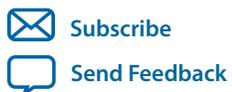


Stratix V Device Handbook

Volume 2: Transceivers



SV5V2
2017.02.15

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



Contents

Transceiver Architecture in Stratix V Devices.....	1-1
Device Layout.....	1-3
Stratix V GX/GT Channel and PCIe Hard IP Layout.....	1-4
Stratix V GS Channel and PCIe Hard IP Layout.....	1-5
PMA Architecture.....	1-9
Receiver Buffer.....	1-10
Receiver Clock Data Recovery Unit.....	1-13
Receiver Deserializer.....	1-15
Transmitter PLLs.....	1-17
Transmitter Serializer.....	1-21
Transmitter Buffer.....	1-22
Transceiver Calibration Blocks.....	1-24
PMA Reconfiguration.....	1-27
Standard PCS Architecture.....	1-27
Receiver Standard PCS Datapath.....	1-29
Transmitter Standard PCS Datapath.....	1-41
10G PCS Architecture.....	1-45
Receiver 10G PCS Datapath.....	1-46
Transmitter 10G PCS Datapath.....	1-50
PCIe Gen3 PCS Architecture.....	1-55
Receiver PCIe Gen3 PCS Datapath.....	1-56
Transmitter PCIe Gen3 PCS Datapath.....	1-57
PIPE Interface.....	1-58
Document Revision History.....	1-59
Transceiver Clocking in Stratix V Devices.....	2-1
Input Reference Clocking.....	2-1
Input Reference Clock Sources.....	2-2
Internal Clocking.....	2-7
Transmitter Clock Network.....	2-8
Transmitter Clocking.....	2-18
Receiver Clocking.....	2-28
FPGA Fabric-Transceiver Interface Clocking.....	2-34
Transmitter Datapath Interface Clocking.....	2-37
Receiver Datapath Interface Clock.....	2-42
GXB 0 PPM Core Clock Assignment.....	2-46
Document Revision History.....	2-47
Transceiver Reset Control in Stratix V Devices.....	3-1
PHY IP Embedded Reset Controller.....	3-1
Embedded Reset Controller Signals.....	3-2

Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Power-Up.....	3-3
Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Operation.....	3-4
User-Coded Reset Controller.....	3-5
User-Coded Reset Controller Signals.....	3-5
Resetting the Transmitter with the User-Coded Reset Controller During Device Power-Up	3-7
Resetting the Transmitter with the User-Coded Reset Controller During Device Operation.....	3-8
Resetting the Receiver with the User-Coded Reset Controller During Device Power-Up Configuration.....	3-9
Resetting the Receiver with the User-Coded Reset Controller During Device Operation..	3-10
Transceiver Reset Using Avalon Memory Map Registers.....	3-11
Transceiver Reset Control Signals Using Avalon Memory Map Registers.....	3-11
Clock Data Recovery in Manual Lock Mode.....	3-12
Control Settings for CDR Manual Lock Mode.....	3-12
Resetting the Transceiver in CDR Manual Lock Mode.....	3-13
Transceiver Blocks Affected by the Reset and Powerdown Signals.....	3-14
Document Revision History.....	3-16

Transceiver Configurations in Stratix V Devices..... 4-1

Protocols and Transceiver PHY IP Support.....	4-1
10GBASE-R and 10GBASE-KR.....	4-5
10GBASE-R and 10GBASE-KR Transceiver Datapath Configuration.....	4-7
10GBASE-R and 10GBASE-KR Supported Features.....	4-11
1000BASE-X and 1000BASE-KX Transceiver Datapath.....	4-14
1000BASE-X and 1000BASE-KX Supported Features.....	4-14
Synchronization State Machine Parameters in 1000BASE-X and 1000BASE-KX Configurations.....	4-17
Transceiver Clocking in 10GBASE-R, 10GBASE-KR, 1000BASE-X, and 1000BASE-KX Configurations.....	4-17
Interlaken.....	4-18
Transceiver Datapath Configuration.....	4-19
Supported Features.....	4-20
Transceiver Clocking.....	4-23
PCI Express (PCIe)—Gen1, Gen2, and Gen3.....	4-25
Transceiver Datapath Configuration.....	4-25
Supported Features for PCIe Configurations.....	4-29
Supported Features for PCIe Gen3.....	4-32
Transceiver Clocking and Channel Placement Guidelines.....	4-35
Advanced Channel Placement Guidelines for PIPE Configurations.....	4-45
Transceiver Clocking for PCIe Gen3.....	4-53
XAUI.....	4-59
Transceiver Datapath in a XAUI Configuration.....	4-60
Supported Features.....	4-62
Transceiver Clocking and Channel Placement Guidelines.....	4-65
CPRI and OBSAI—Deterministic Latency Protocols.....	4-66

Transceiver Datapath Configuration.....	4-67
Phase Compensation FIFO in Register Mode.....	4-68
Channel PLL Feedback.....	4-68
CPRI and OBSAI.....	4-68
CPRI Enhancements.....	4-71
Transceiver Configurations.....	4-71
Standard PCS Configurations—Custom Datapath.....	4-71
Standard PCS Configurations—Low Latency Datapath.....	4-77
Transceiver Channel Placement Guidelines.....	4-82
10G PCS Configurations.....	4-83
Merging Instances.....	4-90
Native PHY IP Configuration.....	4-91
Native PHY Transceiver Datapath Configuration.....	4-92
Standard PCS Features.....	4-93
10G PCS Supported Features.....	4-94
10G Datapath Configurations with Native PHY IP.....	4-96
PMA Direct Supported Features.....	4-99
Channel and PCS Datapath Dynamic Switching Reconfiguration.....	4-99
Stratix V GT Device Configurations.....	4-99
Document Revision History.....	4-100
Transceiver Loopback Support in Stratix V Devices.....	5-1
Serial Loopback.....	5-1
PIPE Reverse Parallel Loopback.....	5-2
Reverse Serial Loopback.....	5-3
Reverse Serial Pre-CDR Loopback.....	5-4
Document Revision History.....	5-5
Dynamic Reconfiguration in Stratix V Devices.....	6-1
Dynamic Reconfiguration Features.....	6-1
Offset Cancellation.....	6-2
PMA Analog Controls Reconfiguration.....	6-2
On-Chip Signal Quality Monitoring ().....	6-3
Decision Feedback Equalization.....	6-3
Adaptive Equalization.....	6-4
Dynamic Reconfiguration of Loopback Modes.....	6-4
Transceiver PLL Reconfiguration	6-5
Transceiver Channel Reconfiguration.....	6-5
Transceiver Interface Reconfiguration	6-5
Document Revision History.....	6-6

Transceiver Architecture in Stratix V Devices

1

2016.01.11

SV52002



Subscribe



Send Feedback

For a complete understanding of Stratix® V transceivers, first review the transceiver architecture chapter, then refer to the subsequent chapters in this volume.

You can implement Stratix V transceivers using Altera's transceiver intellectual property (IP) which are part of the Quartus® II software.

Stratix V devices provide up to 66 backplane-capable full-duplex clock data recovery (CDR)-based transceivers.

Table 1-1: Device Variants

Stratix Device	Channel Type	
	GX	GT
GS	600 Mbps to 14.1 Gbps	Not supported
GX	600 Mbps to 14.1 Gbps	Not supported
GT	600 Mbps to 12.5 Gbps	19.6 Gbps to 28.05 Gbps

Stratix V transceivers are divided into two blocks: physical medium attachment (PMA) and physical coding sublayer (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. The PCS block contains the digital processing interface between the PMA and FPGA core. There are three types of PCS blocks in Stratix V devices: a standard PCS block, a 10G PCS, and a PCIe Gen3 PCS that supports the PCIe Gen3 Base specification.

Stratix V transceivers are structured into full-duplex (transmitter and receiver) six-channel groups called transceiver blocks.

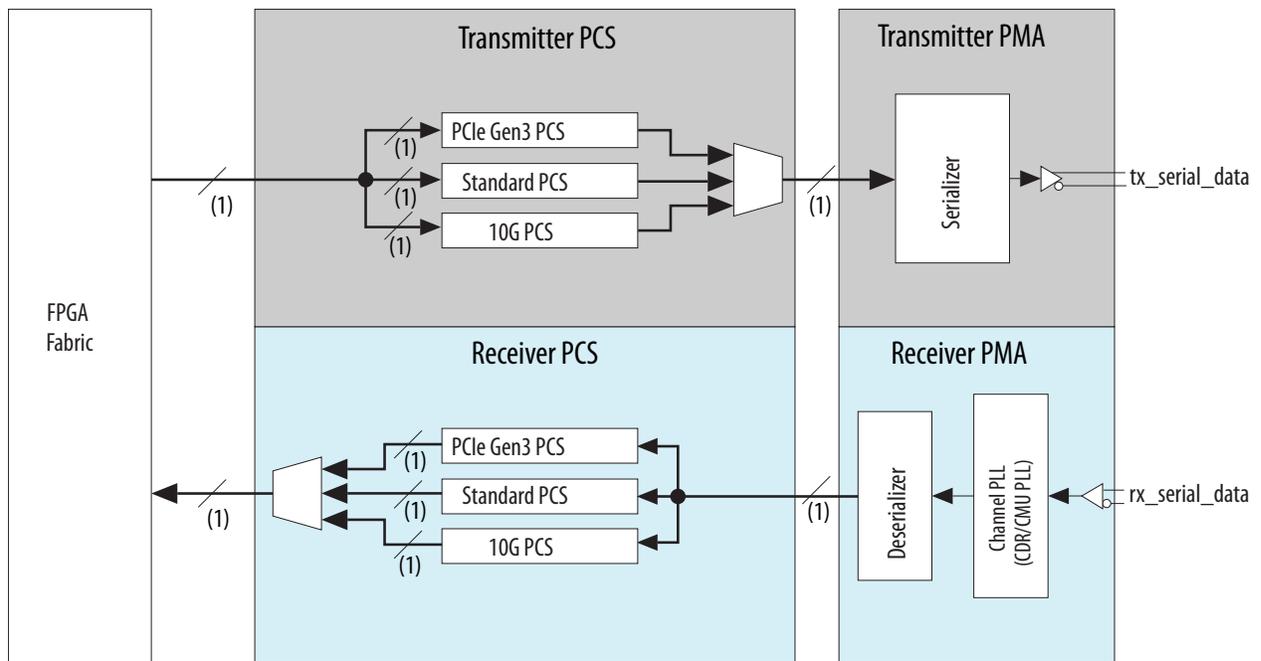
Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

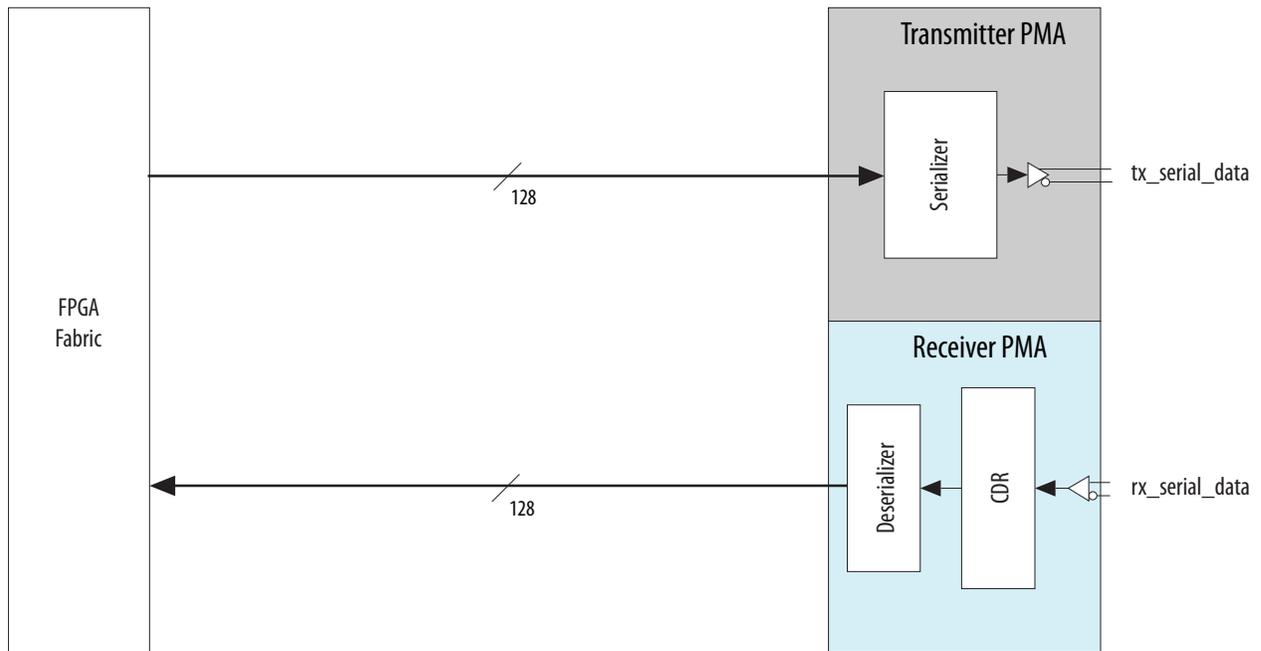
Figure 1-1: Single Full-Duplex GX Channel



Note:
1. The PMA and PCS widths are configurable.

Figure 1-2: Single Full-Duplex GT Channel

The GT channels do not have PCS.



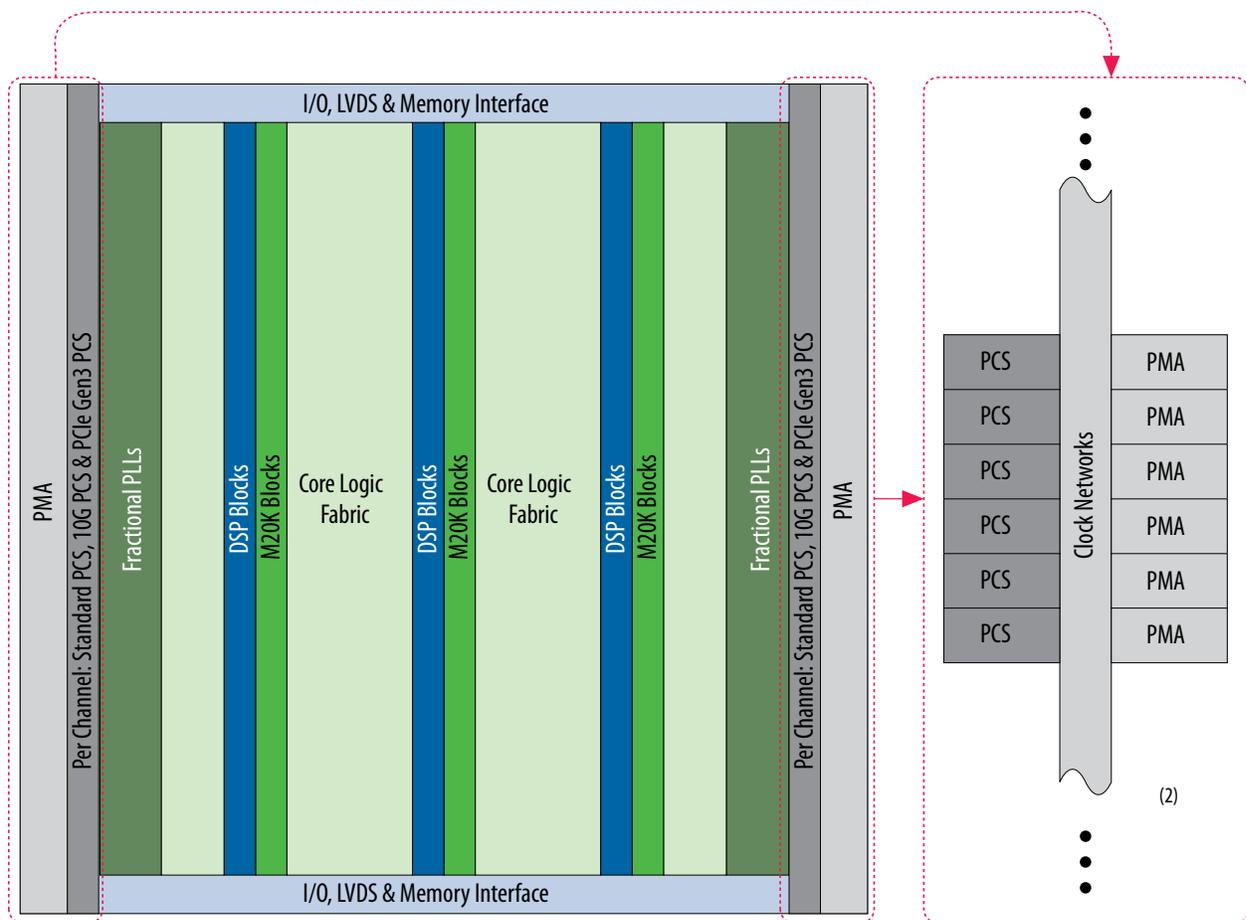
Related Information

- **Stratix V Device Handbook: Known Issues**
Lists the planned updates to the *Stratix V Device Handbook* chapters.
- **For details about using transceiver IPs, refer to the Altera Transceiver PHY IP Core User Guide.**
- **For transceiver performance specifications, refer to the Stratix V Device Datasheet.**
- **For information about features that will be supported in a future release of the Quartus® II software, refer to the Upcoming Stratix V Device Features document.**

Device Layout

Stratix V devices have columns of transceivers on the left and right sides of the devices. However, some Stratix V devices have columns of transceivers on the left side of the devices only.

Figure 1-3: Basic Layout of Transceivers



Notes:

1. This figure represents a given variant of a Stratix V device with transceivers. Other variants may have a different floor plan than the one shown here.
2. You can use the unused transceiver channels as additional transceiver transmitter PLLs.

The location of the transceiver bank boundaries are important for clocking resources, bonding channels, and fitting. The transceivers are grouped in transceiver banks of three and six channels.

Within a transceiver bank the lowest numbered pin name is channel 0 of that bank and the highest numbered pin name of the bank is channel 5.

Stratix V GX/GT Channel and PCIe Hard IP Layout

Stratix V devices have many different channel and PCIe Hard IP variants.

Figure 1-4: General Transceiver Bank Locations for GX Devices

This figure shows the layout for a 66 channel device. Some devices have fewer channels.

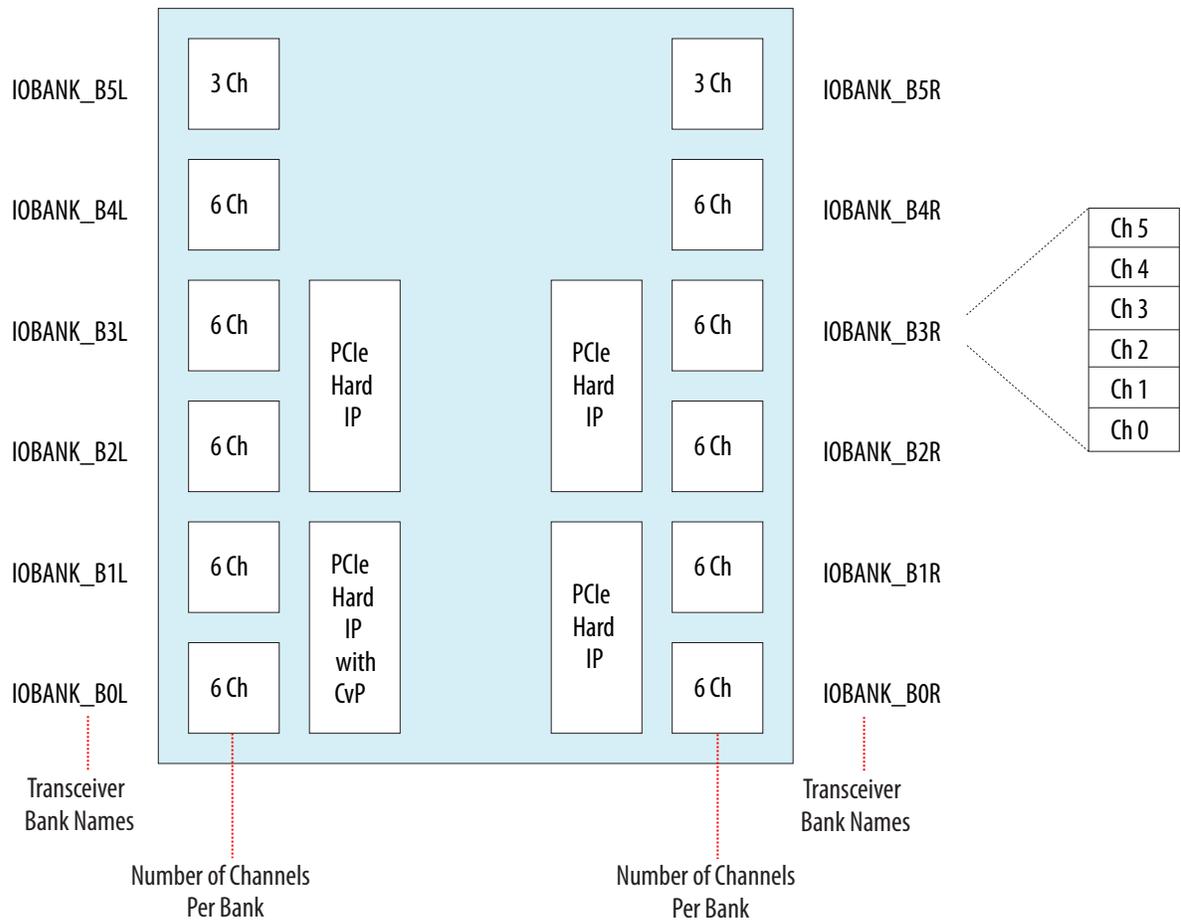
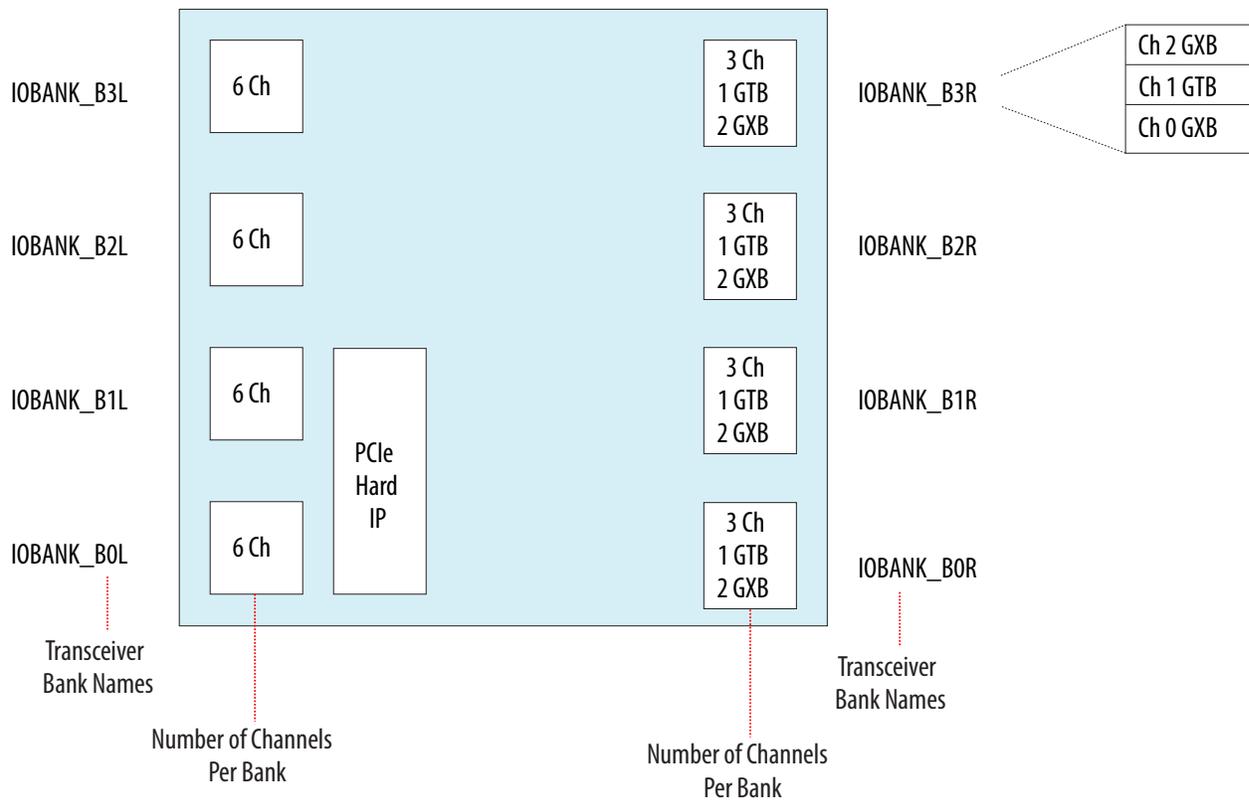


Figure 1-5: General Transceiver Bank Locations for GT Devices



Notes:

1. GT transceiver banks are made up of 1 GT channel and 2 GX channels. The GT channel is the middle channel in the bank.
2. GT devices only come with one PCIe HIP block located across GX banks L0 and L1.

