaken Bypassed for Low Latency	Bypassed	Auto-Sync SM** (7-Bit/10-bit Comma, K28.5) Manual Alignment or Bit Slip Bypassed for Low Latency	Word Aligner (Pattern Length)
Disparity Generator/Checker	Required for Interlaken Bypassed for Low Latency and 10GE/40GE	Bypassed	Optional Bypassed for Low Latency	Run Length Violation Checker
Scrambler, Descrambler (Mode)	Required for 10GE/40GE/Interlaken Bypassed for Low Latency	Bypassed	Bypassed	Deskew FIFO
64B/66B Encoder/Decoder	Required for 10GE/40GE Bypassed for Low Latency and Interlaken	Bypassed	Optional Bypassed for Low Latency	8B/10B Encoder/Decoder
BER Monitor	Required for 10GE/40GE Bypassed for Low Latency and Interlaken	Bypassed	Optional	Byte Serializer, Deserializer
CRC32 Generator, Checker	Required for Interlaken Bypassed for Low Latency and 10GE/40GE	Bypassed	Optional Bypassed for Low Latency	Byte Ordering
Frame Generator, Synchronizer	Required for Interlaken Bypassed for Low Latency and 10GE/40GE	Bypassed	Optional Bypassed for Low Latency	Rate Match FIFO
RX FIFO (Mode)	Phase Compensation Mode (Low Latency) Clock Compensation Mode (10GE/40GE) Interlaken Mode Registered Mode	Bypassed	Phase Compensation Mode (All Others) Registered Mode (CPRI/OBSAI and Deterministic Latency)	RX FIFO (Mode)
TX FIFO (Mode)	Phase Compensation Mode (10GE/40GE and Low Latency Mode) Interlaken Mode Registered Mode	Bypassed	Phase Compensation Mode (All Others) Registered Mode (CPRI/OBSAI and Deterministic Latency)	TX FIFO (Mode)
TX/RX 10G PCS Latency (Parallel Clock Cycles)	TX: 8-12 (10GE/40GE) RX: 15-34 (10GE/40GE) TX: 7-28 (Interlaken) RX: 14-21 (Interlaken) TX: 6-11 (Low Latency)**** RX: 6-11 (Low Latency)****	0	TX: 5-6 (GE) RX: 20-24 (GE) TX: 4-6 (SRIO 2.1) RX: 16-19.5 (SRIO 2.1) TX: 2-4 (CPRI/OBSAI) RX: 6-9 (CPRI/OBSAI) TX: 4-6 (Low Latency) RX: 3-5 (Low latency)	TX/RX Standard PCS Latency (Parallel Clock Cycles)
FPGA Fabric-to-Transceiver Interface Width	32-bit 40-bit 50-bit 64-bit 66-bit 67-bit	8-bit 10-bit 16-bit 20-bit 32-bit 40-bit 64-bit 80-bit	8-bit 10-bit 16-bit 20-bit 32-bit 40-bit	FPGA Fabric-to-Transceiver Interface Width
FPGA Fabric-to-Transceiver Interface Frequency	32-bit: 340.0 MHz 40-bit: 312.5 MHz 50-bit: 213.8 MHz 64-bit: 195.3 MHz 66-bit: 189.4 MHz 67-bit: 186.6 MHz	8-bit: 250.0 MHz 10-bit: 250.0 MHz 16-bit: 250.0 MHz 20-bit: 250.0 MHz 32-bit: 250.0 MHz 40-bit: 250.0 MHz 64-bit: 195.3 MHz 80-bit: 156.25 MHz	8-bit: 470.0 MHz 10-bit: 470.0 MHz 16-bit: 265.0 MHz 20-bit: 265.0 MHz 32-bit: 247.5 MHz 40-bit: 247.5 MHz	FPGA Fabric-to-Transceiver Interface Frequency - GMII Clock

Figure 6-72: Native PHY IP Datapath Configuration

The following figure shows the Standard PCS and 10G PCS blocks, their associated datapaths, and the PMA Direct datapath available for implementation with the Native PHY IP.



Standard PCS Features

The Standard PCS can reach lane datarates up to 9.9 Gbps with the widest PCS-PMA width and FPGA fabric-to-transceiver interface width configuration. The Standard PCS is used when supporting protocols with lane datarates below 10 Gbps such as Gigabit Ethernet, CPRI/OBSAI, SD/HD/3G-SDI, HiGig, Hypertransport, SRIO, JESD204A, SATA and SAS, 1G/2G/4G/8G Fibre Channel, GPON/EPON, SFI-4.2/SFI-5.1, TFI, SPI-4.2/SPI-5.1, STS-12/12c, STS-48/48c, OTU-0.

Standard PCS Receiver and Transmitter Blocks

To implement a Native PHY link with the Standard PCS datapath, instantiate the **Arria V Transceiver Native PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. Select option to enable the Standard PCS by checking the box. A Standard PCS tab appears with the parameters and configuration options for each block.

The following blocks can be enabled or disabled and configured in the Standard PCS.

- Word Aligner
- Deskew FIFO
- Rate Match FIFO
- 8B/10B Encoder/Decoder
- Byte Serializer/De-Serializer
- Byte Ordering
- Receive Phase Compensation FIFO (Can also be configured as registered mode)
- Transmit Phase Compensation FIFO (Can also be configured as registered mode)
- TX Bitslipper

Related Information

- **Transceiver Architecture in Arria V Devices**
- **Altera Transceiver PHY IP Core User Guide**

10G PCS Supported Features

The 10G PCS supports protocols with lane datarates that are 10Gbps and above, such as 10/40 Gigabit Ethernet, Interlaken, SPAUI, 10G SDI, 10G Fibre Channel, Infiniband, 10G GPON/EPON, SFI-5.2, STS-192/192c, STS-768/768c, OTU-2/3. The 10G PCS can reach lane datarates up to 12.5 Gbps with the widest FPGA fabric-to-transceiver interface width configuration.

10G PCS Receiver and Transmitter Blocks

To implement a Native PHY link with the 10G PCS datapath, instantiate the **Transceiver Native PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. When you select the 10G PCS option, a 10G PCS tab appears with the parameters and configuration options for each block.

The following blocks below can be enabled and disabled and configured in the 10G PCS.

- Receive and Transmit FIFO
- CRC32 Generator/Checker
- Metaframe Generator/Synchronizer
- 64B/66B Encoder/Decoder
- Scrambler/Descrambler
- Disparity Generator/Checker
- Block Synchronizer
- Multi-Gearbox

The hard PCS blocks natively support 10/40 Gigabit Ethernet and Interlaken. The remaining protocols are supported via 10G PCS Low Latency datapath configuration with the appropriate gearbox ratios.