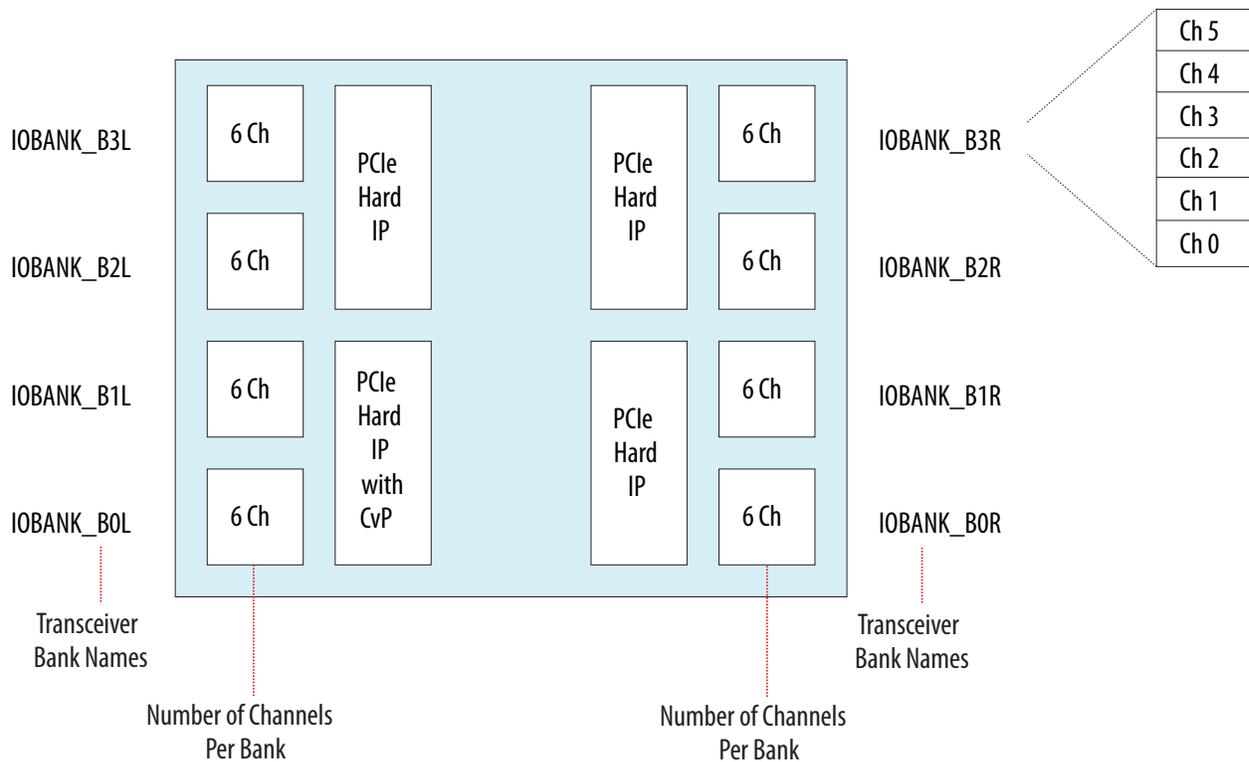
Related Information

- [For more information about device pin-outs, refer to Pin-Out Files for Altera Devices.](#)
- [For more information about device options, refer to the Stratix V Device Overview.](#)

Stratix V GS Channel and PCIe Hard IP Layout

Stratix V devices have many different channel and PCIe Hard IP variants.

Figure 1-6: General Transceiver Bank Locations for GS Devices



Channel Variants

GS/GX Devices

- 12-channel devices use banks B0L and B1L
- 24-channel devices use banks B0L, B1L, B0R and B1R
- 36-channel devices use banks B0L, B1L, B2L, B0R, B1R, and B2R
- 48-channel devices use banks B0L, B1L, B2L, B3L, B0R, B1R, B2R, and B3R
- 66-channel devices use all banks (GX only)

GT Devices

These 36-channel devices use banks B0L, B1L, B2L, B3L, B0R, B1R, B2R, and B3R.

GS/GT/GX Device Variants and Packages

Table 1-2: Stratix V GS, GT, and GX Device Variant Packages

Device Variant	PCIe Hard IP Blocks	# of Transceivers	Package	Side of Device with Transceivers
5SGSD3	1	12	EH29	Left
	1	24	HF35	Left and Right

Device Variant	PCIe Hard IP Blocks	# of Transceivers	Package	Side of Device with Transceivers
5SGSD4	1	12	EH29	Left
	1	24	HF35	
	1	36	KF40	
5SGSD5	1	24	HF35	Left and Right
	1	36	KF40	
5SGSD6	1 or 2	36	KF40	
	1 or 4	48	NF45	
5SGSD8	1 or 2	36	KF40	
	1 or 4	48	NF45	
5SGTC5	1	36	KF40	
5SGTC7				

Device Variant	PCIe Hard IP Blocks	# of Transceivers	Package	Side of Device with Transceivers
5SGXA3	1	12	EH29	Left
	1 or 2	24	HF35	
		36	KF35	
			KF40	
5SGXA4	1 or 2	24	HF35	
		36	KF35	
			KF40	
5SGXA5	1 or 2	24	HF35	
		36	KF35	
			KF40	
	1 or 4	48	NF40	
			NF45	
5SGXA7	1 or 2	24	HF35	
		36	KF35	
			KF40	
	1 or 4	48	NF40	
			NF45	
5SGXA9	1 or 2	36	KH40	
	1 or 4	48	NF45	
5SGXAB	1 or 2	36	KF40	
	1 or 4	48	NF45	
5SGXB5	1 or 4	66	RF40	
			RF43	
5SGXB6	1 or 4	66	RF40	
			RF43	
5SGXB9	1 or 4	66	RH43	
5SGXBB	1 or 4	66	RH43	
				Left and Right

PCIe Hard IP Variants

- 1 PCIe Hard IP variant has a Hard IP block located across GX banks L0 and L1
- 2 PCIe Hard IP variant has Hard IP blocks located across GX banks L0 and L1 and GX banks R0 and R1
- 4 PCIe Hard IP variant has Hard IP blocks located across GX banks L0 and L1, GX banks L2 and L3, GX banks R0 and R1 and GX banks R2 and R3

Related Information

- [For more information about PCIe Hard IP connections for GS and GX devices, refer to Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
- [For more information about PCIe Hard IP connections for GT devices, refer to Stratix V GT Device Family Pin Connection Guidelines](#)

PMA Architecture

The PMA receives and transmits off-chip high-speed serial data streams.

Figure 1-7: Receiver PMA Block Diagram

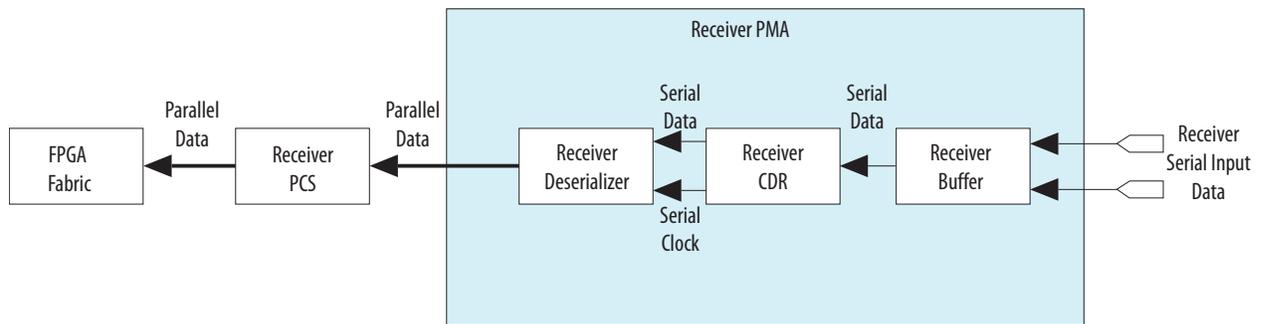
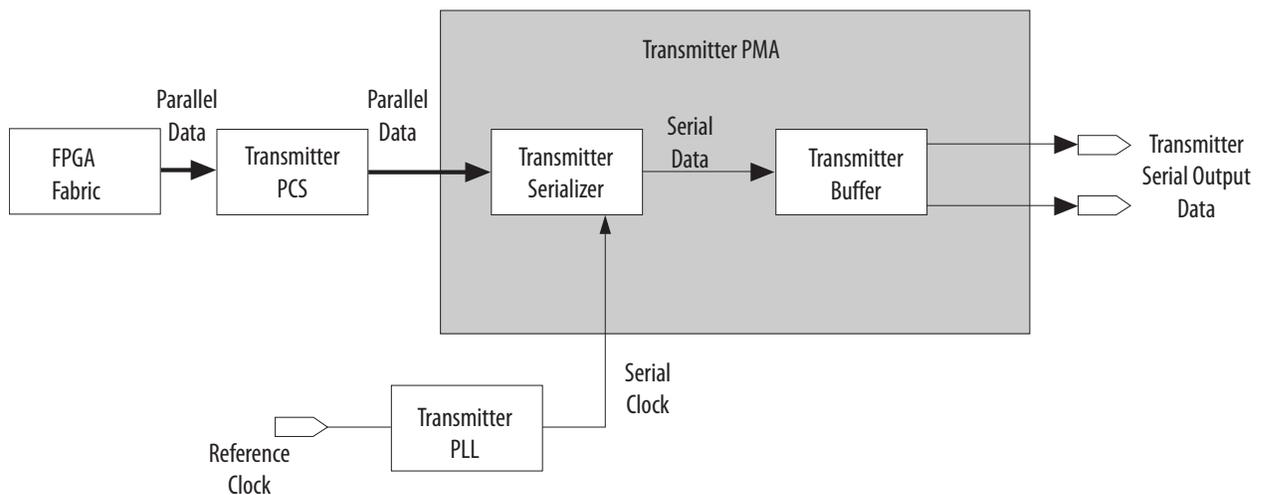


Figure 1-8: Transmitter PMA Block Diagram



Related Information

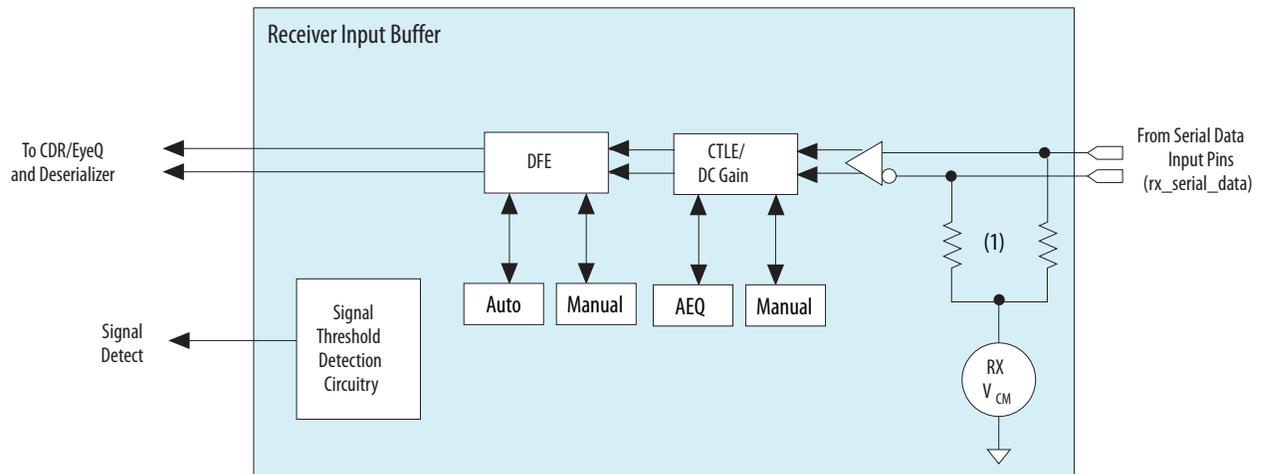
For additional details about clocking, refer to the [Transceiver Clocking in Stratix V Devices](#) chapter.

Receiver Buffer

The receiver input buffer receives serial data from the `rx_serial_data` port and feeds the serial data to the CDR and deserializer.

Figure 1-9: Receiver Input Buffer

Stratix V GT channels do not support AEQ and DFE.

**Note:**

1. For Stratix V GX channels, the available options are OFF, 85, 100, 120, and 150 Ω . Stratix V GT channels only support 100 Ω .

The receiver buffer supports the following features:

- Programmable Equalizer Bandwidth
- Programmable differential On-Chip Termination (OCT)
- Programmable V_{CM}
- AC and DC coupling
- Signal Threshold Detection Circuitry
- Continuous Time Linear Equalization (CTLE)
- DC Gain
- Decision Feedback Equalization (DFE)
- EyeQ

You can program these features using the assignment editor and a reconfiguration controller.

Related Information

Refer to the ["Transceiver Performance Specifications"](#) section of the [Stratix V Device Datasheet](#) for receiver buffer electrical specifications.

Receiver Equalizer Gain Bandwidth

Stratix V GX channels have two receiver equalizer gain bandwidth modes: half-bandwidth and full-bandwidth. When the data rate is ≤ 6.25 Gbps, Altera recommends selecting half bandwidth setting. When the data rate is between 6.25 Gbps and 14.1 Gbps, Altera recommends the full bandwidth setting. You can select the mode in the Assignment Editor of the Quartus II software (Receiver Equalizer Gain Bandwidth Select).

Stratix V GT channels support full-bandwidth mode only.

Related Information

[For more information about the receiver equalizer gain bandwidth modes, refer to the Stratix V Device Datasheet](#)

Programmable Differential On-Chip Termination (OCT)

The receiver buffer supports differential OCT resistances of 85, 100, 120, 150 Ω and OFF for GX channels. The receiver buffer of GT channels supports 100 Ω differential termination only.

Note: The receiver OCT resistors have calibration support to compensate for process, voltage, and temperature (PVT) variations. This does not apply to GT devices.

Programmable V_{CM}

The receiver buffer has on-chip biasing circuitry to establish the required V_{CM} at the receiver input.

The Quartus II software automatically chooses the best V_{CM} setting.

Note: On-chip biasing circuitry is available only if you select one of the **Termination logic options** to configure OCT. If you select external termination, you must implement off-chip biasing circuitry to establish the V_{CM} at the receiver input buffer.

Related Information

[For more information, refer to the Stratix V Device Datasheet.](#)

Signal Threshold Detection Circuitry

You can enable the optional signal threshold detection circuitry. If enabled, this option senses whether the signal level present at the receiver input buffer is above the signal detect threshold voltage that you specified. This option is not available for all transceiver PHYs or for GT channels.

Related Information

- [For more information about transceiver configurations, refer to Transceiver Configurations in Stratix V Devices.](#)
- [For more information about the signal threshold detect signal, refer to the Altera Transceiver PHY IP Core User Guide.](#)

DC Gain

DC gain circuitry provides an equal boost to the incoming signal across the frequency spectrum. The receiver buffer supports DC gain settings of up to 8 dB.

Related Information

[For more information about the DC gain settings for the receiver buffer, refer to the Stratix 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. For Stratix V GX channels, the equalization circuitry provides up to 16 dB of high-frequency boost. For Stratix V GT channels, the equalization circuitry provides up to 15 dB of high-frequency boost. The CTLE operates in two modes: manual mode and adaptive equalization (AEQ) mode.

You can dynamically switch between these modes.

Note: The CTLE cannot be bypassed.

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 Stratix 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 mode is not supported in GT channels.

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 change.
- Powerdown mode—The AEQ of the specific channel is placed in standby mode and the CTLE uses the manually set value.

Related Information

- [For more information about the CTLE specifications, refer to the Stratix V Device Datasheet.](#)
- [For more information about enabling different options and using them to control the AEQ hardware, refer to the "Transceiver Reconfiguration Controller IP Core" chapter of the Altera Transceiver PHY IP Core User Guide.](#)

Decision Feedback Equalization

The decision feedback equalization (DFE) 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 three DFE modes: manual, auto-adaptation, and triggered.

Note: The DFE is not supported in GT channels.

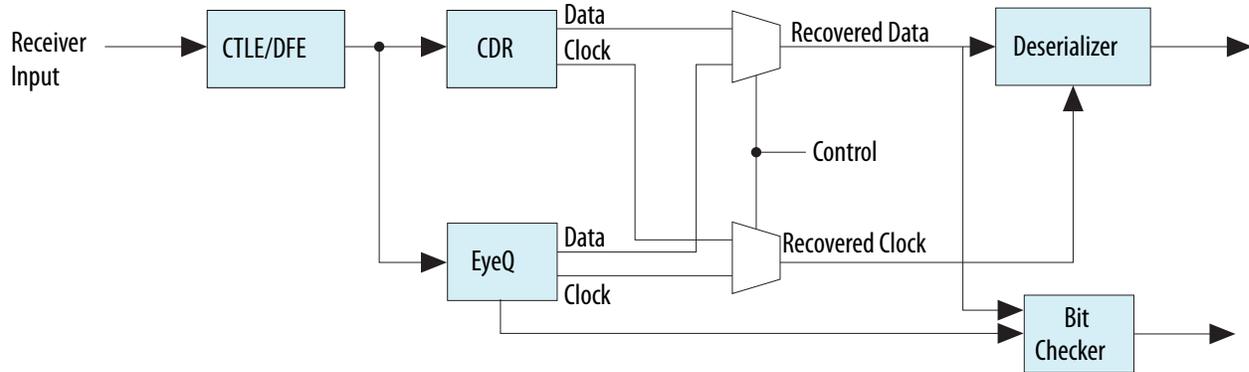
Related Information

For more information, refer to the "Transceiver Reconfiguration Controller IP Core" chapter of the [Altera Transceiver PHY IP Core User Guide](#).

EyeQ

The EyeQ feature is a debug and diagnosis tool that helps you analyze the received data by measuring the horizontal and vertical eye margin. There are two multiplexers, each of which selects one path for the clock and data to feed to the deserializer, respectively.

Figure 1-10: Receiver and EyeQ Architecture



Serial Bit Checker

The serial bit checker is available when you enable EyeQ in the transceiver reconfiguration controller. It provides a means of estimating the number of errors that occur between the captured CDR signal and the sampled EyeQ signal. The advantage of enabling this block, is that it allows bit error register (BER) monitoring over live traffic.

This feature is also available for PCIe configurations.

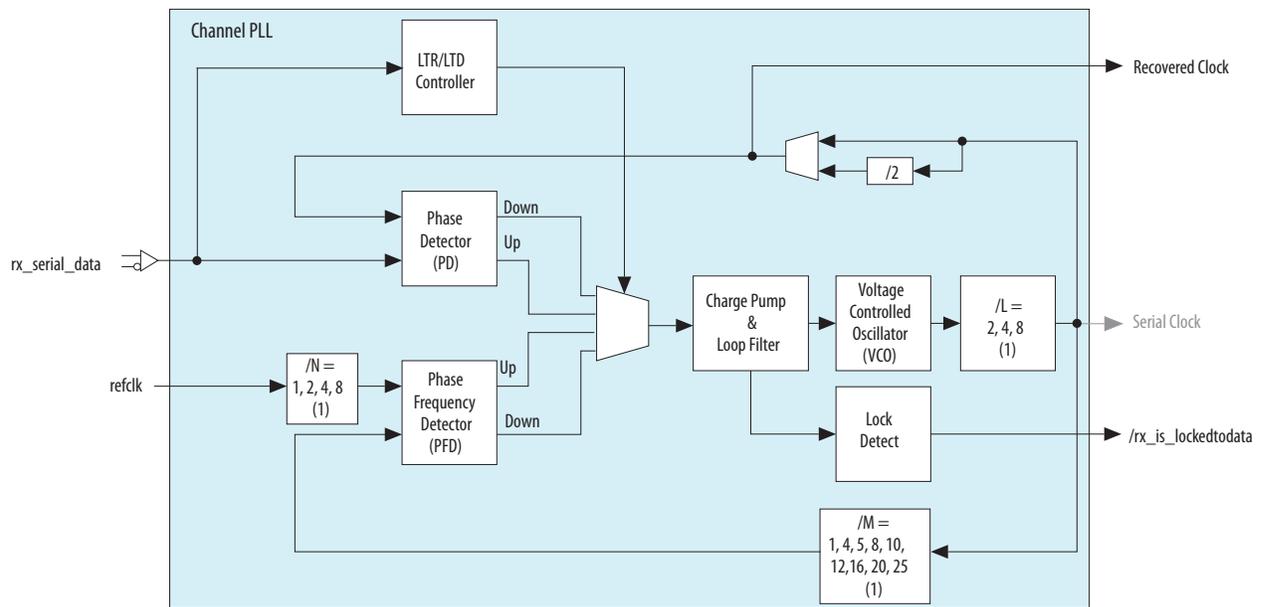
Related Information

For more information about implementing the serial bit checker, refer to the [Transceiver Reconfiguration Controller EyeQ Registers](#) section of the [Altera Transceiver PHY IP Core User Guide](#)

Receiver Clock Data Recovery Unit

The PMA of each channel includes a channel PLL that you can configure as a receiver clock data recovery (CDR) for the receiver, or a clock multiplier unit (CMU) PLL for the transmitter. For more information about using the channel PLL as a CMU PLL, refer to [Transmitter PLLs](#) on page 1-17.

Figure 1-11: Channel PLL Configured as CDR



Note:
1. Not all combinations of N , M , and L values are valid. The Quartus II software automatically chooses the optimal values.

Lock-to-Reference Mode

In LTR mode, the phase frequency detector (PFD) in the CDR tracks the receiver input reference clock. The PFD controls the charge pump that tunes the VCO in the CDR. The `pma_rx_is_lockedtoref` status signal is asserted active high to indicate that the CDR has locked to the phase and frequency of the receiver input reference clock.

Note: The phase detector (PD) is inactive in LTR mode.

Lock-to-Data Mode

During normal operation, the CDR must be in LTD mode to recover the clock from the incoming serial data. In LTD mode, the 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 VCO.

Note: The PFD is inactive in LTD mode. The `rx_is_lockedtoref` status signal toggles randomly and is not significant in LTD mode.

After switching to LTD mode, the `rx_is_lockedtodata` status signal is asserted. 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 `rx_is_lockedtodata` signal toggles until the CDR sees valid data; therefore, you should hold receiver PCS logic in reset (`rx_digitalreset`) for a minimum of 4 μ s after `rx_is_lockedtodata` remains continuously asserted.

CDR Lock Modes

You can configure the CDR in either automatic lock mode or manual lock mode. By default, the Quartus II software configures the CDR in automatic lock mode.

Automatic Lock Mode

In automatic lock mode, the CDR initially locks to the input reference clock (LTR mode). After the CDR locks to the input reference clock, the CDR locks to the incoming serial data (LTD mode) when the following conditions are met:

- The signal threshold detection circuitry indicates the presence of valid signal levels at the receiver input buffer.
- The CDR output clock is within the configured ppm frequency threshold setting with respect to the input reference clock (frequency locked).
- The CDR output clock and the input reference clock are phase matched within approximately 0.08 unit interval (UI) (phase locked).

If the CDR does not stay locked to data because of frequency drift or severe amplitude attenuation, the CDR switches back to LTR mode.

Manual Lock Mode

The PPM detector and phase relationship detector reaction times can be too long for some applications that require faster CDR lock time. You can manually control the CDR to reduce its lock time using two optional input ports (`rx_set_locktoref` and `rx_set_locktodata`).

Table 1-3: Relationship Between Optional Input Ports and the CDR Lock Mode

<code>rx_set_locktoref</code>	<code>rx_set_locktodata</code>	CDR Lock Mode
0	0	Automatic
1	0	Manual-RX CDR LTR
X	1	Manual-RX CDR LTD

Related Information

[For more information about manual lock mode, refer to the Transceiver Reset Control in Stratix V Devices chapter.](#)

Receiver Deserializer

The deserializer block clocks in serial input data from the receiver buffer using the high-speed serial recovered clock and deserializes the data using the low-speed parallel recovered clock. The deserializer forwards the deserialized data to the receiver PCS or FPGA core.

The deserializer in the RX PMA also provides a clock slip feature. The word aligner block in the PCS can contribute up to one parallel clock cycle of latency uncertainty. You can use the clock slip/bit slip feature to control the word alignment instead, so that you can reduce the latency uncertainty, ensuring deterministic latency. The deterministic latency state machine in the word aligner (within the PCS) automatically controls the clock slip/bit slip operation. The de-serializer first performs clock slip/bit slip, after which the

parallel data is word-aligned in the RX PCS. These features are for protocols like CPRI that require deterministic latency through the PHY layer.

The GX channel deserializer supports 8- and 10-bit, 16- and 20-bit, 32- and 40-bit factors. 64-bit factors are also supported depending on the transceiver configuration. Unlike the GX channel deserializer, the GT channel deserializer does not support programmable data widths and is fixed at 128 bits.

Receiver PMA Bit-Slip

The deserializer has a bit slip feature to enable the high speed serial bit slipping to minimize uncertainty in serialization process per common public radio interface (CPRI) requirement. The bit slip feature is useful for other protocols as well. You can enable this feature through the Custom or Native or Deterministic Latency PHY IP. When you enable this feature, the period of a receiver side parallel clock could be extended by 1 unit interval (UI).

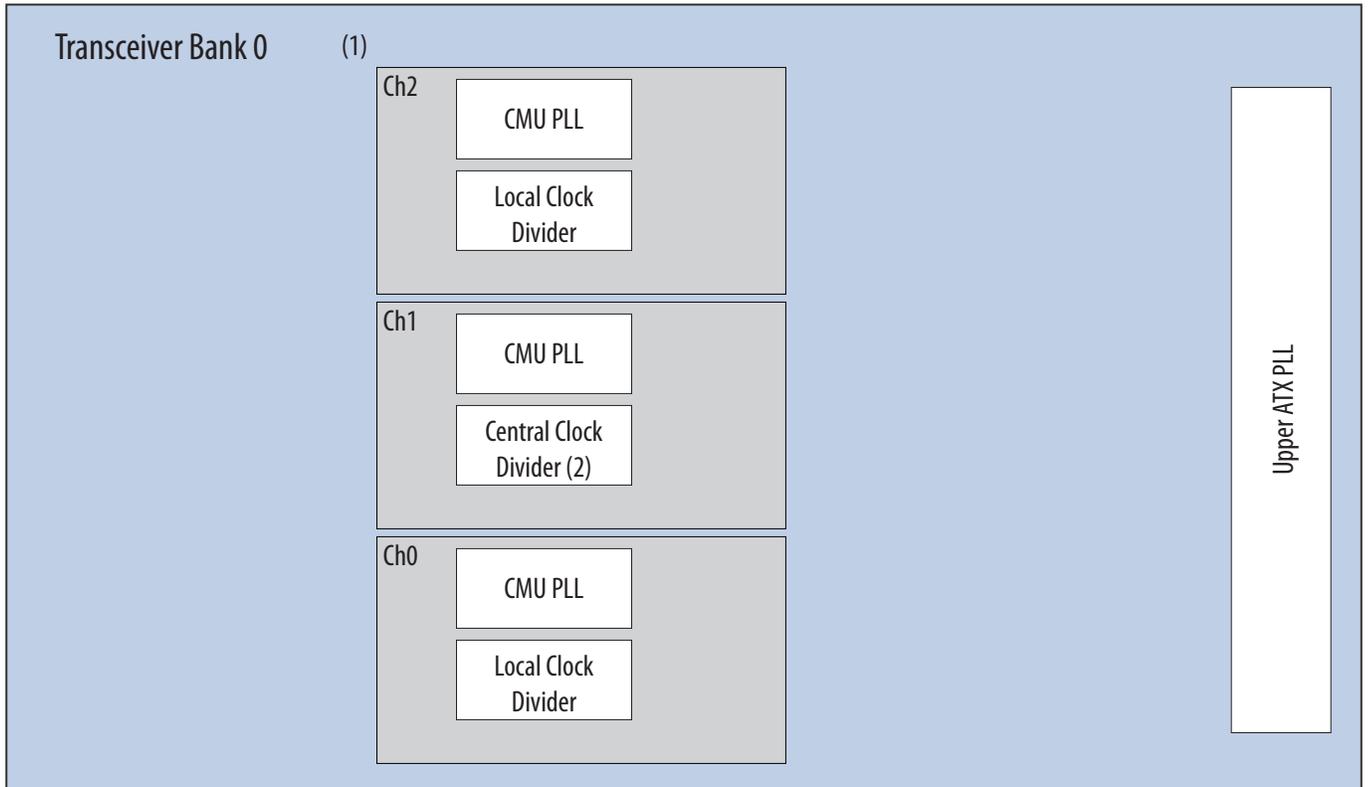
Note: When you enable the bit slip feature and do not use CPRI or deterministic latency state machine, the clock name will be different.

Related Information

For more information, refer to the [Altera Transceiver PHY IP Core User Guide](#).

Transmitter PLLs

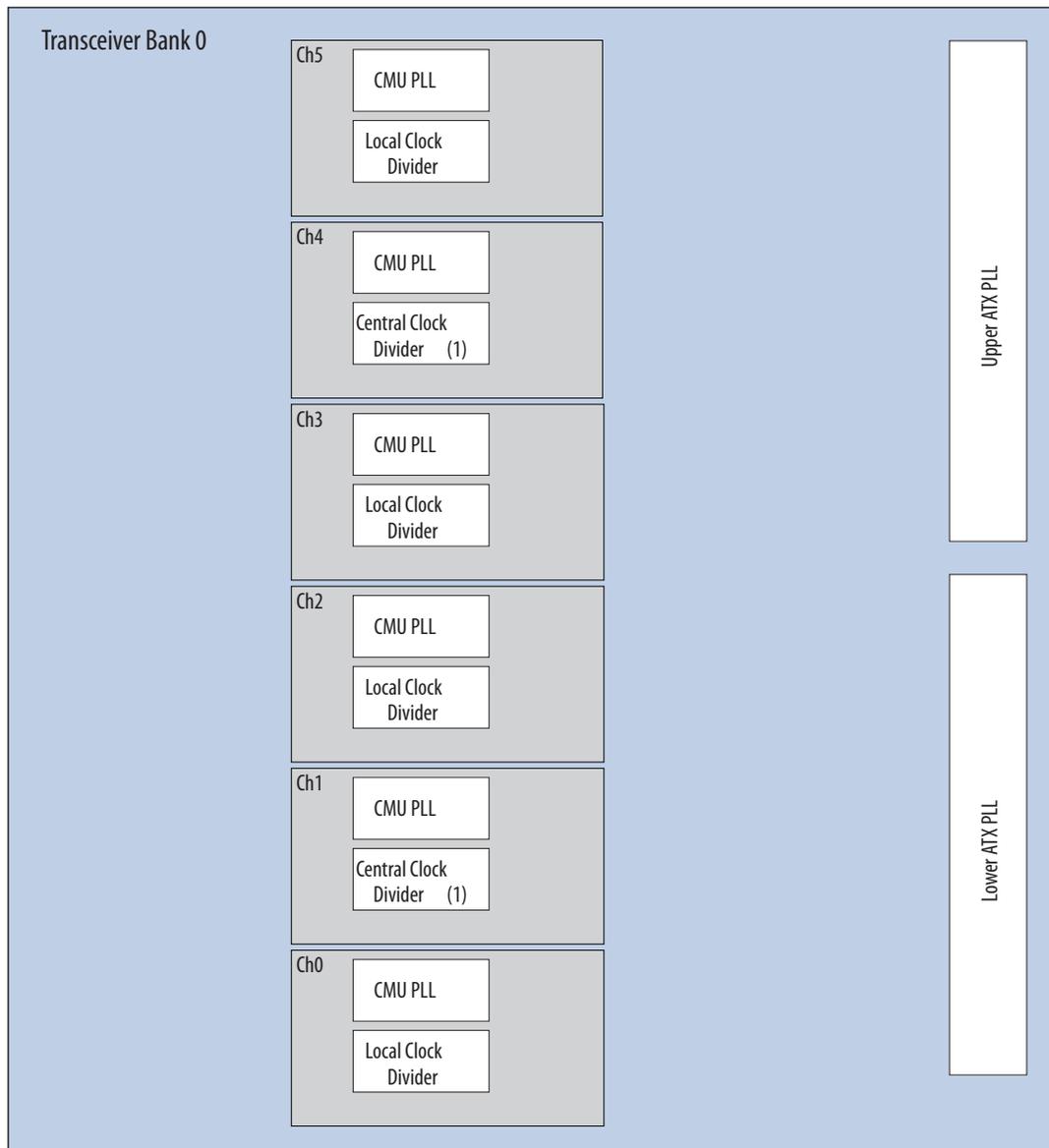
Figure 1-12: Transmitter PLL Locations in GX Devices (3 Channels)



Notes:

1. Stratix V devices 5SGXB5, 5SGXB6, 5SGSB9 and 5SGSBB have one transceiver bank on each side with only three transceiver channels.
2. You can use the central clock divider as a local clock divider.

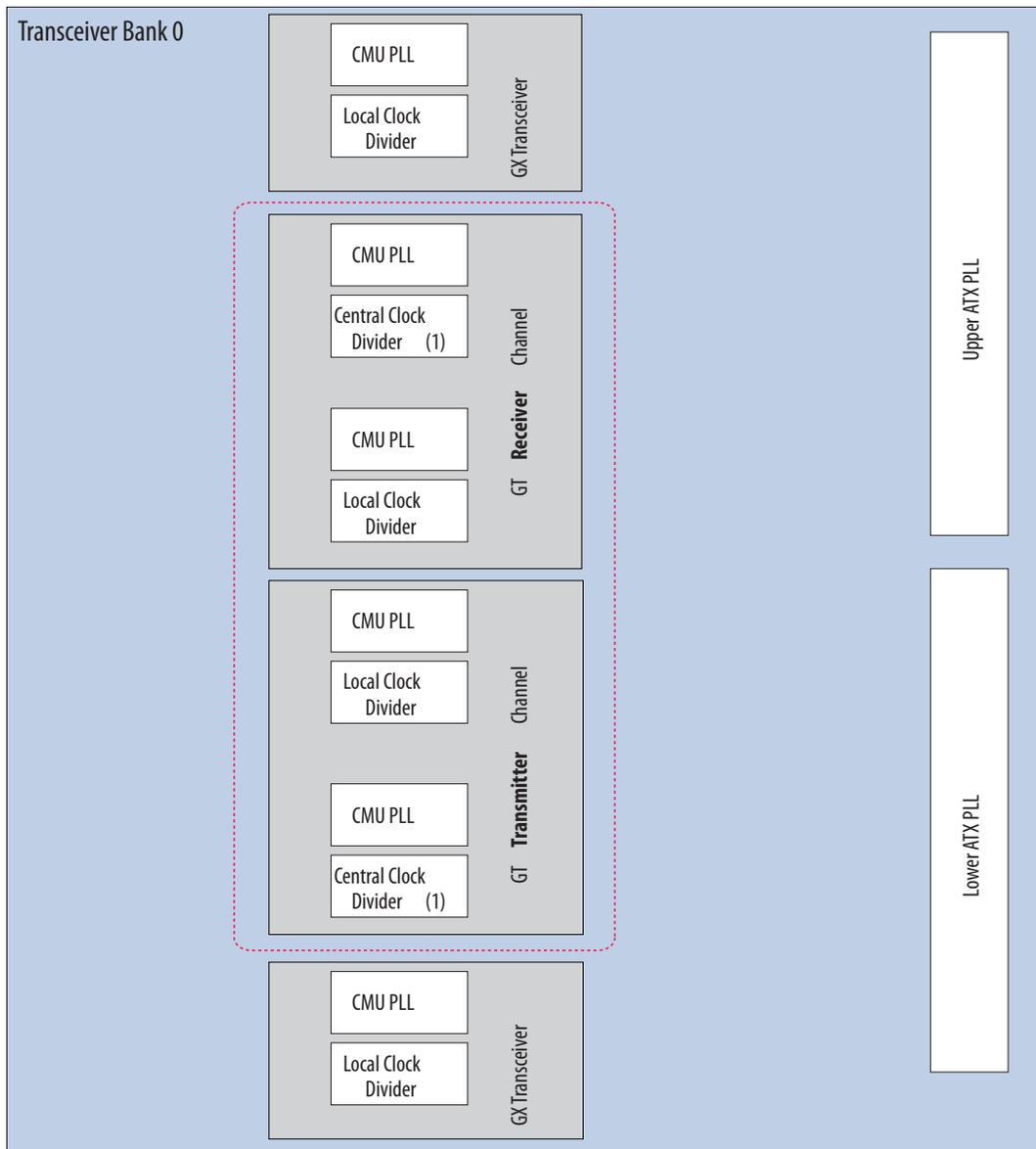
Figure 1-13: Transmitter PLL Locations in GX Devices (6 channels)



Note:

1. You can use the central clock divider as a local clock divider.

Figure 1-14: Transmitter PLL Locations in GT Devices



Notes:

1. You can use the central clock divider as a local clock divider.

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

Each transmitter channel has a clock divider called a local clock divider. The clock dividers generate the parallel and serial clock sources for the transmitter and optionally for the receiver PCS. The clock dividers on channels 1 and 4 are called central clock dividers because they can drive the x6 and xN clock lines. The central clock dividers can feed the clock lines used to bond channels.

Bonded Configurations

The high-speed serial clock and low-speed parallel clock skew between channels and unequal latency in the transmitter phase compensation FIFO contribute to transmitter channel-to-channel skew. In bonded channel configurations the parallel clock is generated by a central clock divider for all channels, rather than using a local clock divider for each transmitter channel. Also, the transmitter phase compensation FIFO in all bonded channels shares common pointers and control logic generated in the central clock divider, resulting in equal latency in the transmitter phase compensation FIFO of all bonded channels. The lower transceiver clock skew and equal latency in the transmitter phase compensation FIFOs in all channels provide lower channel-to-channel skew in bonded channel configurations.

Related Information

For more information about clocking or bonding, refer to the [Transceiver Clocking in Stratix V Devices](#) chapter.

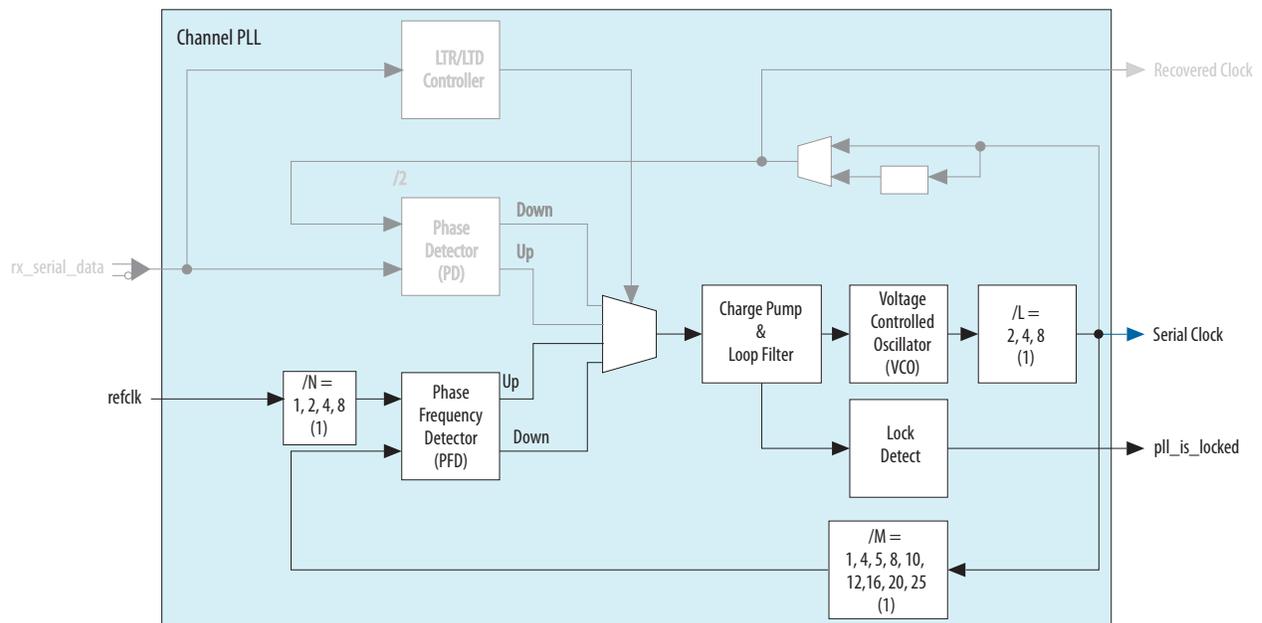
Channel PLL Used as a CMU PLL (Transmitter PLL)

The channel PLL available in the PMA can be used as a CMU PLL. The CMU PLLs in channels 1 and 4 in a transceiver block can also provide a clock to the other transceivers within the same block.

When you use the channel PLL as a CMU PLL, that particular channel cannot be used as a receiver; however, that channel can be used as a transmitter in conjunction with a different transmitter PLL. If all transmitters and receivers within the transceiver block are required, you must use an ATX PLL or a clock from another transceiver block.

For the best performance based on the data rate and input clock frequency, all settings for the CMU PLL and clock dividers are automatically chosen by the Quartus II software.

Figure 1-15: Channel PLL Configured as CMU PLL



Note:

1. Not all combinations of N , M , and L values are valid. The Quartus II software automatically chooses the optimal values.

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

Related Information

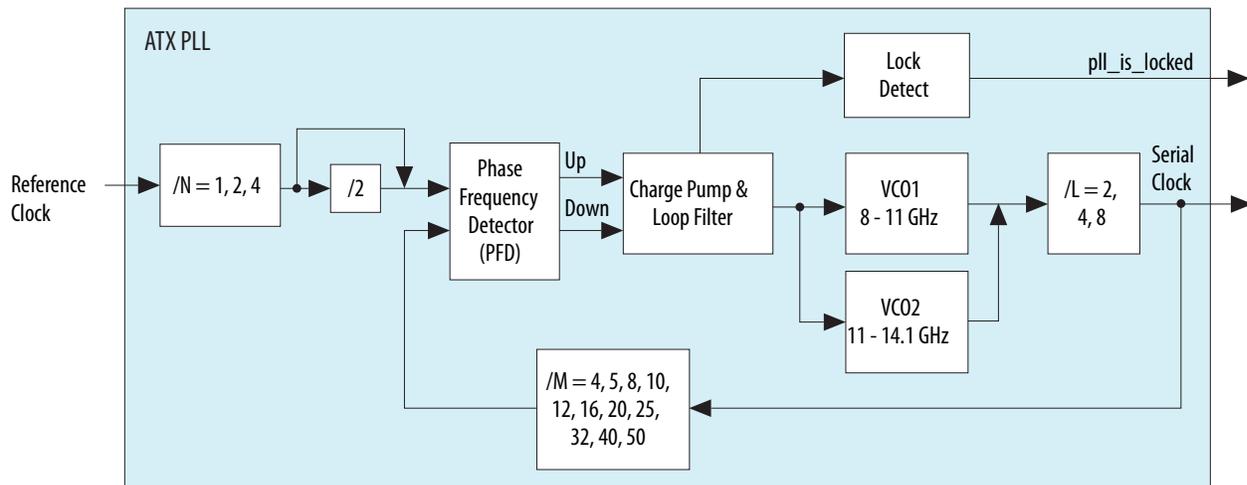
For CMU PLL specifications such as input clock frequency or supported output data ranges, refer to the [Stratix V Device Datasheet](#).

Auxiliary Transmit (ATX) PLL Architecture

Most Stratix V GT, GX, and GS devices contain two ATX PLLs per transceiver bank that can generate the high-speed clocks for the transmitter channels; the 66-channel device is an exception with only one ATX PLL in the top bank. 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 pins that reside in the same transceiver block as your channel. However, you can use any dedicated reference clock input pins along the same side of the device to clock the ATX PLL.

Figure 1-16: ATX PLL Architecture



Related Information

For ATX PLL specifications such as input clock frequency or supported output data ranges, refer to the [Stratix V Device Datasheet](#).

Transmitter Serializer

The serializer converts the incoming low-speed parallel data from the transceiver PCS or FPGA fabric to high-speed serial data and sends the data to the transmitter buffer. The Stratix V GX channel serializer supports an 8- and 10-bit, 16- and 20-bit, 32- and 40-bit, and 64-bit serialization factor. By default, the serializer block sends out the LSB of the input data first. For Stratix V GT channels, the serializer only supports a 128-bit serialization factor.

PCIe Receiver Detect

For a PCIe configuration for Gen1, Gen2, and Gen3 data rates, the transmitter buffers have a built-in receiver detection circuit. This receiver detection circuit detects if there is a receiver downstream by sending out a pulse on the common mode of the transmitter and monitoring the reflection.

PCIe Electrical Idle

The transmitter output buffers support transmission of PCIe electrical idle (or individual transmitter tri-state).

Related Information

[For more information about the PCIe protocol, refer to the Transceiver Configurations in Stratix V Devices chapter.](#)

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.

You can program these features using the assignment editor and a reconfiguration controller.

Note: The Stratix V GT transmitter buffer has only two taps for the pre-emphasis.

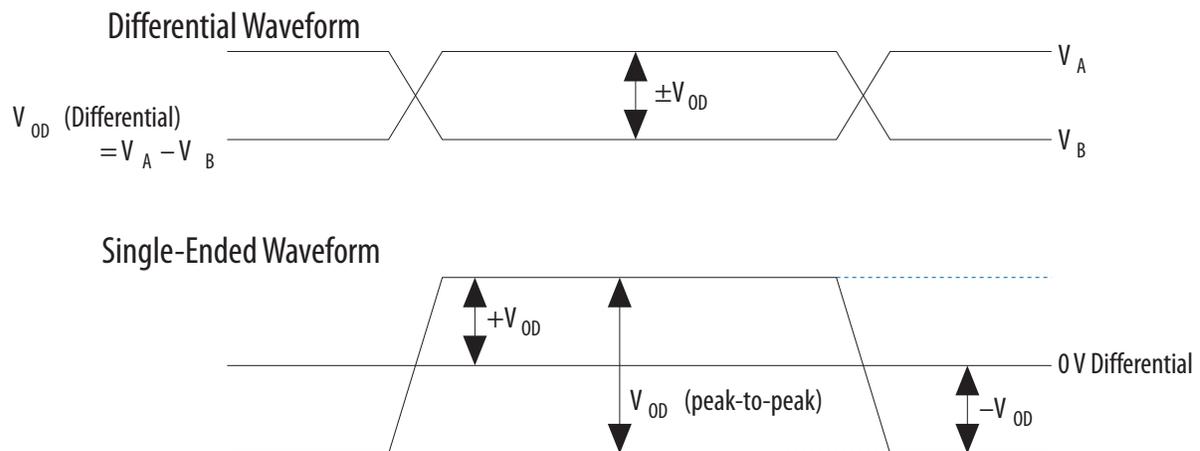
Transmitter Analog Settings

The transmitter analog setting capability improves signal integrity. These analog settings can be dynamically reconfigured without powering down the device using the PMA analog registers within the reconfiguration controller on GX devices only. Dynamic reconfiguration of the PMA settings can be done concurrently and independently per channel

Programmable Output Differential Voltage

You can program the differential output voltage to handle different trace lengths, various backplanes, and receiver requirements.

Figure 1-17: VOD (Differential) Signal Level



Programmable Pre-Emphasis

Pre-emphasis can maximize the data eye margin at the far-end receiver. The programmable pre-emphasis module in each transmit buffer boosts high frequencies in the transmit data signal, to compensate for attenuation in the transmission media.

Stratix V GX channels provide three pre-emphasis taps: pre-tap (16 settings), first post-tap (32 settings), and second post-tap (16 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.

The Stratix V GT channels have two pre-emphasis taps: pre-tap and post-tap. Each tap has 32 settings.