10/40 Gigabit Ethernet Blocks Supported Configuration:

- Receiver FIFO in Clock Compensation Mode and Transmit FIFO in Phase Compensation Mode
- 64B/66B Encoder/Decoder
- Scrambler/Descrambler
- Block Synchronizer
- 66:40 Gearbox Ratio

10/40 Gigabit Ethernet Blocks with 1588 Supported Configuration:

- Receiver and Transmit FIFO in Registered Mode
- 64B/66B Encoder/Decoder
- Scrambler/Descrambler
- Block Synchronizer
- 66:40 Gearbox Ratio

Interlaken Blocks Supported Configuration:

- Receiver and Transmit FIFO in Interlaken Elastic Buffer (Generic) Mode
- CRC32 Generator/Checker
- Metaframe Generator/Synchronizer
- Scrambler/Descrambler
- Disparity Generator/Checker
- Block Synchronizer
- 67:40 Gearbox Ratio

SFI-5.2 Blocks Supported Configuration:

- Receiver and Transmit FIFO in Phase Compensation Mode
- 64:64, 40:40, 64:32, and 32:32 Gearbox Ratios

10G SDI Blocks Supported Configuration:

- Receiver and Transmit FIFO in Phase Compensation Mode
- 50:40 Gearbox Ratio

Other Protocol Blocks Supported Configuration in Basic Mode:

- Receiver and Transmit FIFO in Phase Compensation Mode
- 64:64, 66:40, 40:40, 64:32, and 32:32 Gearbox Ratios

Related Information

- **Transceiver Architecture in Arria V Devices**
- **Altera Transceiver PHY IP Core User Guide**

Receiver and Transmit Gearbox in Native PHY IP

The Native PHY IP supports many 10G PCS:PMA gearbox ratios.

Users have the freedom to choose the best gearbox ratio that matches their core IP. The 67:40 is mainly used for Interlaken configurations and the 66:40 ratio is mainly used in 10, 40, and 100 Gigabit Ethernet configurations and the 50:40 is used in 10 Gigabit SDI applications. The other ratios can support additional standard communication and transport protocols such as GPON, EPON, SFI-5.2 and OTN.

10G PCS Supported Gearbox Ratios:

- 64:64 PCS:PMA Width
- 67:40 PCS:PMA Width
- 66:40 PCS:PMA Width
- 50:40 PCS:PMA Width
- 40:40 PCS:PMA Width
- 64:32 PCS:PMA Width
- 32:32 PCS:PMA Width

10G Datapath Configurations with Native PHY IP

Table 6-13: 10G PCS Datapath Configurations

The table lists the 10G PCS datapath configuration for 10/40 Gigabit Ethernet, 10/40 Gigabit Ethernet with 1588, Interlaken, 10G SDI, and other 10G protocols.

Transceiver PHY IP	Native PHY IP					
Link	10/ 40GBASE-R/ KR	10/ 40GBASE-R with 1588	Interlaken	SFI-5.2	10G SDI	Other 10G Protocols (Basic Mode)
Lane Datarate	10.3125Gbps	10.3125Gbps	3.125 - 12.5Gbps	0.6 - 12.5Gbps[44]	10.692Gbps	0.6 - 12.5Gbps [44]
PMA Channel Bonding Option[45] [46]	Non-bonded, xN,feedback compensation	Non-bonded, xN, feedback compensation	Non-bonded	Non-bonded, xN, feedback compensation	Non-bonded, xN, feedback compensation	Non-bonded, xN, feedback compensation
PCS Datapath	10G PCS	10G PCS	10G PCS	10G PCS	10G PCS	10G PCS
PCS-PMA Interface Width (Serialization Factor)	40-bit	40-bit	40-bit	32/40/64-bit	40-bit	32/40/64-bit
Gearbox Ratios	66:40 [47]	66:40 [47]	67:40	32:32, 64:32[47], 40:40, 64:64	50:40 [47]	32:32, 64:32[47], 40:40, 66:40[47], 64:64

[44] Gearbox ratios of 64:32 and 32:32 have a maximum supported datarate of 10.88Gbps.

[45] For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

[46] Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

[47] May require the use of an internal fractional PLL (fPLL) for selected Gearbox ratio.

Transceiver PHY IP	Native PHY IP					
Link	10/ 40GBASE-R/ KR	10/ 40GBASE-R with 1588	Interlaken	SFI-5.2	10G SDI	Other 10G Protocols (Basic Mode)
Block Synchronizer	Enabled	Enabled	Enabled	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
Disparity Generator, Checker	Bypassed	Bypassed	Enabled	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
Scrambler, Descrambler	Enabled	Enabled	Enabled	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
64B/66B Encoder, Decoder	Enabled	Enabled	Bypassed	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
BER Monitor	Enabled	Enabled	Bypassed	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
CRC32 Generator, Checker	Bypassed	Bypassed	Enabled	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
Frame Generator, Synchronizer	Bypassed	Bypassed	Enabled	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)	Bypassed (Low Latency Mode)
RX FIFO (Mode)	Clock Compensation Mode	Registered Mode	Interlaken Mode	Phase Compensation Mode	Phase Compensation Mode	Phase Compensation Mode (Low Latency Mode)
TX FIFO (Mode)	Phase Compensation Mode	Registered Mode	Interlaken Mode	Phase Compensation Mode	Phase Compensation Mode	Phase Compensation Mode (Low Latency Mode)

Transceiver PHY IP	Native PHY IP					
Link	10/ 40GBASE-R/ KR	10/ 40GBASE-R with 1588	Interlaken	SFI-5.2	10G SDI	Other 10G Protocols (Basic Mode)
TX/RX 10G PCS Latency (Parallel Clock Cycles) [48]	TX: 8-12 RX: 15-34	TX: 1-4 RX: 2-5	TX: 7-28 RX: 14-21	TX: 6-10 (64:32) TX: 7-10 (64:64, 40:40, 32:32) RX: 6-10 (64:32) RX: 7-10 (64:64, 40:40, 32:32)	TX: 7-11 RX: 6-12	TX: 6-10 (64:32) TX: 6-11 (66:40) TX: 7-10 (64:64, 40:40, 32:32) RX: 6-10 (64:32) RX: 6-11 (66:40) RX: 7-10 (64:64, 40:40, 32:32)
FPGA Fabric-to- Transceiver Interface Widths	66-bit	66-bit	67-bit	32-bit 40-bit 64-bit	50-bit	32-bit 40-bit 64-bit 66-bit
FPGA Fabric-to- Transceiver Interface Width Maximum Frequencies	66-bit: 156.25 MHz	66-bit: 156.25 MHz	67-bit: 78.125-312.5 MHz [49]	32-bit (32:32): 340.0 MHz 40-bit (40:40): 312.5 MHz 64-bit (64:32): 170.0 MHz [50] 64-bit (64:64): 195.4 MHz	50-bit: 213.8 MHz [49]	32-bit (32:32): 340.0 MHz 40-bit (40:40): 312.5 MHz 64-bit (64:32): 170.0 MHz [50] 64-bit (64:64): 195.4 MHz 66-bit (66:40): 189.4 MHz [49]

[48] PCS Latency values are with default recommended FIFO partially full and partially empty values. Disabled if Standard PCS 8B/10 Encoder/Decoder is used.

[49] PCS `tx_clkout` frequency output is lane datarate/40 for 10G-SDI, Interlaken, and Basic Mode.

[50] PCS `tx_clkout` frequency output is lane datarate/32 for SFI-S and Basic Mode.

PMA Direct Supported Features

The PMA Direct is used to support protocols that require extremely low or zero transceiver PCS latency such as QPI. In PMA Direct mode, the transceiver can reach lane data rates up to 12.5Gbps with the widest FPGA fabric-to-transceiver interface width configuration.

There are no PCS blocks in the PMA Direct configuration, so clock phase compensation must be designed in the fabric core. Data and clock signals are interfaced directly to the transceiver PMA. Consequently, you must also compensate for the timing and clock phase differences from the core fabric interface of the FPGA to the transceiver PMA. The PMA interface width has a wide range of selections from 8-bit, 10-bit, 16-bit, 20-bit, 32-bit, 40-bit, 64-bit, and 80-bit. The FPGA fabric interface width is fixed at 80-bit and you must select the correct ports for their PMA interface width configurations.

To implement a Native PHY link with the PMA Direct datapath, instantiate the **Transceiver Native PHY IP** in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. Do not select the options to enable the Standard or 10G PCS. The Standard and 10G PCS tabs do not appear, indicating that the PMA Direct datapath configuration has been selected.

Figure 6-72 shows the transceiver PMA Direct datapath and clocking in the device channels.

Channel and PCS Datapath Dynamic Switching Reconfiguration

The Native PHY IP is the only PHY IP that can support transceiver channel dynamic switching between Standard PCS and 10G PCS. Dynamic switching to and from PMA Direct mode is not supported. The dynamic switching mechanism via streamer-based reconfiguration as well as the associated transceiver PLL, standard PMA, and advance transceiver PMA features reconfiguration is employed with the Reconfiguration Controller IP.

Related Information

- Dynamic Reconfiguration in Arria V Devices
- Altera Transceiver PHY IP Core User Guide

Document Revision History

Table 6-14: Document Revision History

Date	Version	Changes
September 2014	2014.09.30	Changed references to MegaWizard Plug-In Manager to IP Catalog.
March 2014	2014.03.07	Updated "Sample Channel Width Options for Supported Serial Data Rates" table in the CPRI and OBSAI section.

Date	Version	Changes
October 2013	2013.10.18	• Updated "Advanced Channel Placement Guidelines for PIPE Configurations" section. • Updated "Transceiver Clocking for PCIe Gen3" section. • Updated "Protocols and Transceiver PHY IP Support" section.
May 2013	2013.05.06	• Added link to the known document issues in the Knowledge Base. • Added second figure to the "10GBASE-R and 10GBASE-KR" section. • Added the "10GBASE-KR Forward Error Correction" section. • Updated the "Transceiver Channel Placement Guidelines for Gen1, Gen2, and Gen3 PIPE Configurations" section. • Added the "Advance Channel Placement Guidelines for PIPE Configurations" section.
November 2012	2012.11.19	Initial release.

Transceiver Loopback Support in Arria V Devices

7

2014.09.30

AV53006 Subscribe Send Feedback

The Arria V loopback options allow you to verify how different functional blocks work in the transceiver.

Related Information

Arria V Device Handbook: Known Issues

Lists the planned updates to the Arria V Handbook chapters.

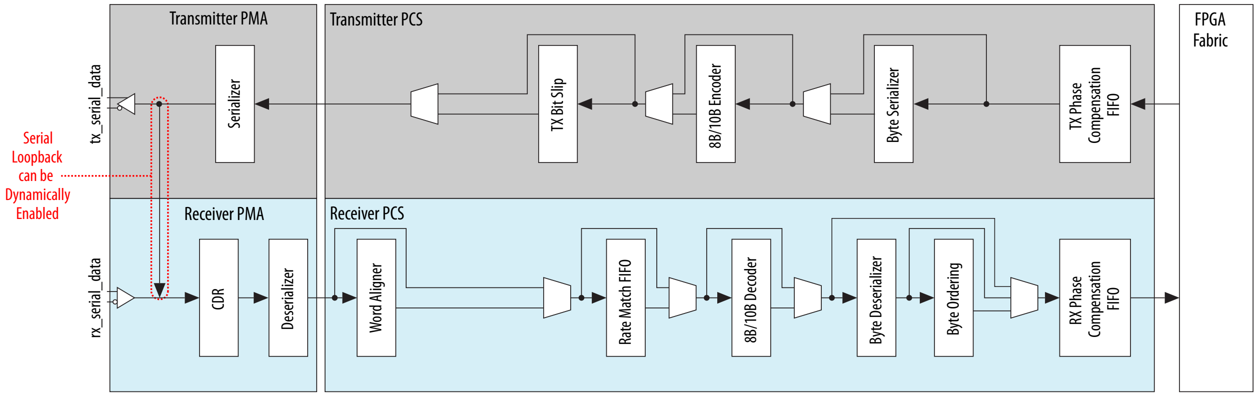
Serial Loopback

Serial loopback is a debugging aid to ensure that the enabled PCS and PMA blocks in the transmitter and receiver channels function correctly.

Serial loopback is available for all transceiver configurations except the PIPE mode. You can use serial loopback as a debugging aid to ensure that the enabled physical coding sublayer (PCS) and physical media attachment (PMA) blocks in the transmitter and receiver channels are functioning correctly. Furthermore, you can dynamically enable serial loopback on a channel-by-channel basis.

The data from the FPGA fabric passes through the transmitter channel and is looped back to the receiver channel, bypassing the receiver buffer. The received data is available to the FPGA logic for verification.

Figure 7-1: Serial Loopback Datapath



You can enable serial loopback using the PHY IP Parameter Editor or the reconfiguration controller, depending on which PHY IP mode you select. When you enable serial loopback, the transmitter channel sends data to both the tx_serial_data output port and to the receiver channel. The differential output voltage on the tx_serial_data port is based on the selected differential output voltage (V_{OD}) settings.

Note: For more information about the PHY IP core registers, refer to the Altera Transceiver PHY IP User Guide.

The looped-back data is forwarded to the receiver clock data recovery (CDR). You must provide an alignment pattern for the word aligner to enable the receiver channel to retrieve the byte boundary.

If the device is not in the serial loopback configuration and is receiving data from a remote device, the recovered clock from the receiver CDR is locked to the data from the remote source.