Related Information

- [For more information about the dynamic reconfiguration feature, refer to the Dynamic Reconfiguration in Stratix V Devices chapter.](#)
- [For details about the transceiver reconfiguration controller IP, refer to the Altera Transceiver PHY IP Core User Guide.](#)

Programmable Transmitter On-Chip Termination (OCT)

Transmitter buffers include programmable on-chip differential termination of 85, 100, 120, 150 Ω or OFF, for GX channels. The transmitter buffer of GT channels only supports 100 Ω differential termination, but allows for finer adjustment. The resistance is adjusted by the on-chip calibration circuit during calibration, which compensates for PVT changes. The transmitter buffers are current mode drivers. Therefore, the resultant V_{OD} is a function of the transmitter termination value.

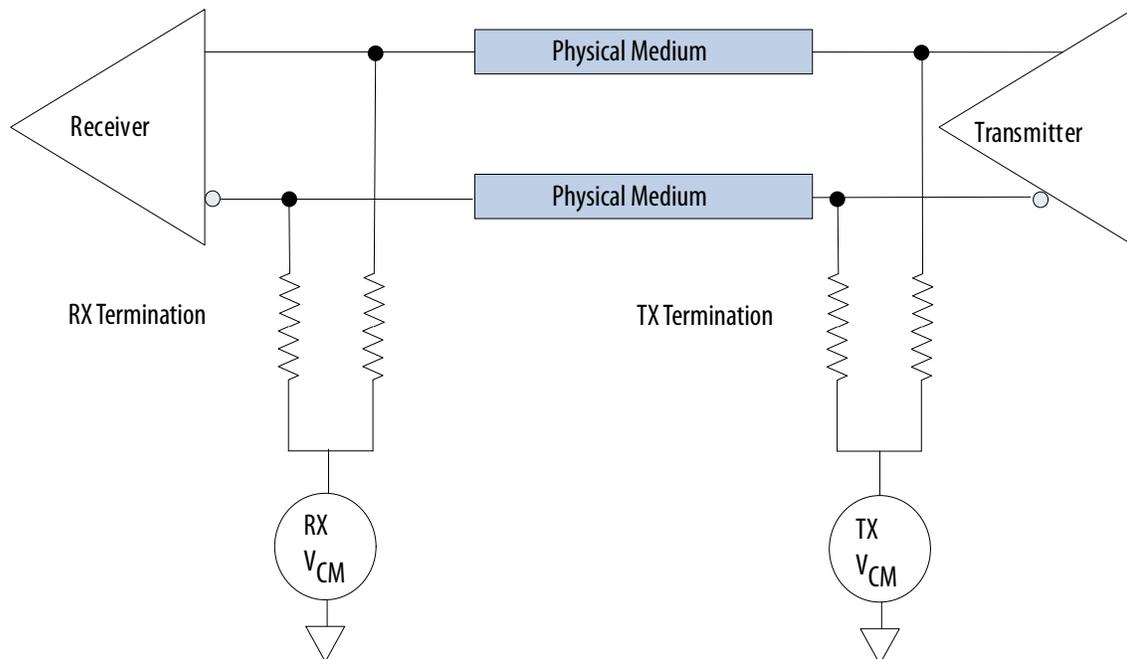
You can disable OCT and use external termination. If you select external termination, the transmitter common mode is tri-stated. Common mode is based on the external termination connection.

Link Coupling

A high-speed serial link can be AC-coupled or DC-coupled, depending on the serial protocol you are implementing.

In a DC-coupled link, the transmitter V_{CM} is seen unblocked at the receiver buffer. The link V_{CM} depends on the transmitter V_{CM} and the receiver V_{CM} . The on-chip or off-chip receiver termination and biasing circuitry must ensure compatibility between the transmitter and receiver V_{CM} .

Figure 1-18: DC-Coupled Link



You can DC-couple a GX/GS channel transmitter to a GX/GS channel receiver for the entire data rate range from 600 Mbps to 14.1 Gbps.

Note: The GT channels only support AC coupling.

Related Information

For DC-coupling other Altera families, and non-Altera devices, refer to the GX/GS transmitter and receiver common-mode requirements listed in the Stratix V Device Datasheet.

Transceiver Calibration Blocks

The Stratix V transceiver calibration blocks include OCT calibration, offset cancellation in the receiver buffer and receiver CDR, and ATX PLL calibration.

OCT Calibration

Stratix V GX and GS devices contain calibration circuits that calibrate the OCT resistors and analog portions of the transceiver blocks to ensure that the functionality is independent of process, voltage, and temperature (PVT) variation. GT devices do not have calibration support.

The calibration block internally generates a constant reference voltage, independent of PVT variation. The calibration block uses the internal reference voltage and external reference resistor to generate constant reference currents.

Note: You must connect a separate 1.8 k Ω (maximum tolerance $\pm 1\%$) external resistor on each R_{REF} pin to ground. To ensure the calibration block operates properly, the R_{REF} resistor connection on the board must be free from external noise.

Related Information

For more information about connections, refer to the [Pin Connection Guidelines](#).

Offset Cancellation in the Receiver Buffer and Receiver CDR

Process variation can lead to a V_{CM} offset between the p and n signals within the differential buffers. Stratix V 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

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 once automatically after device initialization. The calibration can be restarted via register accesses to the Reconfiguration controller if desired.

Both the ATX PLL and reconfiguration reference clocks must be valid and stable upon device power-up to ensure a successful power-up ATX PLL calibration and for its subsequent calibrations. The reference clocks must be stable and the ATX PLL cannot be in reset (`p11_powerdown`) when ATX PLL calibration is performed.

Related Information

- For more information, refer to the [Dynamic Reconfiguration in Stratix V Devices](#) chapter.
- For more information about ATX PLL calibration, refer to the [Auxiliary Transmit \(ATX\) PLL Calibration](#) section of the [Altera Transceiver PHY IP Core User Guide](#)

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 1-4: Transceiver Calibration Block Boundary for Stratix V Devices

Stratix V 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)
5SGTC5	KF40	36	24 (Left) / 12 (Right)	12	12

Stratix V 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)
5SGTC7	KF40	36	24 (Left) / 12 (Right)	12	12
5SGXA3	EH29	12	12 (Left only)	6	6
	HF35	24	12	6	6
	KF35	36	18	9	9
	KF40/ KH40	36	18	9	9
5SGXA4	HF35	24	12	6	6
	KF35	36	18	9	9
	KF40/ KH40	36	18	9	9
5SGXA5	HF35	24	12	6	6
	KF35	36	18	9	9
	KF40/ KH40	36	18	9	9
	NF40	48	24	12	12
5SGXA7	HF35	24	12	6	6
	KF35	36	18	9	9
	KF40/ KH40	36	18	9	9
	NF40	48	24	12	12
5SGXA9	KF40/ KH40	36	18	9	9
	NF45	48	24	12	12
5SGXAB	KF40/ KH40	36	18	9	9
	NF45	48	24	12	12
5SGXB5	RF40	66	33	15	18
	RF43	66	33	15	18
5SGXB6	RF40	66	33	15	18
	RF43	66	33	15	18

Stratix V 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)
5SGXB9	RH43	66	33	15	18
5SGXBB	RH43	66	33	15	18
5SGSD3	EH29	12	12 (Left only)	6	6
	HF35	24	12	6	6
5SGSD4	EH29	12	12 (Left only)	6	6
	HF35	24	12	6	6
	KF40	36	18	9	9
5SGSD5	HF35	24	12	6	6
	KF40	36	18	9	9
5SGSD6	KF40	36	18	9	9
	NF45	48	24	12	12
5SGSD8	KF40	36	18	9	9
	NF45	48	24	12	12

Related Information

[Refer to the Transceiver Reconfiguration Controller PMA Analog Control Registers section of the Altera Transceiver PHY IP Core User Guide](#)

PMA Reconfiguration

Modifying programmable values within transceiver 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 channels (triplet) in the upper-half of a bank, and one for the lower-half. This is due to a single Avalon-Memory Mapped (AVMM) slave interface per 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.

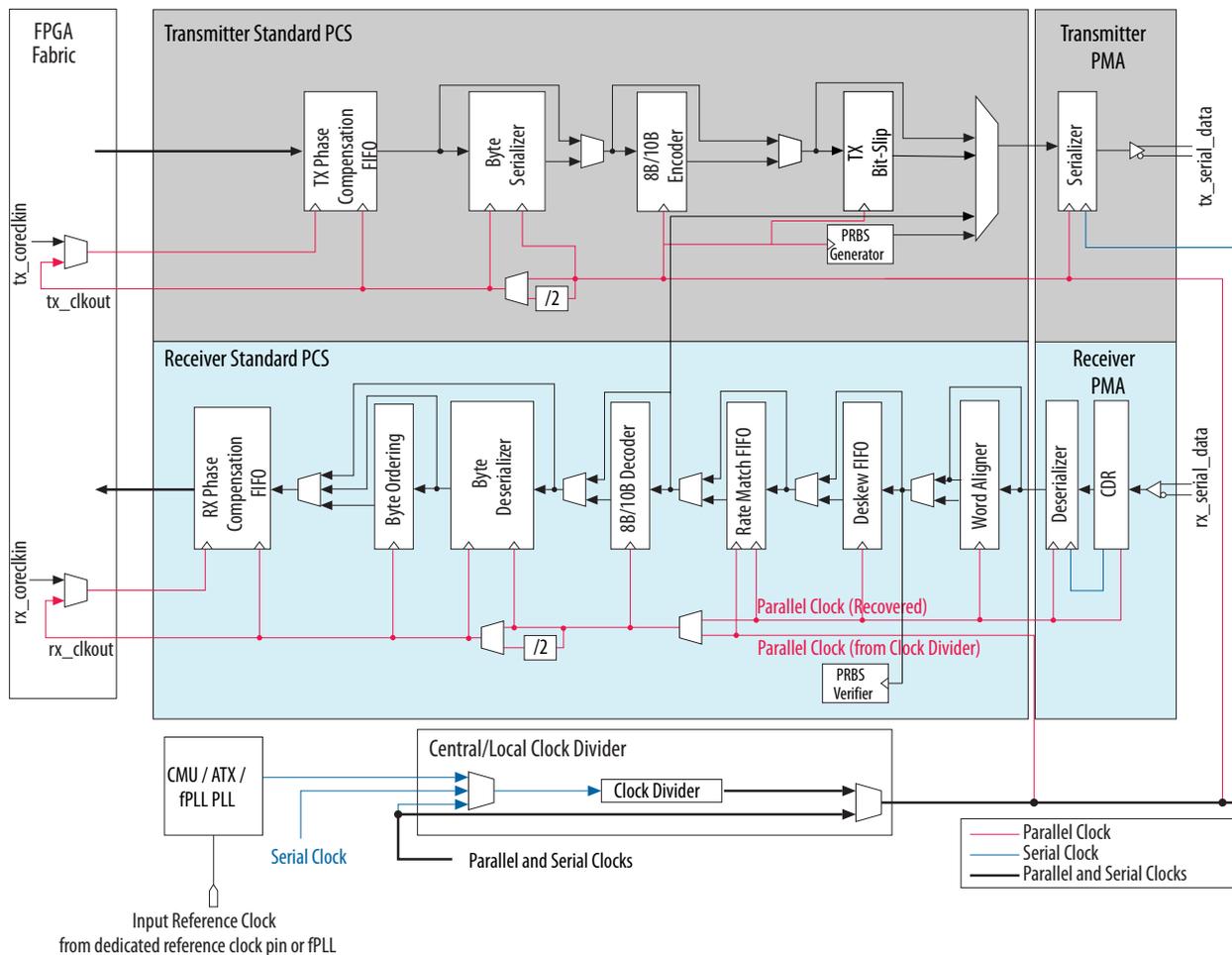
Related Information

[Refer to the "Transceiver Reconfiguration Controller IP Core" chapter of the Altera Transceiver PHY IP Core User Guide.](#)

Standard PCS Architecture

The transceiver Standard PCS circuit blocks support data rates up to 12.2 Gbps depending on the transceiver speed grade.

Figure 1-19: Standard PCS Datapath in GX Channels



Note: The PCS can be completely bypassed using the PMA direct mode. This is an option for GX channels. The GT channels do not have a PCS, and always operate in PMA direct mode.

Some transceiver channels interface to the PCIe hard IP block, the PIPE interface for soft IP implementations of PCIe, or directly to the FPGA fabric (FPGA fabric-transceiver interface). The transceiver channel interfaces to the PCIe hard IP block if you use the hard IP block to implement the PCIe PHY MAC, data link layer, and transaction layer. Otherwise, the transceiver channel interfaces directly to the FPGA fabric.

The PCIe hard IP-transceiver interface is outside the scope of this chapter. This chapter describes the FPGA fabric-transceiver interface only.

You can divide the standard transceiver channel datapath into two configurations based on the FPGA fabric-transceiver interface width (channel width) and the transceiver channel PMA-PCS width (serialization factor).

Table 1-5: Transceiver Interface Widths

Name	8- and 10-Bit PMA-PCS Widths	16- and 20-Bit PMA-PCS Widths
FPGA fabric-transceiver interface widths	8 and 10 bit 16 and 20 bit	16 and 20 bit 32 and 40 bit
Supported configurations	PCIe Gen1 and Gen2 XAUI Custom configuration (Custom or Native or Low Latency PHY IPs)	Custom configuration (Custom or Native or Low Latency PHY IPs)
Data rate range in a custom configuration	0.6 to 3.75 Gbps	1.0 to 12.2 Gbps

The standard PCS can be configured for various protocols by selecting different PCS blocks in the receiver and transmitter datapath.

Related Information

- [For more information about the PMA direct mode, refer to the "Stratix V GX Native PHY IP" section of the Transceiver Configurations in Stratix V Devices chapter.](#)
- [For details about the PMA direct mode, refer also to the "Stratix V Transceiver Native PHY IP Core" section of the Altera Transceiver PHY IP Core User Guide.](#)
- [For details about selecting different PCS blocks in the receiver and transmitter datapath and about the data rates supported in other configurations, refer to the Transceiver Configurations in Stratix V Devices chapter.](#)
- [For details about implementing the different PCS functional blocks, refer to the Altera Transceiver PHY IP Core User Guide.](#)

Receiver Standard PCS Datapath

The functional blocks in the receiver datapath are described in order from the word aligner to the receiver phase compensation FIFO buffer at the FPGA fabric-transceiver interface.

The receiver datapath is flexible and allows multiple modes, depending on the selected configuration.

Note: The Standard PCS is not supported in the GT channels.

Word Aligner

Because the data is serialized before transmission and then deserialized at the receiver, the data loses the word boundary of the upstream transmitter after deserialization. The word aligner receives parallel data from the deserializer and restores the word boundary based on a pre-defined alignment pattern that must be received during link synchronization.

Serial protocols such as PCIe specify a standard word alignment pattern. For proprietary protocols, the transceiver architecture allows you to select a custom word alignment pattern specific to your implementation.

In addition to restoring the word boundary, the word aligner also implements the following features:

- Synchronization state machine in configurations such as PCIe
- Programmable run length violation detection in all configurations
- Receiver polarity inversion in all configurations except PCIe
- Receiver bit reversal in custom configurations
- Receiver byte reversal in custom 16- and 20-bit width configurations

Depending on the configuration, the word aligner operates in one of the following modes:

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

Table 1-6: Word Aligner Options

Configuration	PMA-PCS Interface Width (Bits)	Word Alignment Mode	Word Alignment Pattern Length	Word Alignment Behavior
Custom 8- and 10-Bit Width	8	Manual Alignment	16 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	N/A	User-controlled signal shifts data one bit at a time.
	10	Manual Alignment	7 and 10 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	N/A	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7 and 10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
		Deterministic Latency State Machine	10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.

Configuration	PMA-PCS Interface Width (Bits)	Word Alignment Mode	Word Alignment Pattern Length	Word Alignment Behavior
Custom 16- and 20-Bit Width	16	Manual Alignment	8, 16, and 32 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	N/A	User-controlled signal shifts data one bit at a time.
	20	Manual Alignment	7, 10, and 20 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	N/A	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7, 10, and 20 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
		Deterministic Latency State Machine	10 and 20 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
PCIe	10	Automatic Synchronized State Machine	10 bits	Automatically selected word aligner pattern length and pattern.

Example of Manual Alignment Mode Word Aligner with a 10-Bit PMA-PCS Interface Configuration

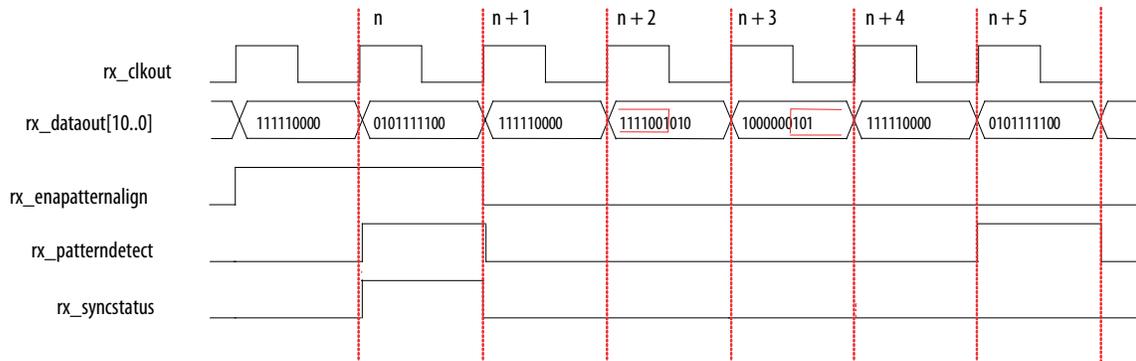
In basic single-width mode with a 10-bit PMA-PCS interface, you can configure the word aligner in manual alignment mode by selecting the **Use manual word alignment mode** option in the IP Catalog.

In manual alignment mode, the word alignment operation is manually controlled with the `rx_std_wa_patternalign` input signal or the `rx_enapatternalign` register. The word aligner operation is level-sensitive to `rx_enapatternalign`. If `rx_enapatternalign` is held high, the word aligner looks for the programmed 7-bit or 10-bit word alignment pattern in the received data stream. It updates the word boundary if it finds the word alignment pattern in a new word boundary. If `rx_enapatternalign` is deasserted low, the word aligner maintains the current word boundary even when it sees the word alignment pattern in a new word boundary.

Two status signals, `rx_syncstatus` and `rx_patterndetect`, with the same latency as the datapath, are forwarded to the FPGA fabric to indicate the word aligner status. After receiving the first word alignment pattern after the `rx_enapatternalign` signal is asserted high, both the `rx_syncstatus` and `rx_patterndetect` signals are driven high for one parallel clock cycle. Any word alignment pattern received thereafter in the same word boundary causes only the `rx_patterndetect` signal to go high for one clock cycle. Any word alignment pattern received thereafter in a different word boundary causes the word aligner to re-align to the new word boundary only if the `rx_enapatternalign` signal is held high. The word aligner asserts the `rx_syncstatus` signal for one parallel clock cycle whenever it re-aligns to the new word boundary.

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 cycle n because the `rx_enapatternalign` signal is asserted high. The `rx_syncstatus` signal goes high for one clock cycle, indicating alignment to a new word boundary. The `rx_patterndetect` signal also goes high for one clock cycle to indicate initial word alignment. At time $n + 1$, the `rx_enapatternalign` signal is de-asserted 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_enapatternalign` signal is held low. The /K28.5/ word alignment pattern is detected again in the current word boundary during cycle $n + 5$, causing the `rx_patterndetect` signal to go high for one parallel clock cycle.

Figure 1-20: Word Aligner with 10-Bit PMA-PCS Manual Alignment Mode



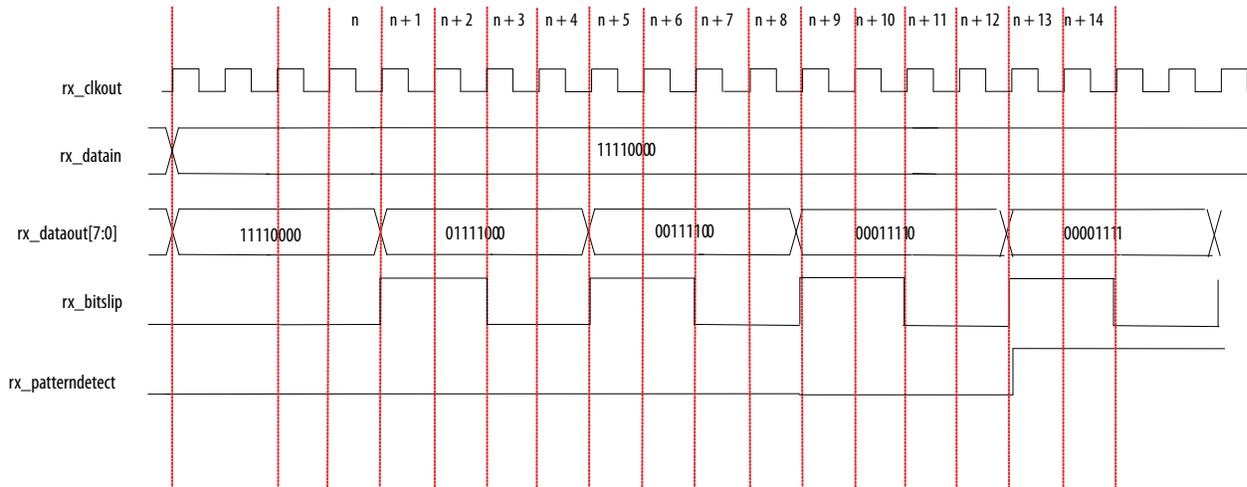
Note: If the word alignment pattern is known to be unique and does not appear between word boundaries, you can constantly hold `rx_enapatternalign` high because there is no possibility of false word alignment. If there is a possibility of the word alignment pattern occurring across word boundaries, you must control `rx_enapatternalign` to lock the word boundary after the desired word alignment is achieved to avoid re-alignment to an incorrect word boundary.

Example of Bit-Slip Mode Word Aligner with an 8-Bit PMA-PCS Interface Configuration

In a custom width configuration with an 8-bit PMA-PCS interface width, you can configure the word aligner in bit-slip mode. In bit-slip mode, the word aligner operation is controlled by the `rx_bitslip` bit of the `pcs8g_rx_wa_control` register. At every 0-1 transition of the `rx_bitslip` bit of the `pcs8g_rx_control` register, the bit-slip circuitry slips one bit into the received data stream, effectively shifting the word boundary by one bit. Also in bit-slip mode, the word aligner `pcs8g_rx_wa_status` register bit for `rx_patterndetect` is driven high for one parallel clock cycle when the received data after bit-slipping matches the 16-bit word alignment pattern programmed.

You can implement a bit-slip controller in the FPGA fabric that monitors the `rx_parallel_data` signal, the `rx_patterndetect` signal, or both, and controls the `rx_bitslip` signal to achieve word alignment.

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

Figure 1-21: Word Aligner Configured in Bit-Slip Mode

Example of Automatic Synchronization State Machine Mode Word Aligner with a 10-Bit PMA-PCS Interface Configuration

Protocols such as PCIe require the receiver PCS logic to implement a synchronization state machine to provide hysteresis during link synchronization. Each of these protocols defines a specific number of synchronization code groups that the link must receive to acquire synchronization, and a specific number of erroneous code groups that the protocol must receive to fall out of synchronization.

In PCIe configurations, the word aligner is in automatic synchronization state machine mode. The word aligner automatically selects the word alignment pattern length and pattern as specified by each protocol.

The synchronization state machine parameters are fixed for PCIe configurations as specified by the respective protocol.

Table 1-7: Synchronization State Machine Modes for a PCI-Express (PIPE) Configuration

Mode	PCIe
Number of valid synchronization code groups or ordered sets received to achieve synchronization	4
Number of erroneous code groups received to lose synchronization	17
Number of continuous good code groups received to reduce the error count by one	16

After deassertion of the `rx_digitalreset` signal in automatic synchronization state machine mode, the word aligner starts looking for the word alignment pattern or synchronization code groups in the received data stream. When the programmed number of valid synchronization code groups or ordered sets is received, the `rx_syncstatus` status bit is driven high to indicate that synchronization is acquired. The `rx_syncstatus` status bit is constantly driven high until the programmed number of erroneous code groups is received without receiving intermediate good groups, after which `rx_syncstatus` is driven low. The word aligner indicates loss of synchronization (`rx_syncstatus` remains low) until the programmed number of valid synchronization code groups are received again.