If the device is placed in the serial loopback configuration, the data source to the receiver changes from the remote device to the local transmitter channel—prompting the receiver CDR to start tracking the phase of the new data source. During this time, the recovered clock from the receiver CDR may be unstable. Because the receiver PCS is running off of this recovered clock, you must place the receiver PCS under reset by asserting the rx_digitalreset signal during this period.

Note: When moving into or out of serial loopback, you must assert the rx_digitalreset signal for a minimum of two parallel clock cycles.

Serial Loopback for Arria V GZ Devices

For Arria V GZ devices, you enable serial loopback by accessing the register space within the reconfiguration controller through the Avalon-MM interface.

Related Information

Altera Transceiver PHY IP Core User Guide

Forward Parallel Loopback

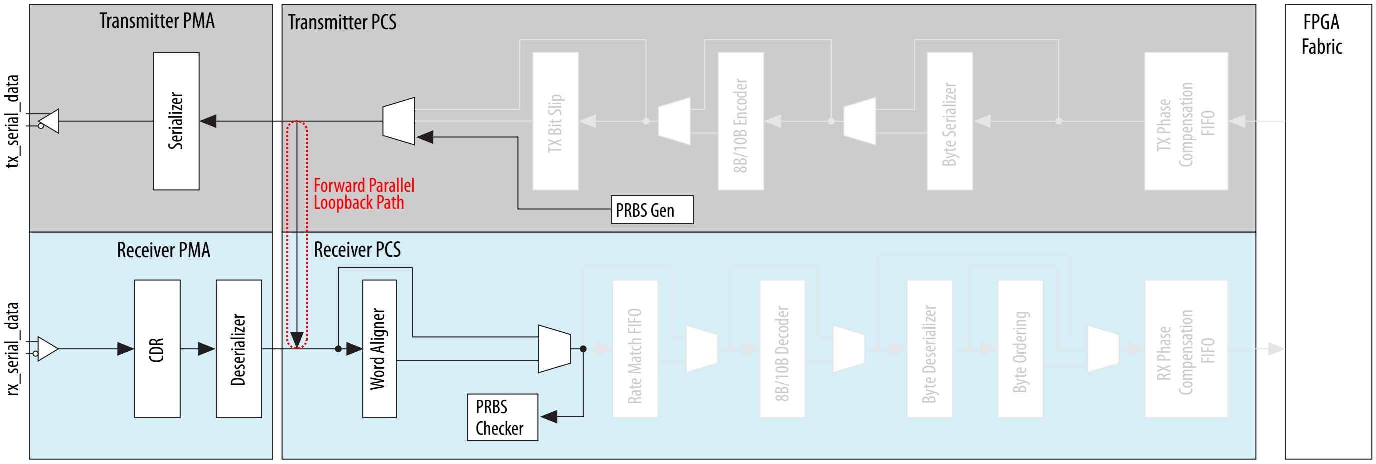
Forward parallel loopback is a debugging aid to ensure the enabled PCS blocks in the transmitter and receiver channel function correctly.

Forward parallel loopback is only available in transceiver Native PHY. You enable forward parallel loopback by enabling the PRBS test mode, through the dynamic reconfiguration controller. You must perform a rx_digitalreset after the dynamic reconfiguration operation has completed.

Parallel data travels across the forward parallel loopback path, passing through the RX word aligner, and finally verified inside the RX PCS PRBS verifier block. Check the operations status from the FPGA fabric.

Figure 7-2: Parallel Loopback Datapath

The following figure shows the parallel PRBS data generated by the TX PCS PRBS generator block.



Note: Usage details for the feature are described in the Altera Transceiver PHY IP Core User Guide.

Related Information

Altera Transceiver PHY IP Core User Guide

PIPE Reverse Parallel Loopback

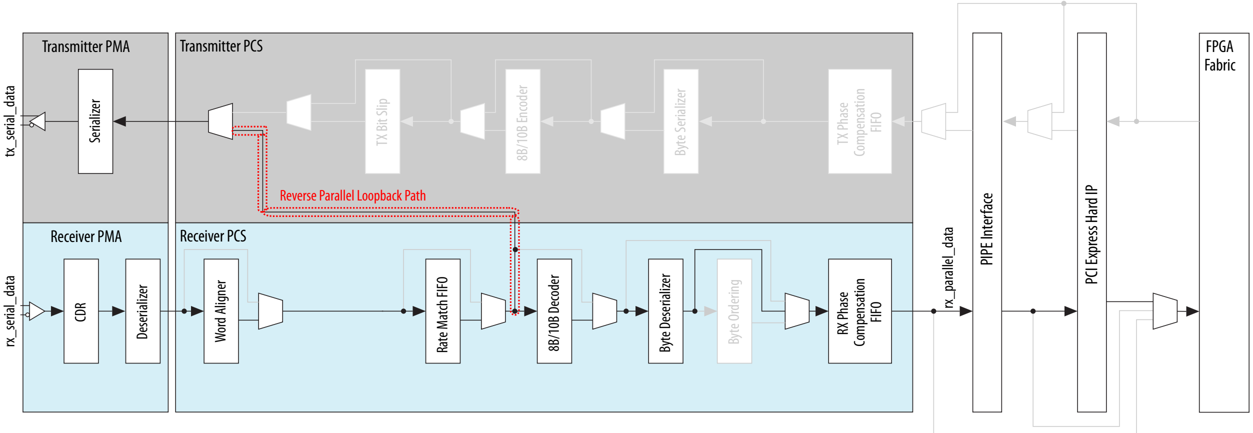
For debugging, the PIPE Reverse Parallel Loopback option uses parallel data through the rate match FIFO, transmitter serializer, and the `tx_serial_data` port path.

PIPE reverse parallel loopback is only available in the PCIe® configuration for Gen1 and Gen2 data rates. The following figure shows the received serial data passing through the receiver CDR, deserializer, word aligner, and rate match FIFO buffer. The parallel data from the rate match FIFO is then looped back to the transmitter serializer and transmitted out through the `tx_serial_data` port. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal.

PIPE reverse parallel loopback is compliant with the PCIe 2.0 specification. To enable this loopback configuration, assert the `pipe_txdetectrx_loopback` signal.

Note: PIPE reverse parallel loopback is the only loopback option supported in the PCIe configuration.

Figure 7-3: PIPE Reverse Parallel Loopback Configuration Datapath



Note: Grayed-out blocks are not active when the PIPE reverse parallel loopback is enabled.

PIPE Reverse Parallel Loopback for Arria V GZ Devices

To enable the loopback configuration for Arria V GZ devices, assert the `tx_detectrx_loopback` signal.