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.

Table 1-8: Word Alignment in Deterministic Latency State Machine Mode

PCS Mode	PMA-PCS Interface Width	Word Alignment Operation
Single Width	10 bits	<ol style="list-style-type: none"> 1. After <code>rx_digitalreset</code> 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 <code>rx_syncstatus</code> register until <code>rx_digitalreset</code> is asserted. 4. To resynchronize to the new word boundary, the Avalon-MM register <code>rx_enapatternalign</code> (not available as a signal) must be reasserted to initiate another pattern alignment. Asserting <code>rx_enapatternalign</code> may cause the extra shifting in the RX datapath if <code>rx_enablepatternalign</code> is asserted while bit slipping is in progress. Consequently, <code>rx_enapatternalign</code> should only be asserted under the following conditions: <ul style="list-style-type: none"> • <code>rx_syncstatus</code> is asserted • <code>rx_bitslipboundaryselectout</code> changes from a non-zero value to zero or 1 5. When the word aligner synchronizes to the new word boundary, <code>rx_syncstatus</code> has a value of 1 until <code>rx_digitalreset</code> is deasserted or <code>rx_enapatternalign</code> is set to 1. <code>rx_patterndetect</code> 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 circuit resides in the word aligner block and detects consecutive 1s or 0s in the data. If the data stream exceeds the preset maximum number of consecutive 1s or 0s, the violation is signified by the assertion of the `rx_rlv` status bit.

Table 1-9: Detection Capabilities of the Run Length Violation Circuit

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

Receiver Polarity Inversion

The positive and negative signals of a serial differential link may be erroneously swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic are expensive. The receiver polarity inversion feature is provided to correct this situation.

Receiver Bit Reversal

By default, the receiver assumes an LSB-to-MSB transmission. If the transmission order is MSB-to-LSB, the receiver forwards the bit-flipped version of the parallel data to the FPGA fabric on `rx_parallel_data`. For example, if in 8 bit width mode, `D[7:0]` is rewired to `D[0:7]`.

Receiver Byte Reversal in Custom 16- and 20-Bit Width Configurations

The MSByte and LSByte of the input data to the transmitter may be erroneously swapped. The receiver byte reversal feature corrects this situation.

Figure 1-22: Receiver Byte Reversal Feature

**Related Information**

For more information on the receiver polarity, bit reversal, and byte reversal features, refer to the "Bit Reversal and Polarity Inversion" section of the [Altera Transceiver PHY IP Core User Guide](#).

PRBS Verifier

The pseudo-random bit stream (PRBS) verifier block verifies the pattern generated by the PRBS generator.

The PRBS verifier:

- Supports 32-bit and 40-bit PMA interfaces
- Supports the following modes and patterns:
 - PRBS31: $x^{31} + x^{28} + 1$
 - PRBS23: $x^{23} + x^{18} + 1$
 - PRBS15: $x^{15} + x^{14} + 1$
 - PRBS9: $x^9 + x^5 + 1$
 - PRBS7: $x^7 + x^6 + 1$

Note: You can enable either the PRBS verifier or the PRP verifier, but you cannot enable both at the same time.

Related Information

[For PRBS verifier implementation information, refer to the Altera Transceiver PHY IP Core User Guide.](#)

Deskew FIFO

The deskew FIFO in each channel receives data from its word aligner. The deskew FIFO circuitry aligns the data across multiple channels.

Note: The deskew circuitry is available only in XAUI mode.

In the XAUI protocol, the code groups received across four lanes of a XAUI link can be misaligned with respect to one another because of skew in the physical medium or differences between the independent clock recoveries per lane.

The XAUI protocol requires the physical layer device to implement a deskew circuit to align all four channels. To enable the deskew circuitry at the receiver, the transmitter sends a /A/ (/K28.3/) code group simultaneously on all four channels during inter-packet gap (IPG). The skew introduced in the physical medium and the receiver channels can cause the /A/ code groups to be misaligned when received.

The deskew operation begins only after link synchronization is achieved on all four channels from the word aligner of each channel. When an aligned set of /A/ code groups is observed at the output of the deskew FIFOs of the four channels, the `rx_channel1aligned` signal is asserted high, indicating channel alignment is acquired.

Rate Match (Clock Rate Compensation) FIFO

The rate match (clock rate compensation) FIFO compensates for small clock frequency differences between the upstream transmitter and the local receiver clocks by inserting or removing skip (SKP) symbols or ordered sets from the interpacket gap (IPG) or idle streams. The rate match FIFO deletes SKP symbols or ordered sets when the upstream transmitter reference clock frequency is higher than the local receiver reference clock frequency. The rate match FIFO inserts SKP symbols or ordered sets when the local receiver reference clock frequency is higher than the upstream transmitter reference clock frequency.

Note: For the Gigabit Ethernet protocol, if you have the auto-negotiation state machine in the FPGA core with rate match FIFO enabled, refer to the "Rate Match FIFO" section in the "Gigabit Ethernet" section in the *Transceiver Configurations in Stratix V Devices* chapter.

Related Information

- [For more information about the skip pattern and control pattern, refer to the "Rate Match FIFO Parameters" section of the Altera Transceiver PHY IP Core User Guide.](#)
- [For more information about how to use the rate match FIFO with PCIe, XAUI, and Custom protocols, refer to the Transceiver Configurations in Stratix V Devices chapter.](#)

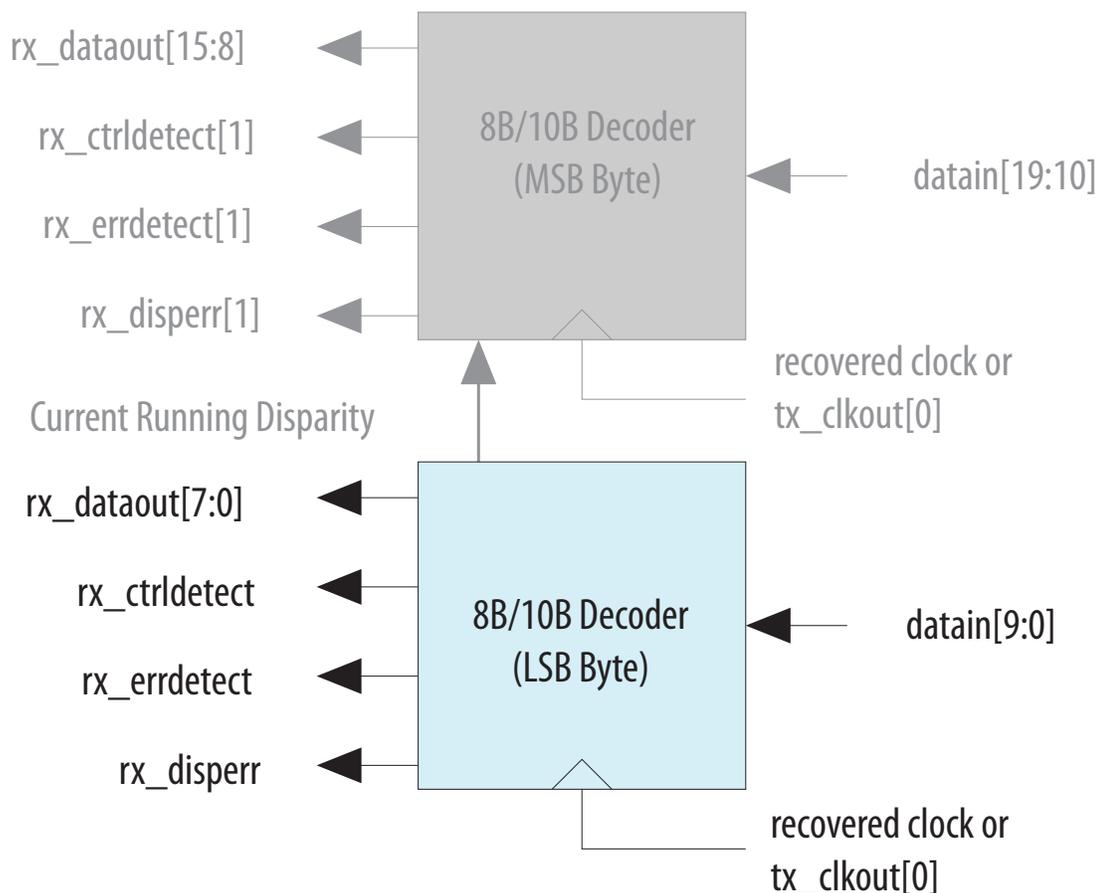
8B/10B Decoder

Many protocols require the serial data sent over the link to be 8B/10B encoded to maintain the DC balance in the transmitted serial data. The PCIe protocol requires the receiver PCS logic to implement an 8B/10B decoder to decode the data before forwarding the data to the upper layers for packet processing.

The receiver channel PCS has an 8B/10B decoder after the rate match FIFO. In configurations with the rate match FIFO enabled, the 8B/10B decoder receives data from the rate match FIFO. In configurations with the rate match FIFO disabled, the 8B/10B decoder receives data from the word aligner.

In 10-bit mode, the 8B/10B decoder receives 10-bit data from the rate match FIFO or word aligner (when the rate match FIFO is disabled) and decodes the data into an 8-bit data +1-bit control identifier. The decoded data is fed to the byte deserializer or the receiver phase compensation FIFO (if the byte deserializer is disabled).

Figure 1-23: 8B/10B Decoder in 10-bit Mode



Note: The 8B/10B decoder is described in IEEE 802.3-2008 clause-49.

In PCIe configuration, the 8B/10B decoder operates only in 10-bit width mode. A PCIe configuration forces selection of the 8B/10B decoder in the receiver datapath.

Control Code Group Detection

The 8B/10B decoder indicates whether the decoded 8-bit code group is a data or control code group on the `rx_dataak` signal (not shown in the figure above). If the received 10-bit code group is one of the 12 control code groups (`/Kx.y/`) specified in the IEEE 802.3 specification, the `rx_dataak` signal is driven high. If the received 10-bit code group is a data code group (`/Dx.y/`), the `rx_dataak` signal is driven low.

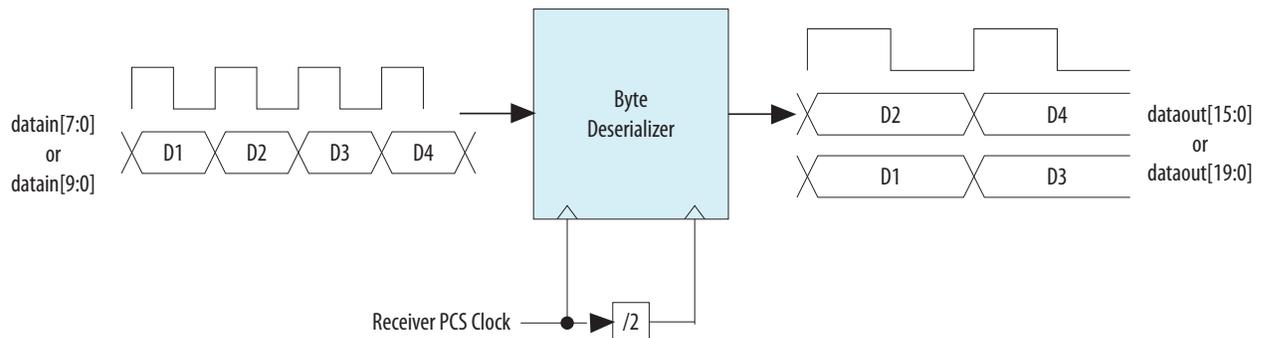
Byte Deserializer

The FPGA fabric-transceiver interface frequency has an upper limit. In designs where the receiver PCS frequency exceeds the upper limit, the byte deserializer is required. The byte deserializer reduces the interface frequency to half while doubling the parallel data width. The byte deserializer is optional in designs that do not exceed the interface frequency upper limit.

Byte Deserializer in 8- and 10-Bit Width Mode

In 8-bit width mode, the byte deserializer receives 8-bit wide data from the 8B/10B decoder or 10-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes the data into 16- or 20-bit wide data at half the speed.

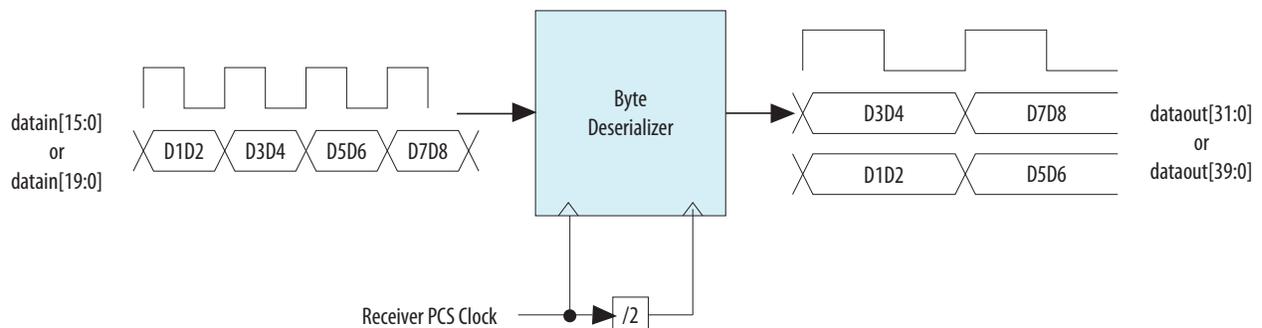
Figure 1-24: Byte Deserializer in 8- or 10-Bit Width Mode



Byte Deserializer in 16- or 20-Bit Width Mode

In 16-bit width mode, the byte deserializer receives 16-bit wide data from the 8B/10B decoder or 20-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes the data into 32- or 40-bit wide data at half the speed.

Figure 1-25: Byte Deserializer in 16- and 20-Bit Width Mode



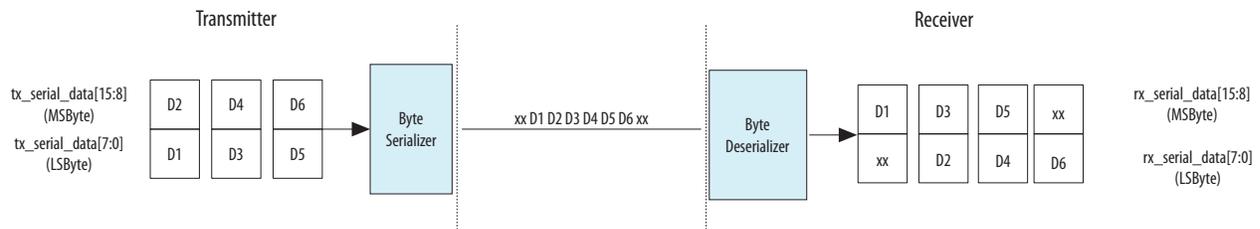
Byte Ordering Block

In 8- or 10-bit width mode with the 16- or 20-bit FPGA fabric-transceiver interface, the byte deserializer receives one data byte (8 or 10 bits) and deserializes the data into two data bytes (16 or 20 bits). Depending on when the receiver PCS logic comes out of reset, the byte ordering at the output of the byte deserializer may or may not match the original byte ordering of the transmitted data. The byte misalignment resulting

from byte deserialization is unpredictable because the byte misalignment depends on which byte is being received by the byte deserializer when the byte comes out of reset.

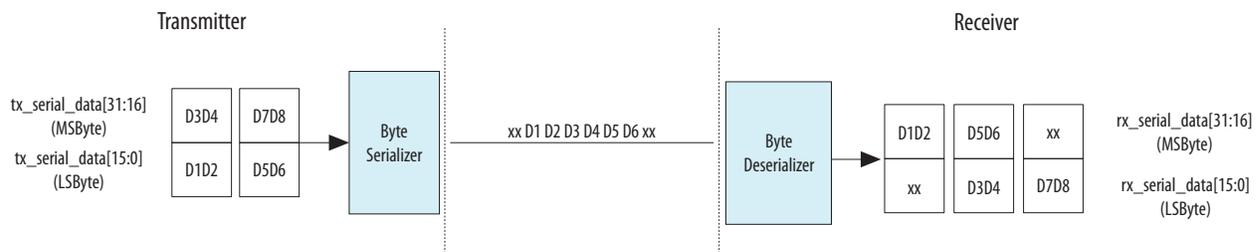
The byte ordering block looks for the user-programmed byte ordering pattern in the byte-deserialized data. You must select a byte ordering pattern that you know is at the LSBytes position of the parallel transmitter data. If the byte ordering block finds the programmed byte ordering pattern in the MSBytes position of the byte-deserialized data, the byte ordering block inserts the appropriate number of user-programmed pad bytes to push the byte ordering pattern to the LSByte(s) position, thereby restoring proper byte ordering.

Figure 1-26: MSByte and LSByte of the Two-Bit Transmitter Data Straddled Across Two Word Boundaries



In 16-bit width mode with a 32-bit FPGA fabric-transceiver interface, the byte deserializer receives two data bytes (16 bits) and deserializes the two data bytes into four data bytes (32 bits).

Figure 1-27: MSByte and LSByte of the Four-Bit Transmitter Data Straddled Across Two Word Boundaries



The byte ordering pattern length and the byte ordering pad pattern length vary depending on the PCS-PMA interface width (8-bit/10-bit/16-bit/20-bit).

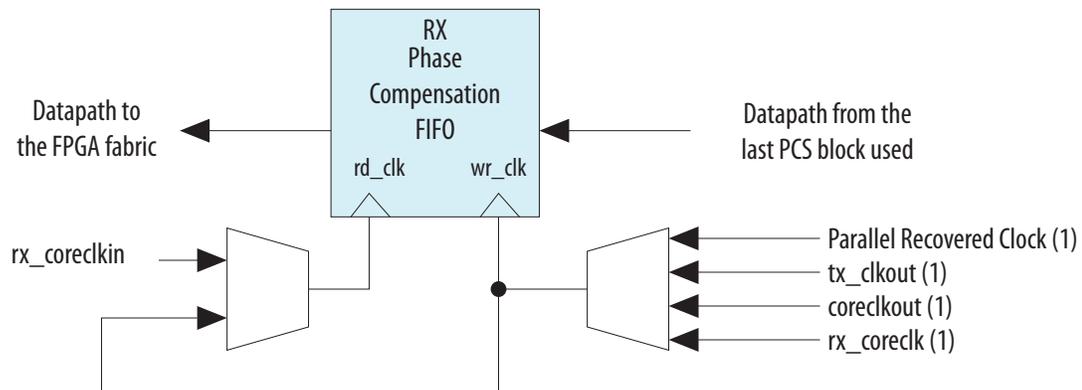
Related Information

For details about byte ordering and byte ordering pad pattern lengths, refer to the "Byte Ordering Parameters" section of the Altera Transceiver PHY IP Core User Guide.

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

Figure 1-28: Receiver Phase Compensation FIFO**Note:**

1. If you use the byte deserializer, these clocks are divided by two.

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, see [Transceiver Clocking in Stratix 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.

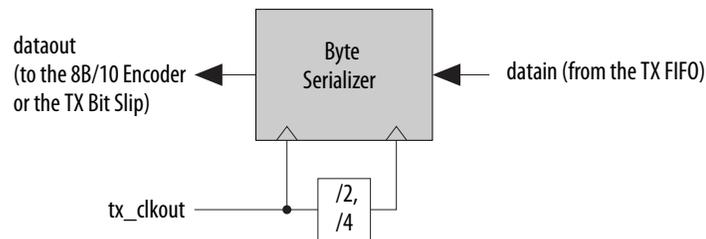
Transmitter Standard PCS Datapath

Note: The Standard PCS is not supported in the GT channels.

Transmitter Phase Compensation FIFO

The transmitter phase compensation FIFO interfaces with the FPGA fabric. The transmitter phase compensation FIFO compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock.

Figure 1-29: Transmitter Phase Compensation FIFO



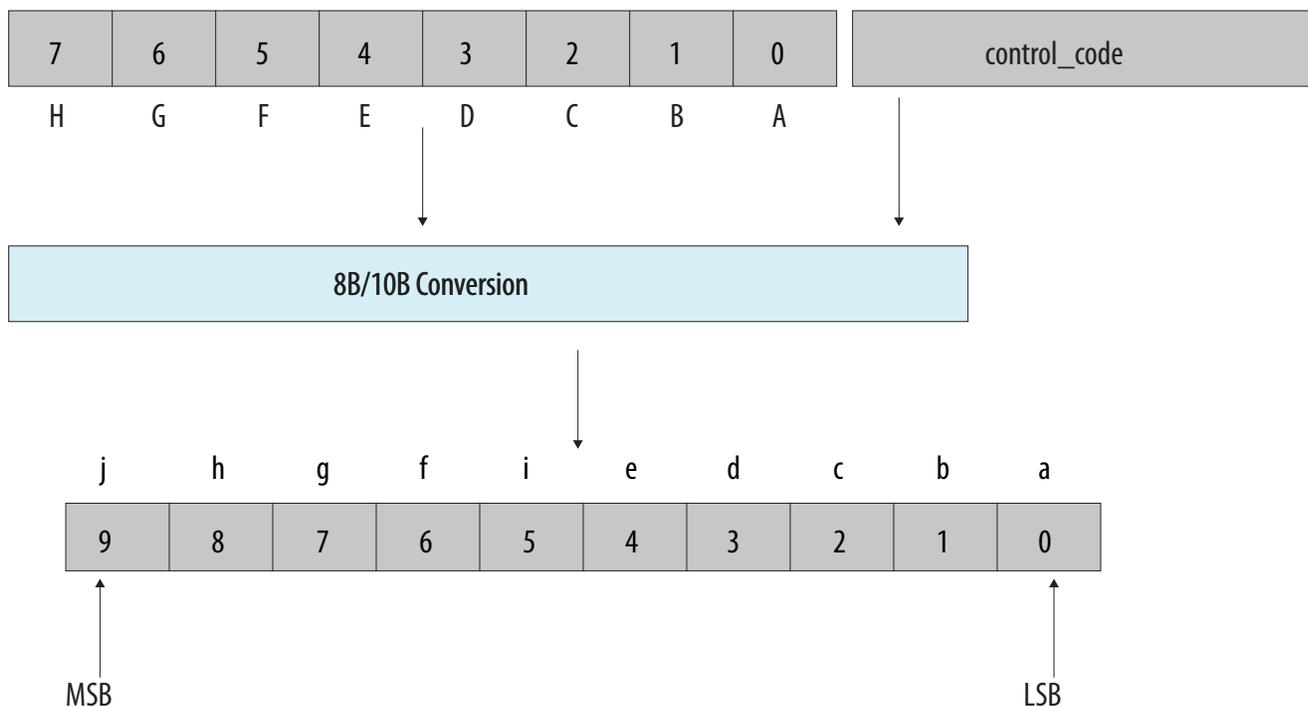
Byte Serializer

The byte serializer allows you to run the transceiver channel at higher data rates while keeping the FPGA fabric interface frequency below the upper limit, by halving the width of the data bus and doubling the data rate. The Byte Serializer forwards the least significant word first followed by the most significant word. For example, assuming a channel width of 32, the byte serializer forwards `tx_parallel_data[15:0]` first, followed by `tx_parallel_data[31:16]`.

8B/10B Encoder

The 8B/10B encoder generates 10-bit code groups from the 8-bit data and 1-bit control identifier. In 8-bit width mode, the 8B/10B encoder translates the 8-bit data to a 10-bit code group (control word or data word) with proper disparity. If the `tx_datak` input is high, the 8B/10B encoder translates the input `data[7:0]` to a 10-bit control word. If the `tx_datak` input is low, the 8B/10B encoder translates the input `data[7:0]` to a 10-bit data word.