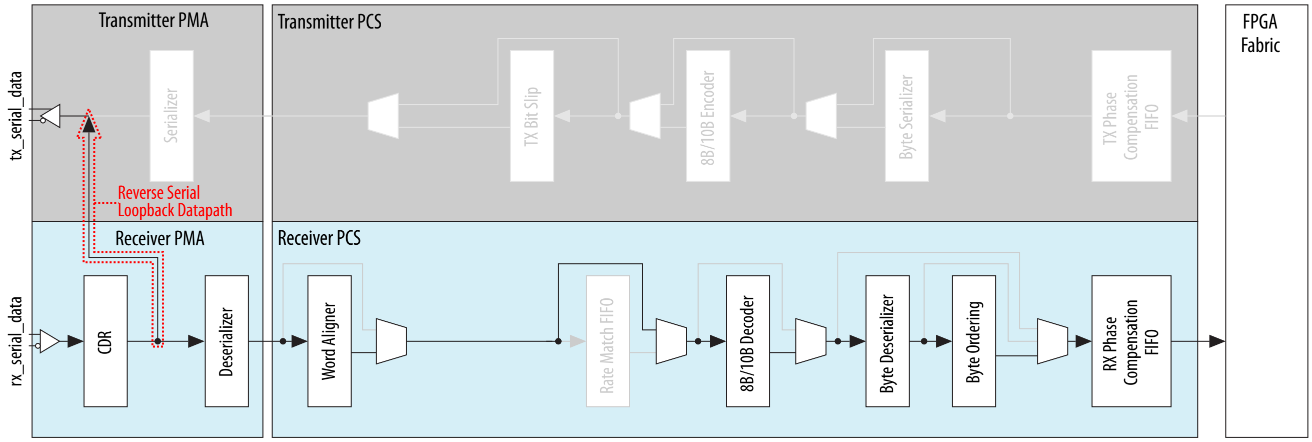
Reverse Serial Loopback

The Reverse Serial Loopback option debugs with data through the `rx_serial_data` port, receiver CDR, and `tx_serial_data` port path.

You can enable reverse serial loopback through the reconfiguration controller. In reverse serial loopback, the data is received through the `rx_serial_data` port, re-timed through the receiver CDR, and sent to the `tx_serial_data` port. The received data is also available to the FPGA logic. No dynamic pin control is available to select or deselect reverse serial loopback.

The transmitter buffer is the only active block in the transmitter channel. You can change the V_{OD} and the pre-emphasis first post tap values on the transmitter buffer through the dynamic reconfiguration controller. Reverse serial loopback is often implemented when using a bit error rate tester (BERT) on the upstream transmitter.

Figure 7-4: Reverse Serial Loopback Datapath



Note: Grayed-out blocks are not active when the reverse serial loopback is enabled.

Reverse Serial Loopback for Arria V GZ Devices

Enable reverse serial loopback by accessing the register space within the reconfiguration controller through the Avalon-MM interface.

Note: For the register definitions needed to enable this functionality, refer to the Altera Transceiver PHY IP Core User Guide.

In reverse serial loopback, the data is received through the `rx_serial_data` port, re-timed through the receiver CDR, and sent out to the `tx_serial_data` port. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. No dynamic pin control is available to select or deselect reverse serial loopback.

You set the reverse serial loopback with the PMA analog registers in the reconfiguration controller.

The only transmitter channel resource used when implementing reverse serial loopback is the transmitter buffer. You can define the V_{OD} and first post tap values on the transmitter buffer using assignment statements in the project .qsf or in the Quartus II Assignment Editor. You can also change these values dynamically with the reconfiguration controller.

Note: For more information about how to dynamically change these analog settings, refer to the Altera Transceiver PHY IP Core User Guide.

Reverse serial loopback is often implemented when using an external bit error rate tester (BERT) on the upstream transmitter.

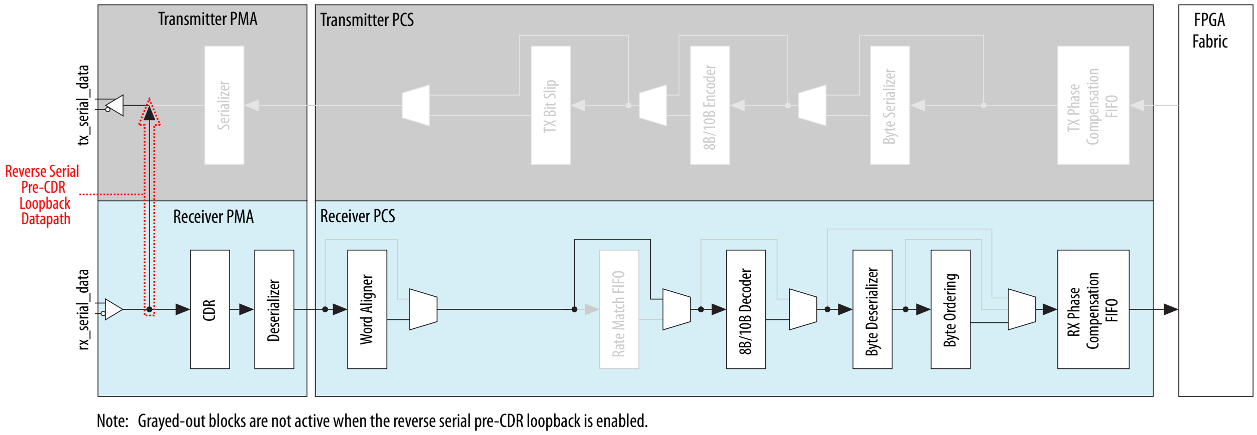
Related Information

Altera Transceiver PHY IP Core User Guide

Reverse Serial Pre-CDR Loopback

The reverse serial pre-CDR loopback option debugs with a data path through the `rx_serial_data` port to the `tx_serial_data` port, and before the receiver CDR.

Figure 7-5: Reverse Serial Pre-CDR Loopback Datapath



You can enable reverse serial pre-CDR loopback through the reconfiguration controller. In reverse serial pre-CDR loopback, the data received through the `rx_serial_data` port is looped back to the `tx_serial_data` port before the receiver CDR. The received data is also available to the FPGA logic.

The transmitter buffer is the only active block in the transmitter channel. You can change the VOD on the transmitter buffer through the dynamic reconfiguration controller. The pre-emphasis settings for the transmitter buffer cannot be changed in this configuration.

Reverse Serial Pre-CDR Loopback for Arria V GZ Devices

Enable the reverse serial pre-CDR loopback by accessing the register space within the reconfiguration controller through the Avalon-MM interface.

Note: For the register definitions needed to enable this functionality, refer to the Altera Transceiver PHY IP Core User Guide.

In reverse serial pre-CDR loopback, the data received through the `rx_serial_data` port is looped back to the `tx_serial_data` port before the receiver CDR. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. No dynamic pin control is available to select or deselect reverse serial pre-CDR loopback.

Set the reverse serial pre-CDR loopback with the PMA analog registers in the reconfiguration controller.