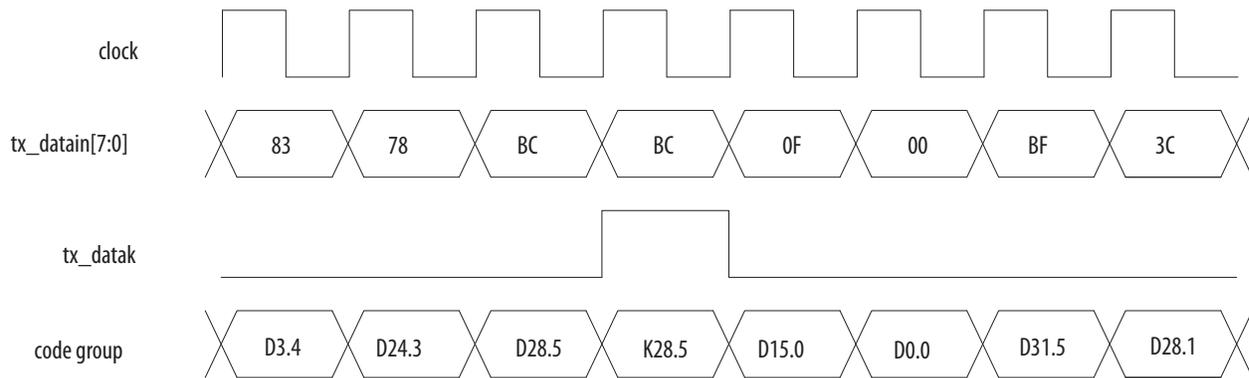
Figure 1-30: 8B/10B Conversion Format



Control Code Encoding

The IEEE 802.3 8B/10B encoder specification identifies only a set of 8-bit characters for which `tx_dataak` must be asserted. If you assert `tx_dataak` for any other set of bytes, the 8B/10B encoder might 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. It is possible for a downstream 8B/10B decoder to decode an invalid control word into a valid `Dx.y` code without asserting code error flags.

Figure 1-31: Control Word and Data Word Transmission



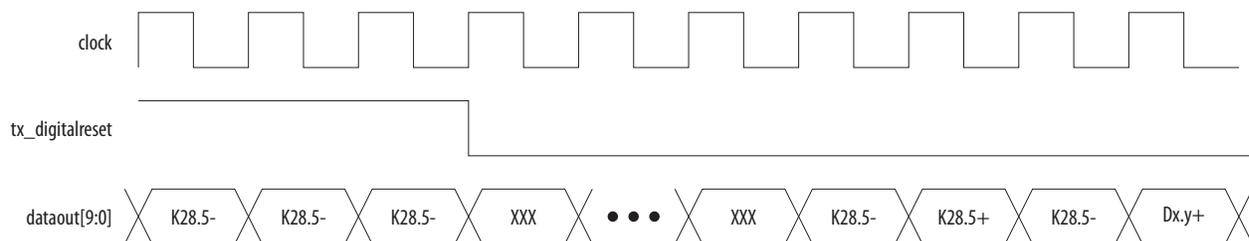
Reset Condition

The `tx_digitalreset` signal resets the 8B/10B encoder. During reset, running disparity (RD) and data registers are cleared. Also, the 8B/10B encoder outputs a K28.5 pattern from the RD- column continuously until `tx_digitalreset` is deasserted. The input data and control code from the FPGA fabric is ignored during the reset state. After reset, the 8B/10B encoder starts with an RD being negative (RD-) and transmits three K28.5 code groups for synchronization before it starts encoding and transmitting the data on its output.

Note: While `tx_digitalreset` is asserted, the downstream 8B/10B decoder that receives the data may observe synchronization or disparity errors.

When in reset (`tx_digitalreset` is high), a K28.5- (K28.5 10-bit code group from the RD-column) is sent continuously until `tx_digitalreset` is low. Because of some pipelining of the transmitter channel PCS, some “don’t cares” (10’hxxx) are sent before the three synchronizing K28.5 code groups. User data follows the third K28.5 code group.

Figure 1-32: 8B/10B Encoder Output During `tx_digitalreset` Deassertion



Transmitter Polarity Inversion

The positive and negative signals of a serial differential link may be erroneously swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic are expensive. The transmitter polarity inversion feature of the 8B/10B encoder is provided to correct this situation.

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 physical medium attachment (PMA).

Related Information

- For more information about the 8B/10B code, refer to the "8B/10B Code" section of the [Stratix II GX Specifications and Additional Information](#) chapter.
- For more information about enabling the polarity inversion and bit-slip features, refer to the "Bit Reversal and Polarity Inversion" section of the [Altera Transceiver PHY IP Core User Guide](#).

PRBS Generator

The PRBS generator block generates PRBS patterns and square wave patterns.

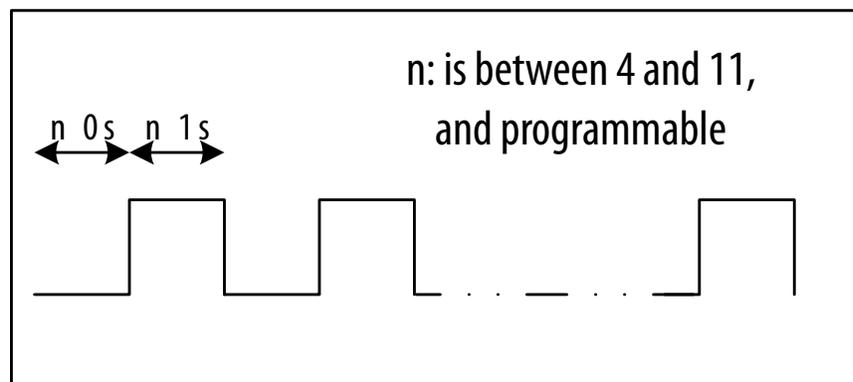
The PRBS generator:

- Supports 32-bit and 40-bit PMA interfaces
- Supports the following modes and patterns:
 - PRBS31: $x^{31} + x^{28} + 1$
 - PRBS9: $x^9 + x^5 + 1$
 - PRBS23: $x^{23} + x^{18} + 1$
 - PRBS7: $x^7 + x^6 + 1$
 - PRBS15: $x^{15} + x^{14} + 1$

The square-wave generator:

- Has programmable n-number of consecutive serial bit 1s and 0s (where $4 \leq n \leq 11$)
- Supports both 32-bit and 40-bit PMA widths

Figure 1-33: Square Wave Generator



Note: You can enable either the PRBS generator or the PRP generator, but you cannot enable both at the same time.

Related Information

For PRBS generator implementation information, refer to the [Altera Transceiver PHY IP Core User Guide](#)

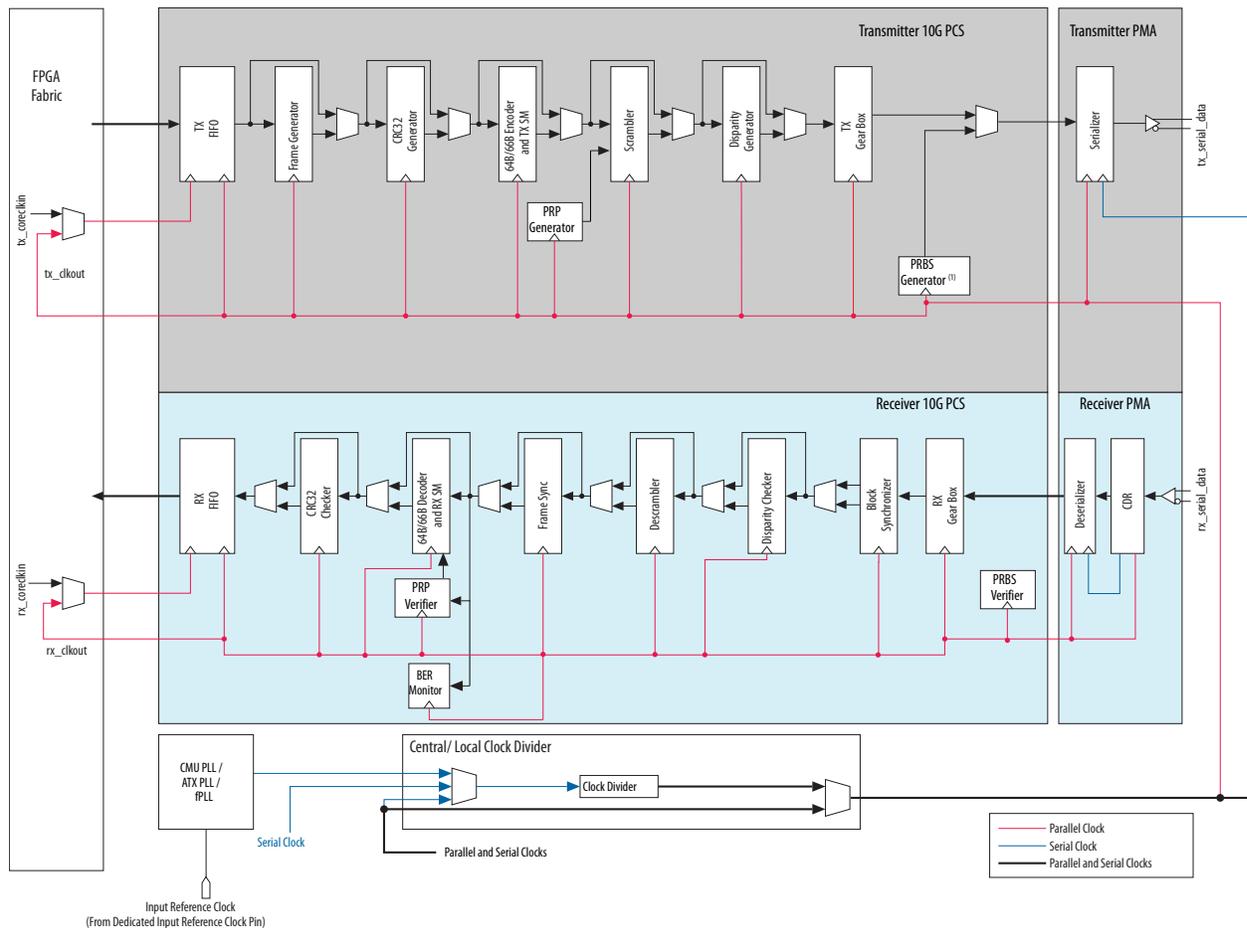
10G PCS Architecture

The 10G PCS architecture offers a full duplex (transmitter and receiver) transceiver channel that supports serial data rates up to 14.1 Gbps for Stratix V GX and GS devices and up to 12.5 Gbps for Stratix V GT 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 IP catalog.

Figure 1-34: 10G PCS Datapath in Stratix V GX Channels

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



Note:
1. The PRBS pattern generator can dynamically invert the data pattern that leaves the PCS block.

Related Information

- [For more information about 10G PCS configurations, and clocking schemes and placement restrictions for the different datapath configurations supported in Stratix V devices, refer to the Transceiver Configurations in Stratix V Devices chapter.](#)
- [For details about implementing the different PCS functional blocks, refer to the Altera Transceiver PHY IP Core User Guide](#)

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 PMA bus width is smaller than the physical coding sublayer (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 polarity inversion and receiver bit reversal features.

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

- [For more information about enabling the polarity inversion and bit reversal, refer to the "Bit Reversal and Polarity Inversion" section of the Altera Transceiver PHY IP Core User Guide.](#)
- [For more information about calculating expected latency as a function of the PCS or fabric clocks, refer to the "10G RX FIFO" section of the Altera Transceiver PHY IP Core User Guide.](#)

PRBS Verifier

The pseudo-random bit stream (PRBS) verifier block verifies the pattern generated by the PRBS generator.

The PRBS verifier:

- Supports 32-bit and 40-bit PMA interfaces
- Supports the following modes and patterns:
 - PRBS31: $x^{31} + x^{28} + 1$
 - PRBS23: $x^{23} + x^{18} + 1$
 - PRBS15: $x^{15} + x^{14} + 1$
 - PRBS9: $x^9 + x^5 + 1$
 - PRBS7: $x^7 + x^6 + 1$

Note: You can enable either the PRBS verifier or the PRP verifier, but you cannot enable both at the same time.

Related Information

For PRBS verifier implementation information, refer to the [Altera Transceiver PHY IP Core User Guide](#).

Receiver Inversion

The PRBS pattern verifier can dynamically invert the data pattern that enters the PCS block.

Table 1-10: PRBS Verifier Inversion Offset

Offset	Bits	R/W	Name	Description
0x16D	[2]	R/W	RX Inversion	Set to 1'b1 to invert the data entering the PCS block.

To invert the PRBS pattern entering the PRBS verifier:

1. Select the logical channel. In this case, logical channel 0.
2. Set the MIF streaming mode to 1.
3. Perform a read-modify-write to bit [2] of offset 0x16D.
4. Assert the channel reset to begin testing on the new PRBS pattern.

The inversion bit on the RX should be disabled to prevent normal data traffic from being inverted while entering the PCS.

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

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

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.

Table 1-11: Interpretation of the MSB in the 67-Bit Payload for Stratix 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

This function 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

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

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. After three consecutive mismatches, the frame synchronizer deasserts the lock signal and starts the synchronization process again. Lock status is available to the FPGA fabric.

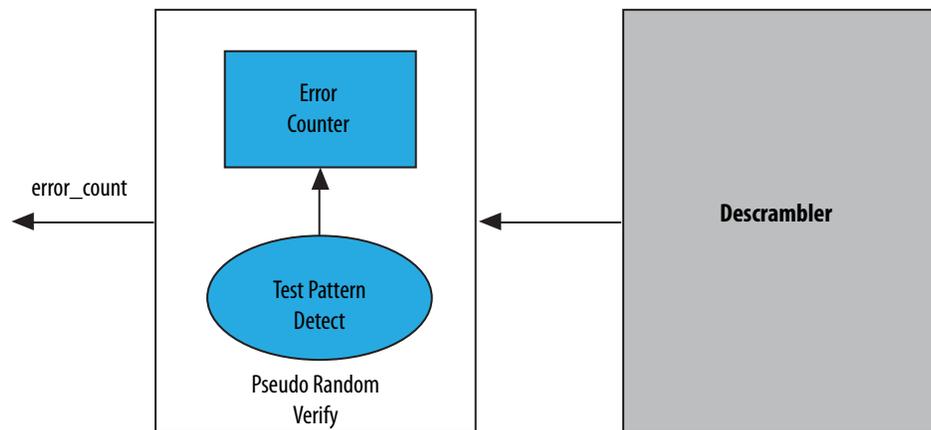
Bit-Error Rate (BER) Monitor

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- μ s period. If more than 16 invalid synchronization headers are observed in a 125- μ s period, the BER monitor provides the status signal to the FPGA fabric, indicating a high bit error rate condition.

PRP Verifier

The PRP verifier is available in Stratix V devices for the 10GBASE-R protocol mode. The PRP verifier monitors the output of the descrambler when block synchronization is achieved. You can enable this block for custom 10GBASE-R configurations. The PRP verifier:

- Searches for a test pattern (two local faults, or all 0's) or its inverse
- Tracks the number of mismatches with a 16-bit error counter

Figure 1-35: PRP Verifier Block Diagram

Note: You can enable either the PRP verifier or the PRBS verifier, but you cannot enable both at the same time.

Related Information

For implementation details, refer to the [Altera Transceiver PHY IP User Guide](#).

64B/66B Decoder

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

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.

The 64B/66B decoder block is designed in accordance with 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 Custom and Low Latency PHY IPs automatically select an appropriate receiver FIFO mode for the configuration you use; however, you should select the receiver FIFO mode when using the Native PHY IP.

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.

Note: Altera recommends a minimum of 32 words for the soft FIFO depth in the FPGA fabric for the following conditions:

- When the 10G PCS RX FIFO is set to register mode
- When using the recovered clock to drive the core logics
- When there is no soft FIFO being generated along with the IP Catalog

Related Information

For more information about the different receiver FIFO operating modes, refer to the [Transceiver Configurations in Stratix V Devices](#) chapter.

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.

Note: Altera recommends a minimum of 32 words for the soft FIFO depth in the FPGA fabric for the following conditions:

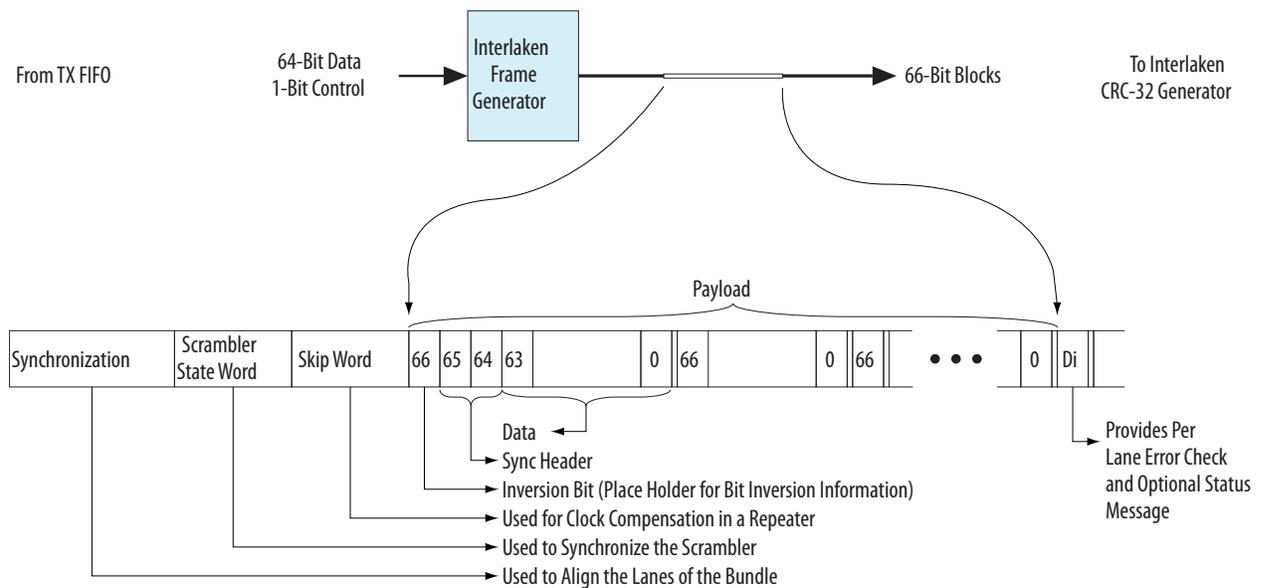
- When the 10G PCS TX FIFO is set to register mode
- When using the recovered clock to drive the core logics
- When there is no soft FIFO being generated along with the IP Catalog

Frame Generator

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

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 Parameter Editor allows you to set the metaframe length.

Figure 1-36: Frame Generator



CRC-32 Generator

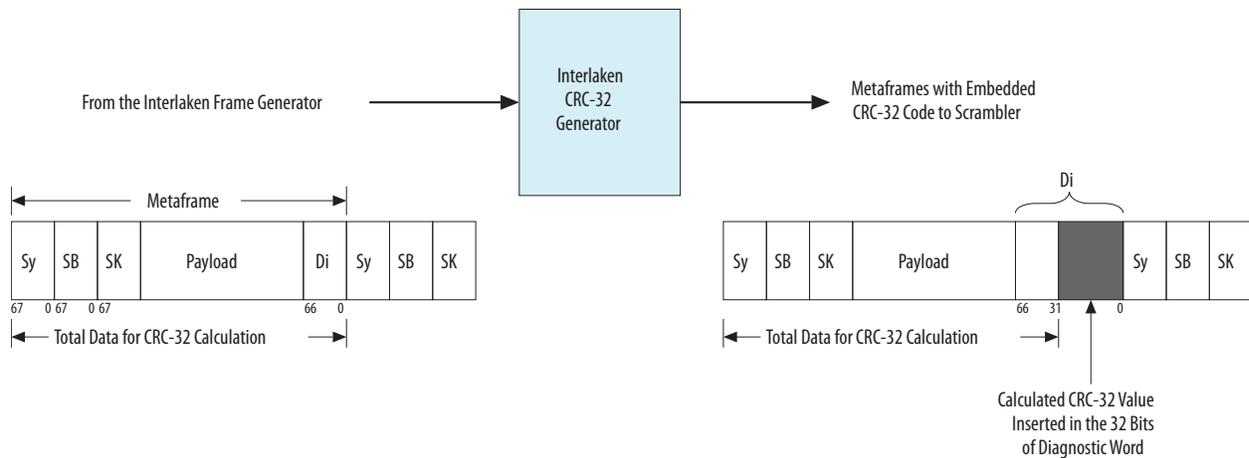
Note: The CRC-32 generator is used only in Interlaken configurations.

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.

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 1-37: CRC-32 Generator



64B/66B Encoder

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

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

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 is used in Interlaken configurations. Self synchronous mode operates in 10GBASE-R configurations, as specified in IEEE 802.3-2008 clause-49.

PRP Generator

The pseudo-random pattern (PRP) generator block operates in conjunction with the scrambler to generate pseudo-random patterns for the RX and TX tests in 10G Ethernet mode. It generates various test patterns from various seeds loaded to the scrambler and select data patterns. You can enable this block for custom 10GBASE-R configurations.

Note: You can enable either the PRP generator or the PRBS generator, but you cannot enable both at the same time.

Related Information

For details about seed and data pattern selection, refer to the [Altera Transceiver PHY IP Core User Guide](#).

Disparity Generator

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

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.

Table 1-12: Interpretation of the MSB in the 67-Bit Payload for Stratix 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

PRBS Generator

The PRBS generator block generates PRBS patterns and square wave patterns.

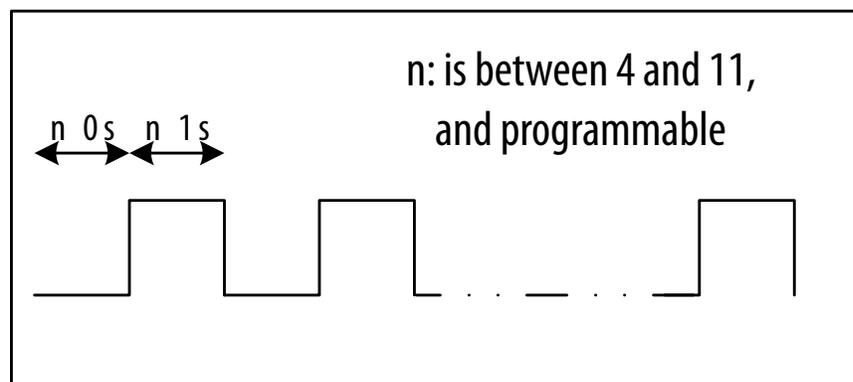
The PRBS generator:

- Supports 32-bit and 40-bit PMA interfaces
- Supports the following modes and patterns:
 - PRBS31: $x^{31} + x^{28} + 1$
 - PRBS9: $x^9 + x^5 + 1$
 - PRBS23: $x^{23} + x^{18} + 1$
 - PRBS7: $x^7 + x^6 + 1$
 - PRBS15: $x^{15} + x^{14} + 1$

The square-wave generator:

- Has programmable n-number of consecutive serial bit 1s and 0s (where $4 \leq n \leq 11$)
- Supports both 32-bit and 40-bit PMA widths

Figure 1-38: Square Wave Generator



Note: You can enable either the PRBS generator or the PRP generator, but you cannot enable both at the same time.

Related Information

For PRBS generator implementation information, refer to the [Altera Transceiver PHY IP Core User Guide](#)

Transmitter Inversion

The PRBS pattern generator can dynamically invert the data pattern that leaves the PCS block.

Table 1-13: PRBS Generator Inversion Offset

Offset	Bits	R/W	Name	Description
0x141	[0]	R/W	TX Inversion	Set to 1'b1 to invert the data leaving the PCS block.

To invert the PRBS pattern leaving the PRBS generator:

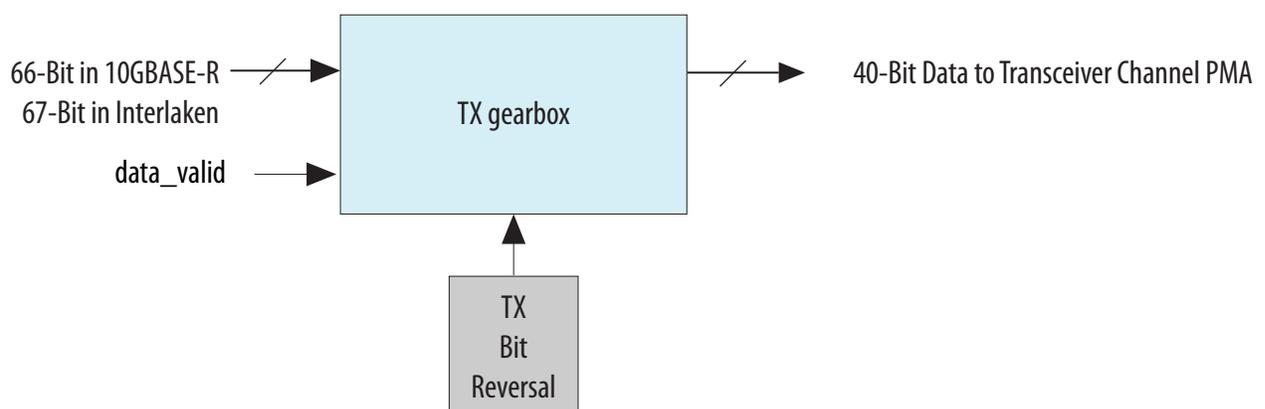
1. Select the logical channel. In this case, logical channel 0.
2. Set the MIF streaming mode to 1.
3. Perform a read-modify-write to bit [0] of offset 0x141.
4. Assert the channel reset to begin testing on the new PRBS pattern.