The only transmitter channel resource used when implementing reverse serial pre-CDR loopback is the transmitter buffer. You can change the V_{OD} on the transmitter buffer in the available Parameter Editor of the available PHY IP or using the reconfiguration controller. The receiver data characteristics that are looped back in reverse serial pre-CDR loopback are preserved by the transmitter buffer. The pre-emphasis settings for the transmitter buffer cannot be changed in this configuration.

Related Information

Altera Transceiver PHY IP Core User Guide

Document Revision History

Table 7-1: Document Revision History

Date	Version	Changes
September 2014	2014.09.30	• Changed Reverse Serial Pre-CDR Loopback section to indicate that VOD transmitter buffer settings can be modified through the Parameter Editor and the reconfiguration controller IP. • Changed MegaWizard Plug-in Manager reference to Parameter Editor.
May 2013	2013.05.06	• Added link to the known document issues in the Knowledge Base. • Updated the Serial Loopback topic. • Added the Forward Parallel Loopback topic. • Updated the Reverse Serial Loopback topic. • Updated the Reverse Serial Pre-CDR Loopback topic.
November 2012	2012.11.19	• Reorganized content and updated template. • Minor updates for the Quartus II software version 12.1.
June 2012	1.2	Minor updates for the Quartus II software version 12.0.
November 2011	1.1	Added the following two loopback options: • “Reverse Serial Loopback” • “Reverse Serial Pre-CDR Loopback”
August 2011	1.0	Initial release.

Dynamic Reconfiguration in Arria V Devices

8

2019.04.08

AV53007 Subscribe Send Feedback

The transceiver reconfiguration controller offers several different dynamic reconfiguration modes. You can choose the appropriate reconfiguration mode that best suits your application needs. All the dynamic reconfiguration modes are implemented through the transceiver Reconfiguration Controller PHY IP.

Related Information

Arria V Device Handbook: Known Issue

Lists the planned updates to the *Arria V Device Handbook* chapters.

Dynamic Reconfiguration Features

The following table lists the available dynamic reconfiguration features.

Table 8-1: Reconfiguration Features

Reconfiguration Feature	Description	Affected Blocks
Offset Cancellation	Counter offset variations due to process operation for the analog circuit. This feature is mandatory if you use receivers.	CDR For Arria V GZ devices, offset cancellation is also available for the RX buffer.
DCD Calibration	Compensates for the duty cycle distortion caused by clock network skew.	TX buffer and clock network skew Arria V GZ devices do not support DCD calibration.
Analog Controls Reconfiguration	Fine-tune signal integrity by adjusting the transmitter (TX) and receiver (RX) analog settings while bringing up a link.	Analog circuit of TX and RX buffer
Loopback Modes	Enable or disable Pre- and Post-CDR Reverse Serial Loopback dynamically.	PMA

Reconfiguration Feature	Description	Affected Blocks
Data Rate Change	Increase or decrease the data rate (/1, /2, /4, /8) for autonegotiation purposes such as CPRI and SATA/SAS applications	TX Local clock dividers
	Reconfigure the TX PLL settings for protocols with multi-data rate support such as CPRI	TX PLL
	Switch between multiple TX PLLs for multi-data rate support	• TX PLL • Fractional PLL (Reconfigure the fPLL data rate with the ALTERA_PLL_RECONFIG megafunction.)
	Channel reconfiguration—Reconfigure the RX CDR from one data rate to another data rate	CDR
	FPGA fabric - transceiver channel data width reconfiguration	FPGA fabric - transceiver channel interface.
On-Chip Signal Quality Monitoring Supported by Arria V GZ devices	Eye Viewer is a debug and diagnostic tool that analyzes the incoming data, including the receiver’s gain, noise level, and jitter after the receive buffer.	CDR
Continuous Time Linear Equalization Supported by Arria V GZ devices	Compensates for backplane losses and dispersion which degrade signal quality. You can implement it adaptively once or continuously.	RX PMA
Decision Feedback Equalization Supported by Arria V GZ devices	Helps compensate for backplane attenuation because of insufficient bandwidth. You can implement it once or continuously.	RX PMA

Related Information

- Backplane Applications with 28 nm FPGAs

- **AN661: Implementing Fractional PLL Reconfiguration with ALTERA_PLL and ALTERA_PLL_RECONFIG Megafunctions**
  For information about reconfiguration of the fPLL data rate.

Offset Cancellation

The CDR in all Arria V devices requires offset cancellation. Offset cancellation is available for the RX buffer in Arria V GZ devices only.

Every transceiver channel has offset cancellation circuitry to compensate for the offset variations that are caused by process operations. The offset cancellation circuitry is controlled by the offset cancellation control logic IP within the Transceiver Reconfiguration Controller. Resetting the Transceiver Reconfiguration Controller during user mode does not trigger the offset cancellation process.

When offset cancellation calibration is complete, the `reconfig_busy` status signal is deasserted to indicate the completion of the process.

The clock (`mgmt_clk_clk`) used by the Transceiver Reconfiguration Controller is also used for transceiver calibration. For Arria V GT, GX, ST, and SX devices, `mgmt_clk_clk` must be within the range of 75-125 MHz. For Arria V GZ devices, `mgmt_clk_clk` must be within the range of 100-125 MHz. If the clock (`mgmt_clk_clk`) is not free-running, hold the reconfiguration controller reset (`mgmt_rst_reset`) until the clock is stable.

The clock (`mgmt_clk_clk`) used by the Transceiver Reconfiguration Controller is also used for transceiver calibration and must be 75-125 MHz if the Hard IP for PCIe Express IP core is not enabled.

Transmitter Duty Cycle Distortion Calibration

The duty cycle calibration function tunes the transmitter to minimize duty cycle distortion.

The transmitter clocks generated by the CMU that travel across the clock network may introduce duty cycle distortions (DCD). Reduce DCD with the DCD calibration IP that is integrated in the transceiver reconfiguration controller.

For improved TX jitter performance, enabled the DCD calibration IP in Arria V GX and GT devices if either of the following conditions are met:

- Data rate is ≥ 4915.2 Mbps
- Clock network switching (TX PLL switching) and the data rate is ≥ 4915.2 Mbps

The following DCD calibration modes are supported:

- DCD calibration at power-up mode
- Manual DCD calibration during user mode

DCD calibration is performed automatically after device configuration and before entering user mode if the transceiver channels connected have the **Calibrate duty cycle during power up** option enabled. You can optionally trigger DCD calibration manually during user mode if:

- You reconfigure the transceiver from lower data rates to higher data rates (≥ 4.9152 Gbps)
- You perform clock network switching (TX PLL switching) and switch to data rates of ≥ 4915.2 Mbps

You do not have to enable DCD calibration if the transceivers are operating below 4.9152 Gbps.

Note: If you want to enable the DCD calibration IP for transceiver channels on the left and right sides of the device, use at least one Transceiver Reconfiguration Controller per side of the device. This applies to both power-up and manual DCD modes.