The inversion bit on the TX should be disabled to prevent normal data traffic from being inverted while leaving the PCS.

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 1-39: Transmitter Gearbox



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

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 Reversal

The transmitter gearbox can reverse the order of transmitted bits. By default, the transmitter sends out the LSB of a word first. 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 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.

The transmitter bit-slip is not supported for all PHYs. Low latency PHY does not allow this feature.

Related Information

- [For the supported PMA-PCS widths and for custom configurations, refer to the Transceiver Configurations in Stratix V Devices chapter.](#)
- [For more information about enabling the polarity inversion and bit reversal, refer to the "Bit Reversal and Polarity Inversion" section of the Altera Transceiver PHY IP Core User Guide.](#)
- [For more information about calculating expected latency as a function of the PCS or fabric clocks, refer to the "10G TX FIFO" section of the Altera Transceiver PHY IP Core User Guide.](#)

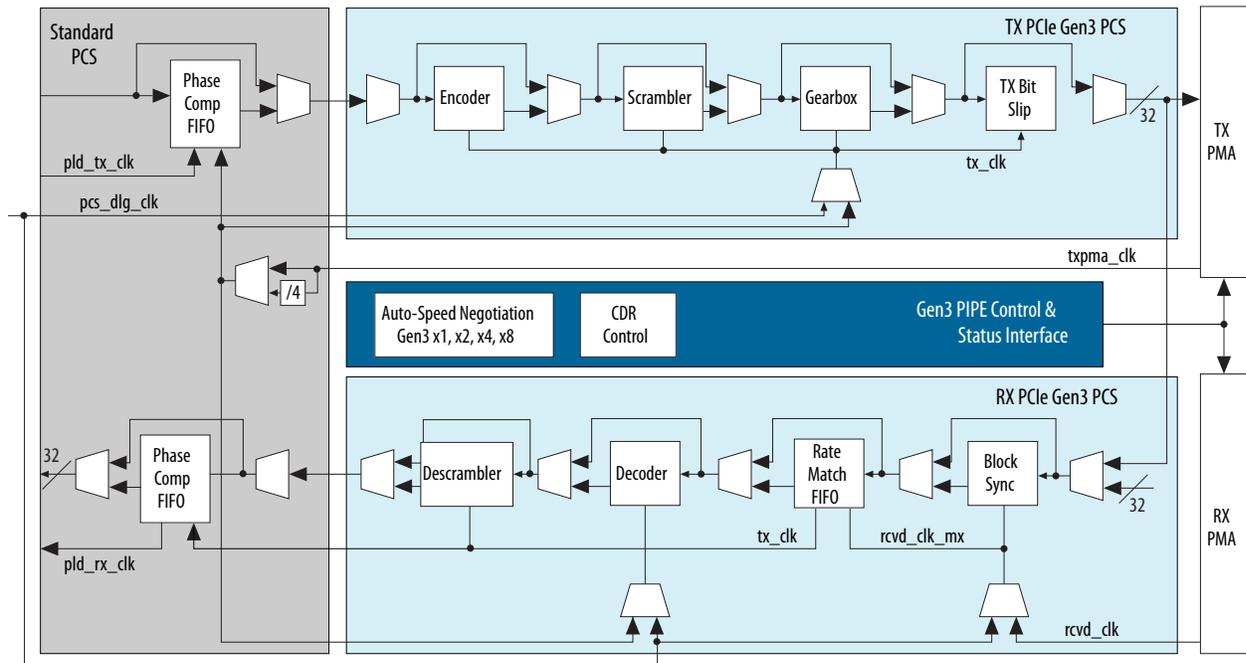
PCIe Gen3 PCS Architecture

Stratix V architecture supports the PCIe Gen3 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, Stratix V devices include a separate Gen3 PCS that supports functionality at Gen3 speeds. You can use Altera hard IP and interface to the transceivers, or you can implement your MAC and connect it to the transceiver through the PIPE interface.

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.

Figure 1-40: PCIe Gen3 PCS Top Level Block Diagram

The RX/TX Phase Comp FIFOs are physically placed in, and shared with, the Standard PCS.



Receiver PCIe Gen3 PCS Datapath

The receiver channel PCIe Gen3 PCS datapath architecture is described from block sync to phase compensation FIFO.

Block Synchronizer

PMA parallelization occurs at arbitrary word boundaries. Consequently, the parallel data from the RX PMA CDR needs to 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 the FPGA soft IP clock or FPGA 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.

The Rate Match FIFO is fully compliant with the GigE and PCI-Express (Gen1 and Gen2) protocols. 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
- 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

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

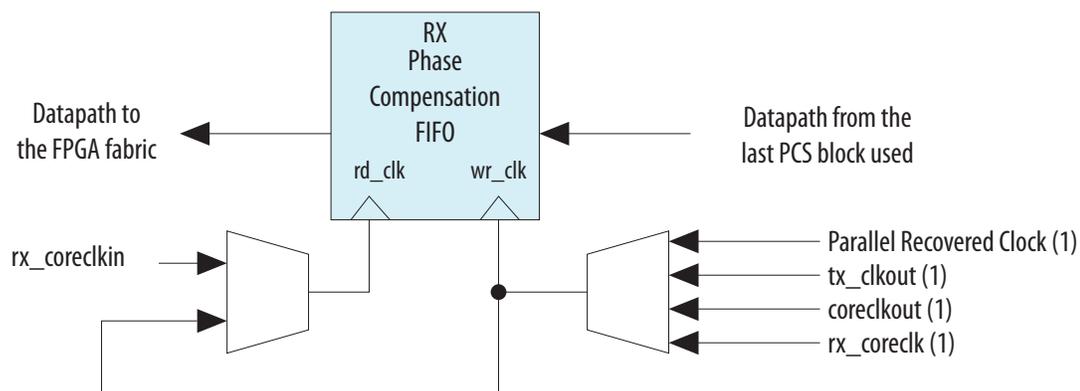
In a multi-lane link environment, each of the receiver lanes may implement a separate linear feedback shift register (LFSR) for de-scrambling. The LFSR uses the following polynomial: $G(X) = X^{23} + X^{21} + X^{16} + X^8 + X^5 + X^2 + 1$. It is a standard PRBS23 polynomial.

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

Figure 1-41: Receiver Phase Compensation FIFO



Note:

1. If you use the byte deserializer, these clocks are divided by two.

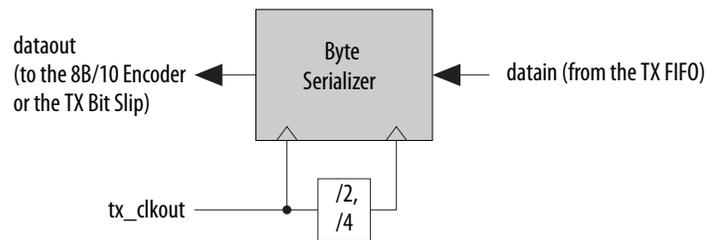
Transmitter PCIe Gen3 PCS Datapath

The transmitter channel PCIe Gen3 PCS datapath is described from phase compensation FIFO to gearbox.

Transmitter Phase Compensation FIFO

The transmitter phase compensation FIFO interfaces with the FPGA fabric. The transmitter phase compensation FIFO compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock.

Figure 1-42: Transmitter Phase Compensation FIFO



Scrambler

In a multi-lane link environment, each of the transmitter lanes may implement a LFSR for scrambling. The LFSR uses the following polynomial: $G(X) = X^{23} + X^{21} + X^{16} + X^8 + X^5 + X^2 + 1$. It is a standard PRBS23 polynomial. The scrambler is used to provide enough edge density, since there is no 8B/10B encoding in PCIe Gen3, so that the RX PMA CDR can lock to the incoming data stream and generate the recovered clock.

Encoder

The PCIe Gen3 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 a 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 takes significant resources, so the PCIe Gen3 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.

PIPE Interface

The PIPE Data interface on the hard transceiver is compatible with the PIPE 3.0 specification. Advanced equalization signals are not provided in Altera's PIPE interface.

Auto Speed Negotiation

Auto Speed Negotiation controls the operating speed of the transceiver when operating under PIPE 3.0 modes. By monitoring the rate control signal from the physical MAC (PHY MAC) layer, this feature changes the transceiver from PCIe Gen1 operation mode to Gen2 operation mode, or from PCIe Gen1 operation mode to Gen2 operation mode to Gen3 operation mode, or vice versa, with all the appropriate settings.

Electrical Idle Inference

In conjunction with side band signals from the FPGA side, the Electrical Idle Inference feature 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

The CDR control feature is used for Rx.L0s fast exit when operating in PIPE/PCIe Gen3 mode. After detecting an Electrical Idle Ordered Set (EIOS), it takes manual control of the CDR by forcing it into a lock-to-reference (LTR) mode. When an exit from electrical idle is detected, this feature moves the CDR into lock-to-data (LTD) mode to achieve fast data lock.

Document Revision History

The revision history for this chapter.

Table 1-14: Document Revision History

Date	Version	Changes
January 2016	2016.01.11	Made the following changes: <ul style="list-style-type: none">Added a note to the "Receiver FIFO" section.Added a note to the "Transmitter FIFO" section.

Date	Version	Changes
September 2014	2014.09.30	<ul style="list-style-type: none"> • Added a link to Altera mySupport in the <i>Link Coupling</i> section. • Added a note to the <i>10G PCS Datapath in Stratix V GX Channels</i> figure. • Added a note to the <i>Rate Match (Clock Compensation) FIFO</i> section. • Added the "Word Aligner in Deterministic Latency State Machine Mode" section and updated the Manual Mode description in the <i>Word Aligner</i> section. • Removed the values for Word Alignment Pattern Length in Bit-Slip mode in the <i>Word Aligner Options</i> table. • Removed XAUI Mode and PCIe Mode from the <i>Receiver Phase Compensation FIFO</i> section. • Added the following columns to the Transceiver Calibration Block Boundary for Stratix V Devices table: <ul style="list-style-type: none"> • Package • Total Number of Transceiver channels in device • Total Number of Transceiver Channels per Side • Changed the description of the receiver phase compensation FIFO. • Added the <i>Phase Compensation Mode</i> section. • Added the <i>Registered Mode</i> section. • Added the <i>Receiver Inversion</i> section. • Added the <i>Transmitter Inversion</i> section. • Changed "MegaWizard Plug-in Manager" to "IP catalog" in the <i>10G PCS Architecture</i> section. • Updated the <i>PCIe Gen3 PCS Top Level Block Diagram</i> to show the <code>p1d_rx_clk</code> as an input to the Phase Compensation FIFO block.
January 2014	2014.01.07	<ul style="list-style-type: none"> • Updated the <i>Stratix V GX/GT Channel and PCIe Hard IP Layout</i> section. • Updated the <i>Stratix V GS Channel and PCIe Hard IP Layout</i> section. • Updated the <i>Channel Variants</i> section. • Updated the <i>GS/GT/GX Device Variants and Packages</i> section. • Updated the <i>Receiver Equalizer Gain Bandwidth</i> section.

Date	Version	Changes
October 2013	2013.10.11	<ul style="list-style-type: none"> Updated the <i>Word Aligner</i> section. Updated the <i>Lock-to-Reference Mode</i> section.
May 2013	2013.05.06	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base Updated Figure 1-10. Updated Figure 1-15. Updated the <i>Receiver Deserializer</i> section. Updated the <i>Continuous Time Linear Equalization</i> section. Added the <i>GS/GT/GX Device Variants and Packages</i> section. Added the <i>Stratix V GS Channel and PCIe Hard IP Layout</i> section. Updated Figure 1-17. Updated Figure 1-18. Updated Figure 1-30. Added the <i>PRBS Verifier</i> section. Added the <i>PRBS Generator</i> section. Added the <i>PRP Verifier</i> section. Added the <i>Serial Bit Checker</i> section. Updated the <i>Decision Feedback Equalization</i> section. Updated the <i>Transmitter Analog Settings</i> section. Updated the <i>Receiver PMA Bit-Slip</i> section. Updated the <i>ATX PLL Calibration</i> section. Updated the <i>Calibration Block Boundary</i> section. Updated Figure 1-20. Updated the <i>8B/10B Decoder</i> section. Updated the <i>Transmitter Phase Compensation FIFO</i> section.
December 2012	2012.12.17	Reorganized content and updated template .

Date	Version	Changes
June 2012	2.3	<ul style="list-style-type: none">• Updated Figure 1–6, Figure 1–10, and Figure 1–11.• Updated Table 1–3• Updated “Stratix V Device Layout”, “PMA Architecture”, “Standard PCS Architecture” and “10G PCS Architecture” sections.• Updated Table 1–2, Table 1–4, Table 1–1, and Table 1–5.• Updated Figure 1–1, Figure 1–3, Figure 1–4, Figure 1–8, and Figure 1–21.• Updated “Transmitter Polarity Inversion” section.• Added “PCIe Gen3 PCS Architecture” section.
February 2012	2.2	<ul style="list-style-type: none">• Updated Figure 1–1.• Updated “Transmitter Polarity Inversion” section.

2016.01.11

SV52003



Subscribe



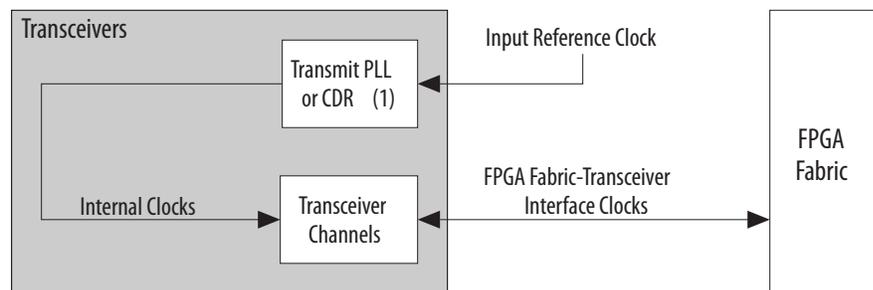
Send Feedback

This chapter provides information about the Stratix 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 (Auxiliary Transmit PLL).

Related Information

[Stratix V Device Handbook: Known Issues](#)

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

Input Reference Clocking

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

Each transceiver channel has a channel PLL that can be configured as a transmitter clock multiplier unit (CMU) PLL or a receiver CDR PLL. In the CMU PLL configuration, the channel PLL uses the input reference clock to generate a serial clock. In the receiver CDR PLL configuration, the channel PLL locks to

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

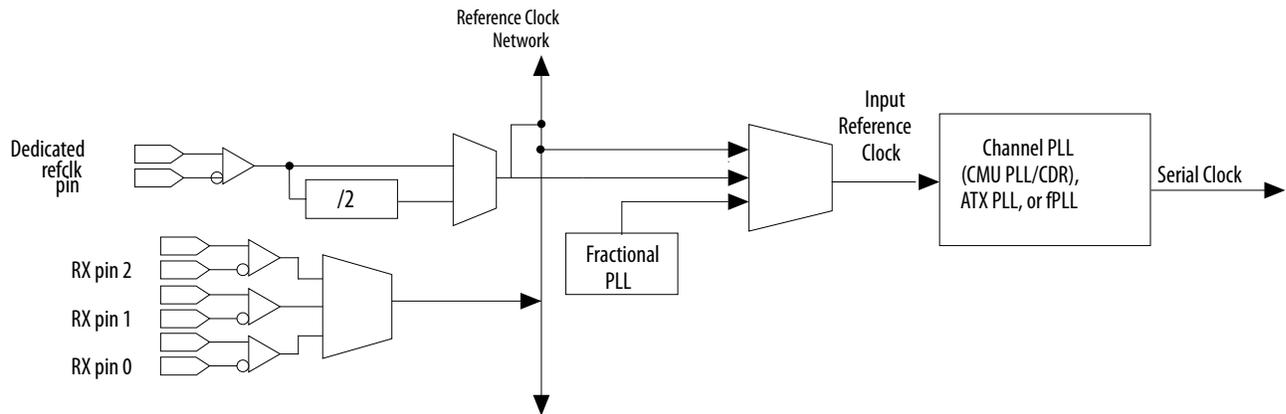
ALTERA
now part of Intel

the input reference clock in lock-to-reference (LTR) mode. The auxiliary transmit (ATX) PLL and the fractional PLL use the input reference clock to synthesize a serial clock.

Input Reference Clock Sources

The channel PLL, ATX PLL, and fractional PLL can derive the input clock from a dedicated `refclk` pin, another fractional PLL, or through the reference clock network.

Figure 2-2: Input Reference Clock Sources to Transmit PLLs and CDR

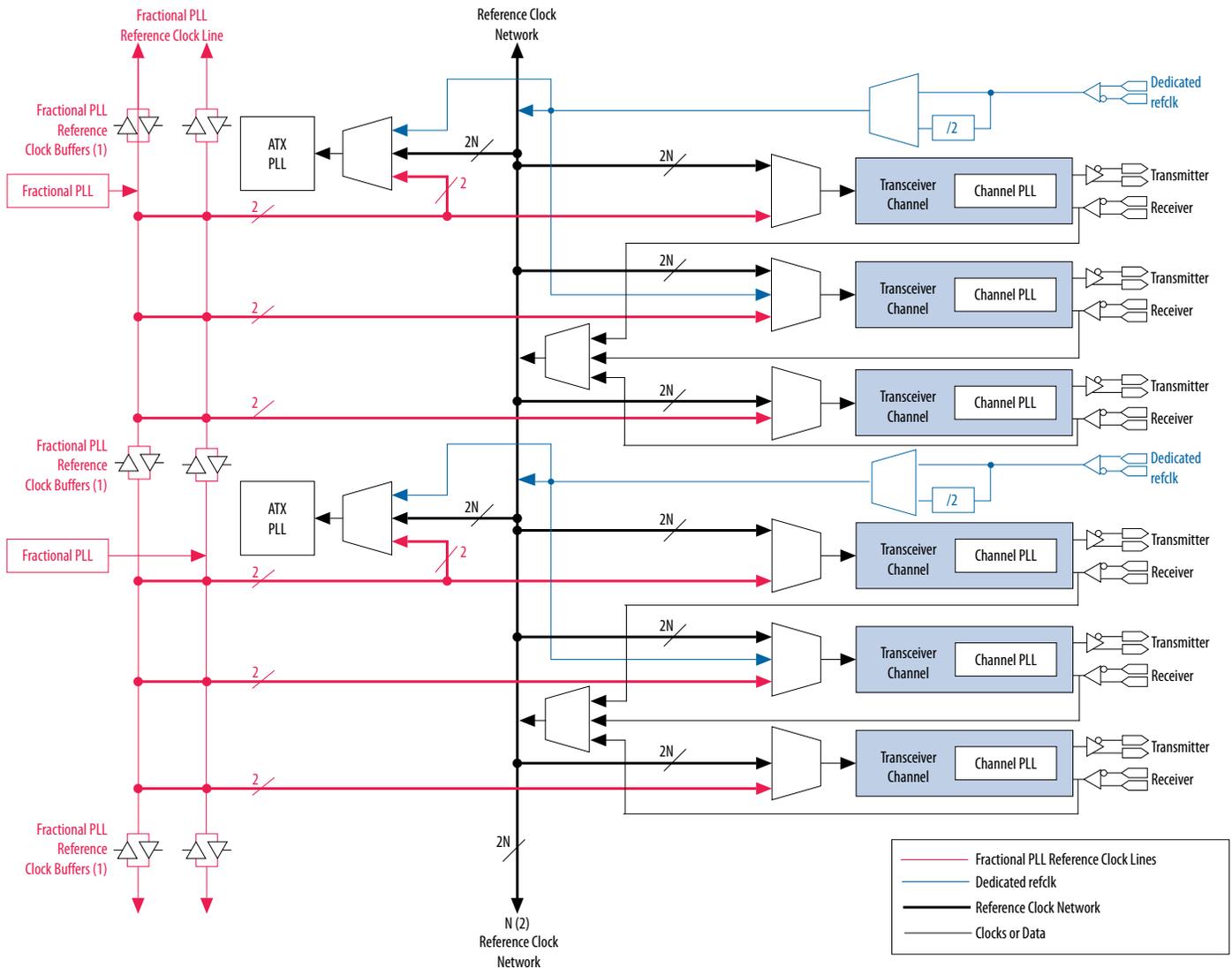


Note : You can choose only one of the three RX pins to be used as a reference clock source.

Note: For optimal performance, use the `refclk` source that is closest to the transmit PLL in the same transceiver bank.

Figure 2-3: Input Reference Clock Sources for GX Transceiver Channels

For more information about the fractional PLL input clock sources shown in the following figure, refer to **Figure 2-6**.

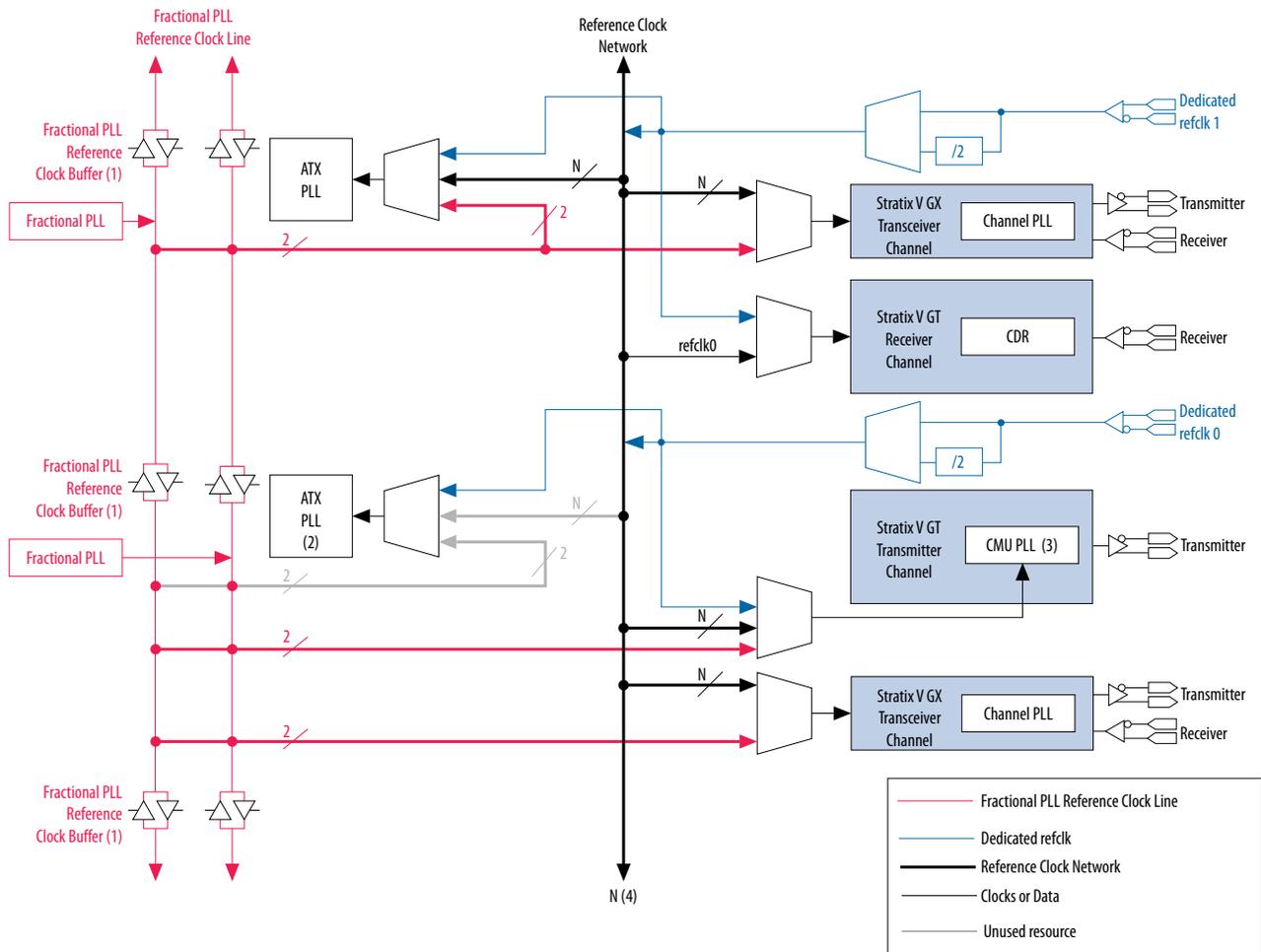


Notes: (1) The fractional PLL refclk buffers allow you to segment the reference clock line into multiple segments, such that fractional PLLs in different transceiver banks can drive the same fractional PLL reference clock line.
(2) N equals the number of transceiver channels on a side divided by 3, which is equal to the number of dedicated refclk pins.