When DCD calibration and offset cancellation are enabled, the `reconfig_busy` status signal from the reconfiguration controller is deasserted to indicate the completion of both processes. If DCD calibration is not enabled, the deassertion of `reconfig_busy` signal indicates the completion of the offset cancellation process.

Note: Arria V GZ devices do not require DCD calibration.

Related Information

- **AN 676: Using the Transceiver Reconfiguration Controller for Dynamic Reconfiguration in Arria V and Cyclone V Devices**
- **Altera Transceiver PHY IP Core User Guide**

PMA Analog Controls Reconfiguration

You can dynamically reconfigure the analog controls setting after offset cancellation is complete and the reset sequence is performed. You can continue with the subsequent reconfigurations of the analog controls when the `reconfig_busy` status signal is low. A high on the `reconfig_busy` signal indicates that the reconfiguration operation is in progress.

You can reconfigure the following transceiver analog controls:

- Transmitter pre-emphasis
- Differential output voltage (V_{OD})
- Receiver equalizer control
- Direct-current (DC) gain settings

To reconfigure the analog control settings, perform read and write operations to the PMA analog settings reconfiguration control IP within the reconfiguration controller.

Related Information

Altera Transceiver PHY IP Core User Guide

For information about the read and write operations with the reconfiguration controller

Dynamic Reconfiguration of Loopback Modes

You can enable the pre- and post-CDR reverse serial loopback modes by writing the appropriate bits of the Transceiver Reconfiguration Controller.

The following loopback paths are available:

- **Post-CDR reverse serial loopback path**— The RX captures the input data and feeds it into the CDR. The recovered data from the CDR output feeds into the TX driver and sends to the TX pins through the TX driver. For this path, the RX and CDR can be tested. For this path, the TX driver can be programmed to use either the main tap only or the main tap and the pre-emphasis first post-tap. Enabling or disabling the post-CDR reverse serial loopback modes is done through the PMA Analog Reconfiguration IP in the Transceiver Reconfiguration PHY IP.
- **Pre-CDR reverse serial loopback path**— The RX captures the input data and feeds it back to the TX driver through a buffer. With this path, you can perform a quick check for the quality of the RX and TX buffers. Enabling or disabling the pre-CDR reverse serial loopback mode.

Note: Serial loopback can be implemented with the transceiver PHY IP directly using the Avalon interface or a control port.

Related Information

Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide

Transceiver PLL Reconfiguration

You can use the PLL reconfiguration registers to switch the reference clock input to the TX PLL or the clock data recovery (CDR) circuitry.

For example, you can switch the reference clock from 100 MHz to 125 MHz. You can also change the data rate from 2.5 Gbps to 5 Gbps by reconfiguring the transmitter PLL connected to the transceiver channel.

Note: Reference clock switching is only supported on the dedicated REFCLK pin.

The Transceiver Reconfiguration PHY IP provides an Avalon ® -MM user interface to perform PLL reconfiguration.

Related Information

"PLL Reconfiguration" section in the Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide
For information about performing PLL reconfiguration.

Transceiver Channel Reconfiguration

You can use channel reconfiguration to dynamically reconfigure the channel in a transceiver PHY IP core. Among the settings that you can change dynamically are the data rate and interface width.

You can reconfigure the channels in the following ways:

- Reconfigure the CDR of the receiver channel.
- Enable and disable all static PCS sub-blocks.
- Select an alternate PLL within the transceiver block to supply a different clock to the transceiver clock generation block.
- Reconfigure the TX local clock divider with a 1, 2, 4, or 8 division factor.

Every transmitter channel has a clock divider. When you reconfigure these clock dividers, ensure that the functional mode of the transceiver channel supports the reconfigured data rate. For example, you can change the data rate from 6.25 Gbps to 3.125 Gbps by reconfiguring the clock divider.

Transceiver Interface Reconfiguration

You can reconfigure the transceiver interfaces by reconfiguring the FPGA fabric transceiver channel data width that includes PCS-PLD and PMA-PCS interfaces.

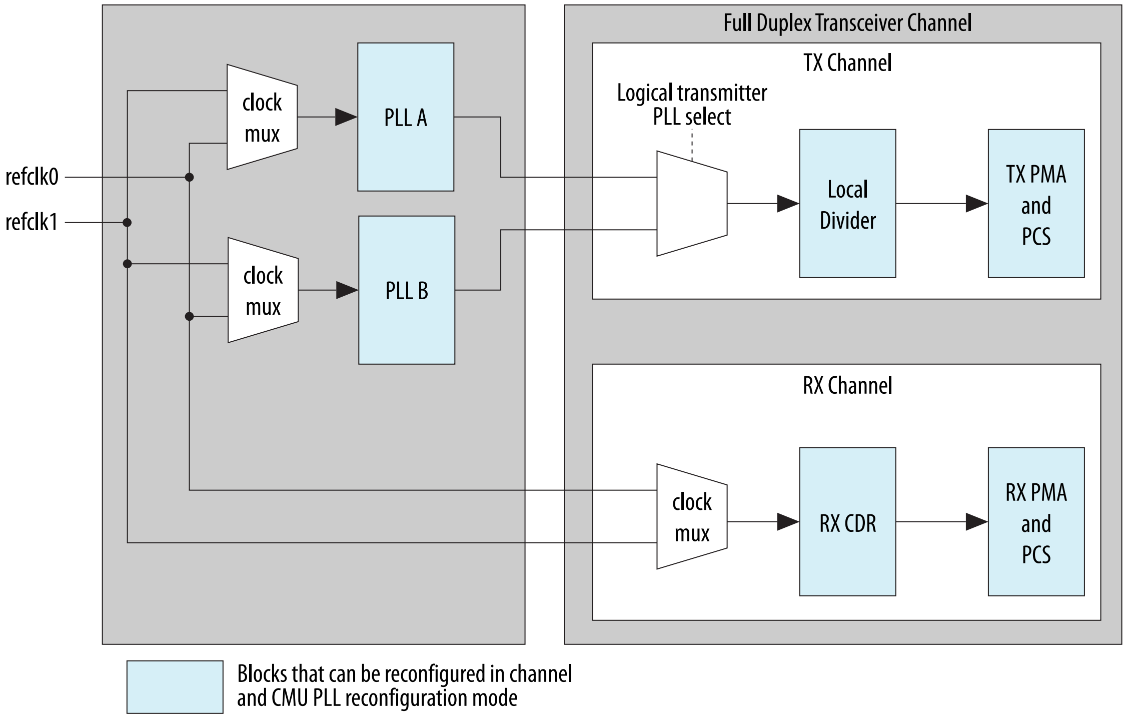
For example, you can reconfigure the custom PHY IP to enable or disable the 8B/10B encoder/decoder. There is no limit to the number of functional modes you can reconfigure the transceiver channel to if the various clocks involved support the transition. When you switch the custom PHY IP from one function mode to a different function mode, you may need to reconfigure the FPGA fabric-transceiver channel data width, enable or disable PCS sub-blocks, or both, to comply with the protocol requirements.

Channel reconfiguration only affects the channel involved in the reconfiguration (the transceiver channel specified by the unique logical channel address), without affecting the remaining transceiver channels controlled by the same Transceiver Reconfiguration Controller. PLL reconfiguration affects all channels that are currently using that PLL for transmission.

Channel reconfiguration from either a transmitter-only configuration to a receiver-only configuration or vice versa is not allowed.

Figure 8-1: Transceiver Channel and PLL Reconfiguration in a Transceiver Block

The following figure shows the functional blocks you dynamically reconfigure using transceiver channel and PLL reconfiguration mode.


Full Duplex Transceiver Channel
TX Channel
Logical transmitter PLL select
Local Divider
TX PMA and PCS
RX Channel
clock mux
RX CDR
RX PMA and PCS
refclk0
refclk1
clock mux
PLL A
clock mux
PLL B
Blocks that can be reconfigured in channel and CMU PLL reconfiguration mode

Related Information

“Channel and PLL Reconfiguration” section in the Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide

For information about transceiver channel and PLL reconfiguration.

Reduced .mif Reconfiguration

Reduce reconfiguration time by reconfiguring only the affected blocks in the transceiver channels.

This reconfiguration mode affects only the modified settings of the channel to reduce reconfiguration time significantly. For example, in SATA/SAS applications, auto-rate negotiation must be completed within a short period of time to meet the protocol specification. The reduced **.mif** method helps to reconfigure the channel to meet these specifications. You can generate the reduced **.mif** files manually or by using the **xcvr_diffmifgen.exe** utility.

Related Information

Altera Transceiver PHY IP Core User Guide

For information about reduced **.mif** creation.

On-Chip Signal Quality Monitoring (Eye Viewer)

The bit error rate (BER) eye contour can be used to measure the quality of the received data. Eye Viewer is a debug and diagnostic tool that analyzes the received data recovery path, including the receiver's gain, noise level, and recovery clock jitter. Eye Viewer can also measure vertical eye height, effectively allowing a BER eye contour to be plotted.

Eye Viewer is supported by Arria V GZ devices and not Arria V GX and GT devices. .

Eye Viewer uses a phase interpolator (PI) and sampler (SMP) to estimate the horizontal eye opening. Controlled by a logic generator, the PI generates a sampling clock and the SMP samples the data from the receiver output. The SMP outputs parallel data that is monitored for CRC or BER errors. When the PI output clock phase is shifted by small increments, the data error rate goes from high to low to high if the receiver is good. The number of steps of valid data is defined as the width of the eye. If none of the steps yield valid data, the width of the eye is equal to 0, which means the eye is closed.

The Transceiver Reconfiguration Controller provides an Avalon-MM user interface to enable the Eye Viewer feature.

Related Information

Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide.

For information about enabling the Eye Viewer feature.

Adaptive Equalization

Adaptive equalization (AEQ) solves issues related to changing data rates and backplane losses.

High-speed interface systems require different equalization settings to compensate for changing data rates and backplane losses. Manual tuning of the receiver channel equalization stages involves finding the optimal settings through trial and error, and then locking in those values during compilation. This manual static method is cumbersome and inefficient when system characteristics vary. The AEQ automatically tunes an active receiver channel equalization filter based on a frequency content comparison between the incoming signals and the internally generated reference signals.

Adaptive equalization is supported by Arria V GZ devices and not Arria V GX and GT devices.

The Transceiver Reconfiguration Controller provides an Avalon-MM user interface to enable the AEQ feature.

Related Information

"AEQ" section in the Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide
For information about enabling different options and using them to control the AEQ hardware.

Decision Feedback Equalization

Decision feedback equalization (DFE) helps compensate for backplane attenuation because of insufficient bandwidth.

DFE works by estimating the intersymbol interference (ISI) that is imposed by the channel on an incoming bit and canceling out the ISI as that bit is sampled by the CDR circuitry. The advantage of DFE is that it boosts the power of the highest frequency component of the received data without increasing its noise power. Use DFE in conjunction with the transmitter pre-emphasis and receiver linear equalization.

Decision feedback equalization is supported by Arria V GZ devices and not Arria V GX and GT devices.

Related Information

“DFE” section in the Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide
For more information about DFE.

Unsupported Reconfiguration Modes

The following reconfiguration modes are not supported:

- Switching between a receiver-only channel and a transmitter-only channel
- Switching between bonded to non-bonded mode or bonded mode with different xN lanes count (for example, switching from bonded x2 to bonded x4)
- Switching between one PHY IP to another PHY IP (for example Deterministic Latency PHY IP to Custom PHY IP)
- Switching between PMA Direct modes to non PMA Direct mode
- TX PLL reconfiguration is not supported if the TX PLL is connected to bonded channels

You can achieve PHY to PHY IP reconfiguration only if you use the Native PHY IP to configure your transceiver. For example, if you use the Native PHY IP to configure the SDI mode and a custom proprietary IP mode (also configured using Native PHY IP), you can reconfigure these two modes within the Native PHY IP. The switching refers to enabling and disabling the PCS sub-blocks for both SDI and the custom proprietary IP mode.

Document Revision History

Date	Version	Changes
April 2019	2019.04.08	Rebranded the following: • EyeQ to Eye Viewer
January 2016	2016.01.08	Removed mgmt_clk_clk statement from "Offset Cancellation" section.
September 2014	2014.09.30	Added FPGA fabric to transceiver channel interface width reconfiguration feature in *Table: Dynamic Reconfiguration Features.*
May 2013	2013.05.06	• Updated TX DCD calibration information • Included link AN 661 for fPLL reconfiguration • Added link to the known document issues in the Knowledge Base
November 2012	2012.11.19	• Rewritten and reorganized content, and updated template • Updated Transceiver Interface Reconfiguration • Added Reduced **.mif** Reconfiguration • Added On-Chip Signal Quality Monitoring (Eye Viewer) for Arria GZ devices • Adaptive Equalization for Arria GZ devices • Decision Feedback Equalization for Arria GZ devices • Listed unsupported features
June 2012	1.2	• Added “Transceiver PLL Reconfiguration” and “Transceiver Channel and Interface Reconfiguration” sections. • Updated the “Offset Cancellation and TX Duty Cycle Distortion Calibration” section • Updated Table 7–1

Date	Version	Changes
November 2011	1.1	• Added Table 7–1 • Deleted “Transceiver Reconfiguration Controller” and “Channel and PLL Reconfiguration” sections. • Updated “Offset Cancellation” and “PMA Analog Settings Reconfiguration” sections • Added “Enabling and Disabling Loopback Modes” section.