The following figure shows the input reference clock sources for a GT transceiver channel and two GX transceiver channels in a GT transceiver bank.

Figure 2-4: Input Reference Clock Sources for GT and GX Transceiver Channels in Stratix V GT Devices

For more information about the fractional PLL input clock sources shown in the following figure, refer to [Figure 2-6](#).



Notes:

- (1) The fractional PLL refclk buffers allow you to segment the reference clock line into multiple segments, such that fractional PLLs in different transceiver banks can drive the same fractional PLL reference clock line.
- (2) The bottom ATX PLL of a GT transceiver bank provides the serial clock to the GT transmitter channel.
- (3) The CMU PLL of the GT transmitter channel drives an x1 clock line that can be used by the top and bottom GX transceiver channels in the GT transceiver bank.
- (4) N equals twice the number of GT channels.

Note: Altera recommends using a dedicated clock `refclk0` for the bottom ATX PLL that provides the serial clock to the GT transmitter channel.

Dedicated refclk Pins

GX transceiver banks have one dedicated `refclk` pin for each group of three transceiver channels. The dedicated `refclk0/refclk1` pins can drive reference clock network or `ch1/ch4` channel PLLs respectively in a transceiver bank.

There are two dedicated `refclk` pins available in each GT transceiver bank. The two `refclk` pins can also provide the reference clocks to the GX channels in a GT transceiver bank through the reference clock network.

Power pins associated with transceiver banks must be powered up. At least one transceiver must be instantiated in the design if a dedicated transceiver `refclk` pin is used as a clock reference for a core fPLL.

The following table lists the electrical specifications for the input reference clock signal driven on the `refclk` pins.

Table 2-1: Electrical Specifications for the Input Reference Clock

Protocol	I/O Standard	Coupling	Termination
PCI Express (PCIe)	<ul style="list-style-type: none"> 1.2V PCML, 1.4 PCML 1.4V PCML 1.5V PCML 2.5V PCML Differential LVPECL LVDS 	AC	On - Chip ⁽¹⁾
	<ul style="list-style-type: none"> HCSL ⁽²⁾ 	DC	Off - Chip ⁽³⁾
All other protocols	<ul style="list-style-type: none"> 1.2V PCML, 1.4 PCML 1.4V PCML 1.5V PCML 2.5V PCML Differential LVPECL LVDS 	AC	On - Chip ⁽¹⁾

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):

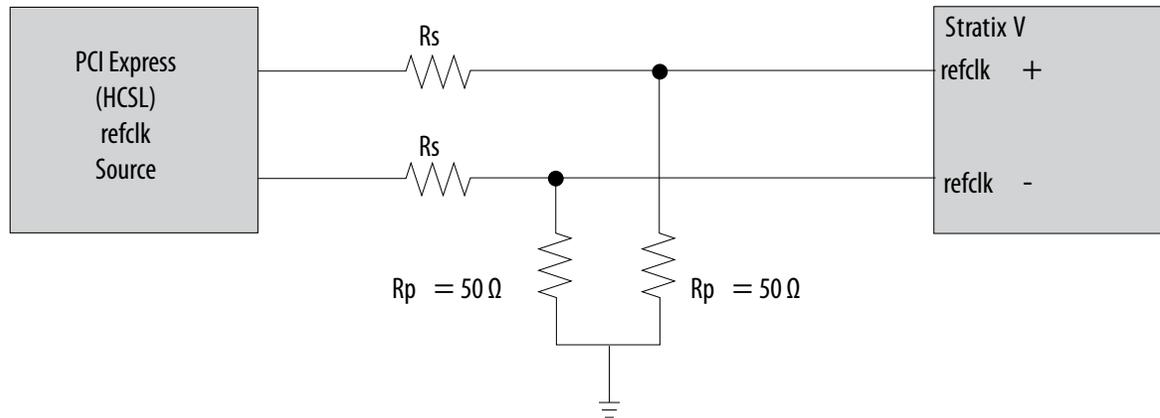
```
set_instance_assignment -name XCVR_REFCLK_PIN_TERMINATION_DC_COUPLING_EXTERNAL_RESISTOR -to <refclk_pin_name>
```

⁽¹⁾ For more information about termination values supported, refer to the *DC Characteristics* section in *Stratix V Device Datasheet*.

⁽²⁾ 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.

⁽³⁾ For an example termination scheme, refer to [Figure 2-5](#)

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



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

Related Information

[Stratix V Device Datasheet.](#)

Dedicated refclk Pins Using the Reference Clock Network

Each dedicated `refclk` pin can drive any transmitter PLL on the same side of the device through the reference clock network. Designs using multiple transmitter PLLs that require the same reference clock frequency and are located along the same side of the device can share the same dedicated `refclk` pin.

RX Pins Using the Reference Clock Network

The RX pins can be used as `refclk` pins. The RX pins can drive any transmitter PLL on the same side of the device through the reference clock network. Only one RX differential pin pair per three channels can be used as a reference clock and there is no $/2$ factor available, unlike the dedicated reference clock pin, as shown in the [Figure 2-3](#).

Note: For more information about the QSF assignments, refer to the *Altera Transceiver PHY IP Core User Guide* and the *Stratix V Device Datasheet* for the supported I/O standards.

Related Information

- [Altera Transceiver PHY IP Core User Guide](#)
- [Stratix V Device Datasheet](#)

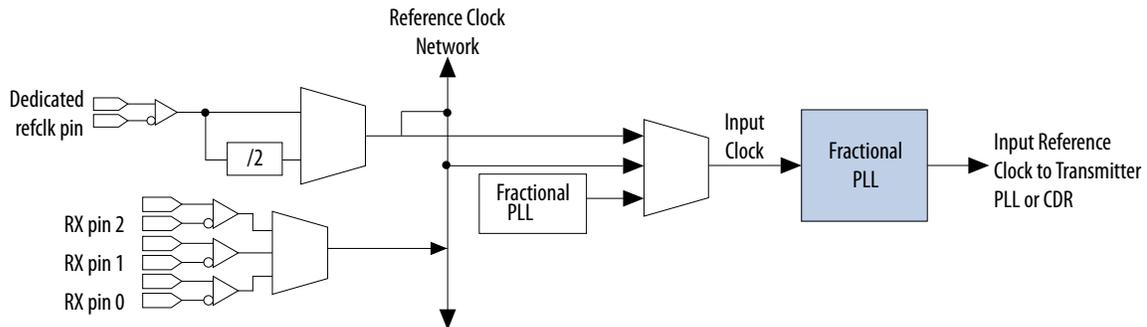
Fractional PLLs

Stratix V devices provide a fractional PLL for each group of three transceiver channels.

Each fractional PLL drives one of two clock lines spanning the side of the device that can provide an input reference clock to any transmitter PLL or CDR on the same side of the device. A fractional PLL enables you to use an input reference clock in your system that is not supported by the transmitter PLL or CDR to synthesize a supported input reference clock.

Figure 2-6: Fractional PLL Input Clock Sources

The following figure shows the input clock sources for the fractional PLLs located within the transceiver banks.



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

Internal Clocking

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

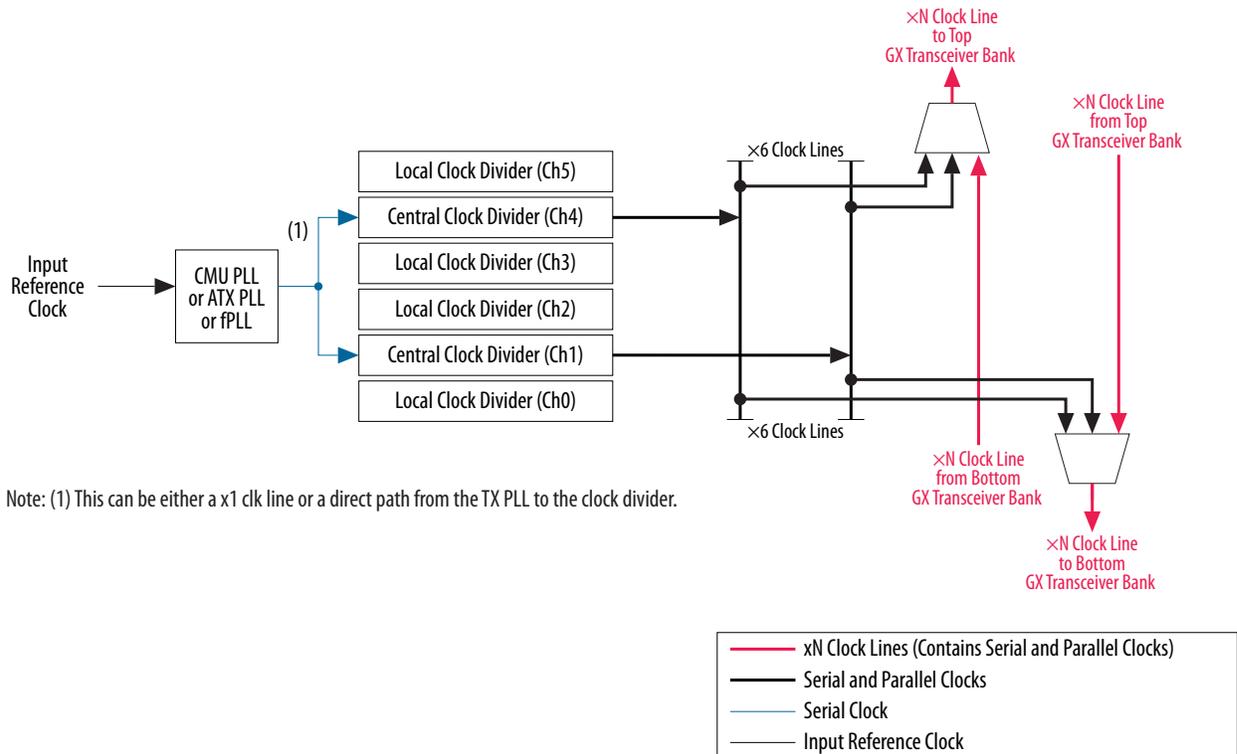
Table 2-2: Internal Clocking Subsections

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 2-8: Transmitter Clock Network

The following figure shows the transceiver clock network, beginning with the input reference clock, followed by the transmitter PLL, clock dividers, and ending with the x6 and xN clock lines.



Note: For more information about bonding, refer to the “Bonded Configurations” section of the *Transceiver Architecture in Stratix V Devices* chapter.

Related Information

Transceiver Architecture in Stratix V Devices

Transmitter Clock Lines

The transmitter clock network consists of two types of dedicated clocking resources.

The dedicated clocking resources are:

- Non-bonded configurations
 - x1 clock lines
 - xN clock lines (in non-bonded configurations available only for Native PHY)
- Bonded configurations (not available for GT transceiver channels)
 - x6 clock lines
 - x6 PLL Feedback Compensation
 - xN clock lines (available only for PCIe and Native PHY)

Note: The Quartus II software performs the clock routing related to the transmitter clock network based on the transceiver configuration selected.

Table 2-3: Data Rates and Spans Supported Using Stratix V Clock Sources and Clock Networks

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
x1	GX	ATX PLLs in a transceiver bank	14.1 Gbps ⁽⁴⁾	No	Transceiver bank
		CMU PLLs in a transceiver bank	12.5 Gbps ⁽⁴⁾		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 data rate 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		
x1	GT	Bottom ATX PLL in a GT transceiver bank	28 Gbps ⁽⁴⁾	No	Transceiver bank

⁽⁴⁾ For the fastest speed grade only. For the remaining speed grades, refer to the *Stratix V Device Datasheet*.

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
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 parallel clock from the central clock dividers.	14.1 Gbps ⁽⁴⁾	Yes	Transceiver bank
		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 ⁽⁴⁾		
		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 ⁽⁵⁾		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.	14.1 Gbps ⁽⁴⁾	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 ⁽⁴⁾		

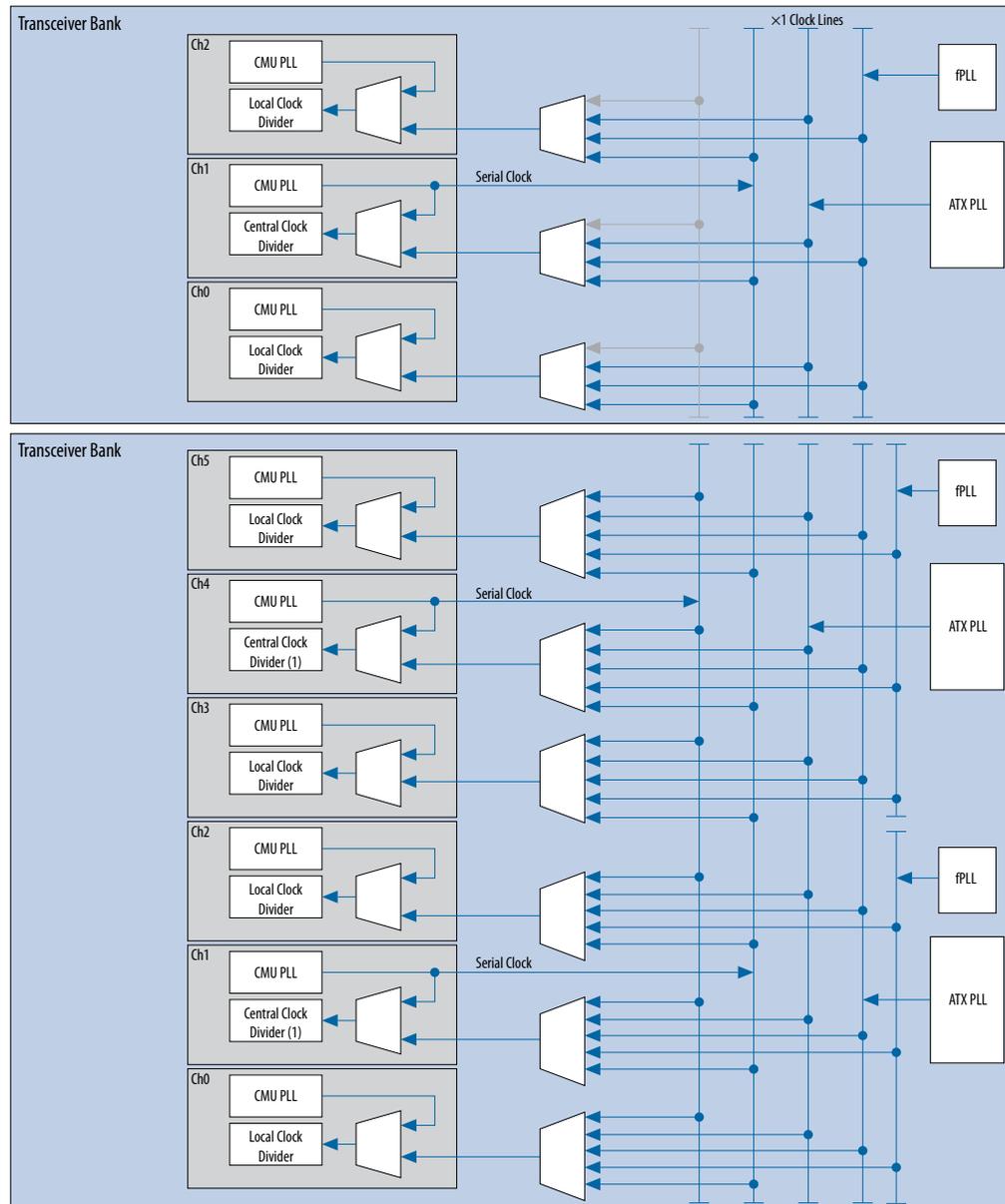
⁽⁵⁾ The input reference clock frequency of the transmit PLL must be the same as the parallel clock frequency which clock the PCS bonded channels.

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
xN (PCIe) ⁽⁶⁾	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 ⁽⁴⁾	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
	GX	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.		

⁽⁶⁾ For more information about PCIe x8 configurations, refer to the section titled “Hard IP x8 Configuration” in the *Transceiver Configurations in Stratix V Devices* chapter.

Note: Stratix V devices 5SGXB5, 5SGXB6, 5SGSB, and 5SGSB have one transceiver bank on each side with only three transceiver channels. For more information, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

Figure 2-9: x1 Clock Lines Used by GX Transmitter Channels for Non-Bonded Configuration



Note: (1) You can use the central clock divider as a local clock divider.

The x1 clock lines route the serial clock to the clock dividers of any channel within a transceiver bank. Refer to [Table 2-3](#) for details. The channel PLL, if configured as a CMU PLL, can drive the clock divider of its own channel, but you will not be able to use the channel PLL as a CDR. Without a CDR, you can use the channel only as a transmitter channel.

The x6 clock lines are used for bonded configurations within transceiver banks and PLL Feedback Compensation when bonding across multiple transceiver banks. The x6 clock lines are also used to route

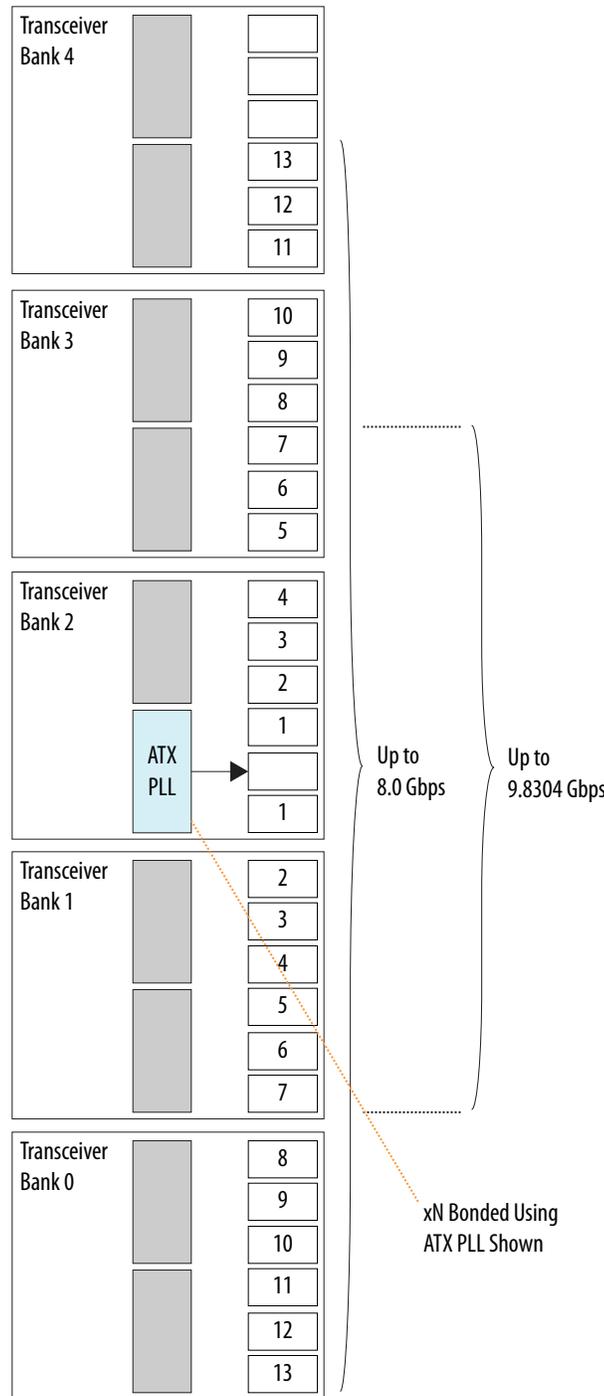
both the serial clock and parallel clock from the central clock dividers to the transceiver channels. When spanning across multiple transceiver banks, the xN clock lines can be used for both non-bonded configuration and bonded configurations.

The central clock dividers of channel 1 and channel 4 in a transceiver bank drive the x6 clock lines. The x6 clock lines then drive the xN clock lines. The xN clock lines used for both non-bonded and bonded configurations, span the entire side of the device and can provide the serial and parallel clock to contiguous channels within or outside a transceiver bank.

For both xN bonded and xN non-bonded configurations, the xN clock lines can support up to 13 contiguous channels above and up to 13 contiguous channels below the selected transmitter PLL which drives the central clock dividers of channel 1 or channel 4 of the same transceiver bank.



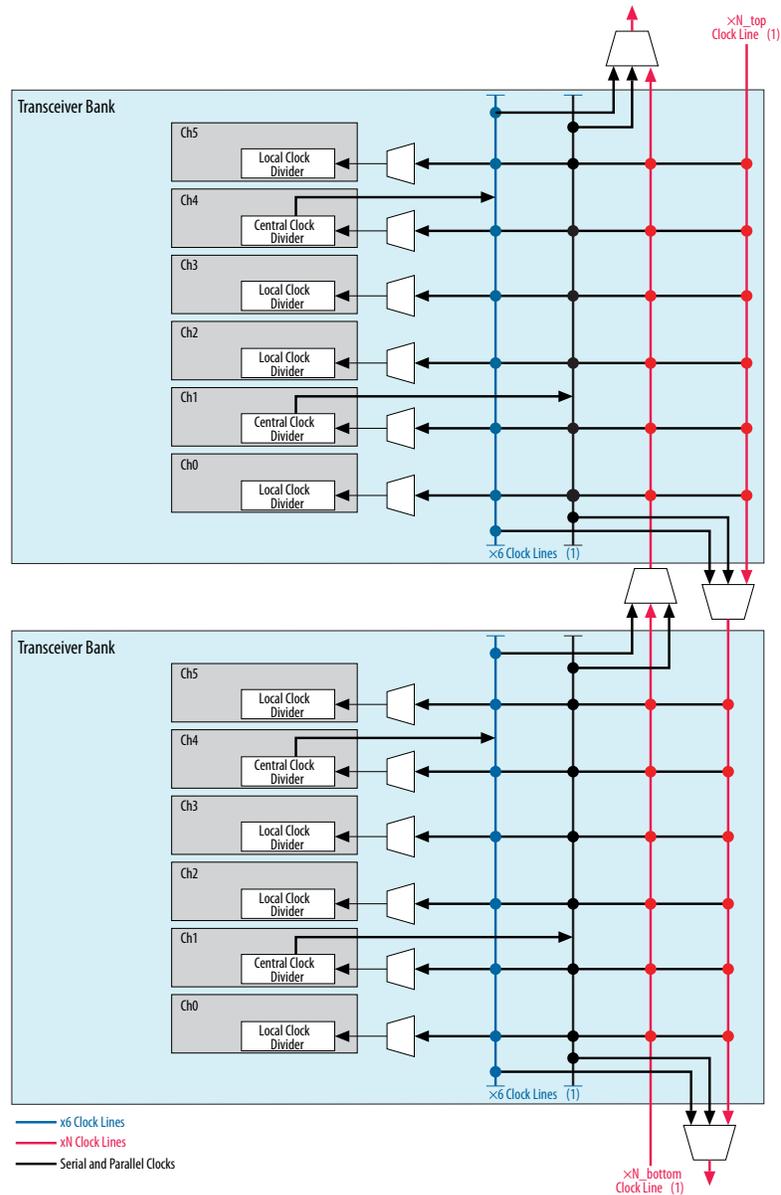
Figure 2-10: Channel Span for xN Bonded and Non Bonded Configurations



For xN bonded configurations, the channel where the central clock divider resides (channel 1 or 4) can be used as a data channel. Hence, a total of up to 27 contiguous data channels can be supported in the bonded configuration with the xN clock lines. However, for xN non-bonded configurations, the channel 1 or channel 4 of the 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. Hence, a total of up to 26 contiguous data channels can be supported in the non-bonded configuration with the xN clock lines.

Figure 2-11: x6 and xN Clock Lines Used for Bonded Configurations



Related Information

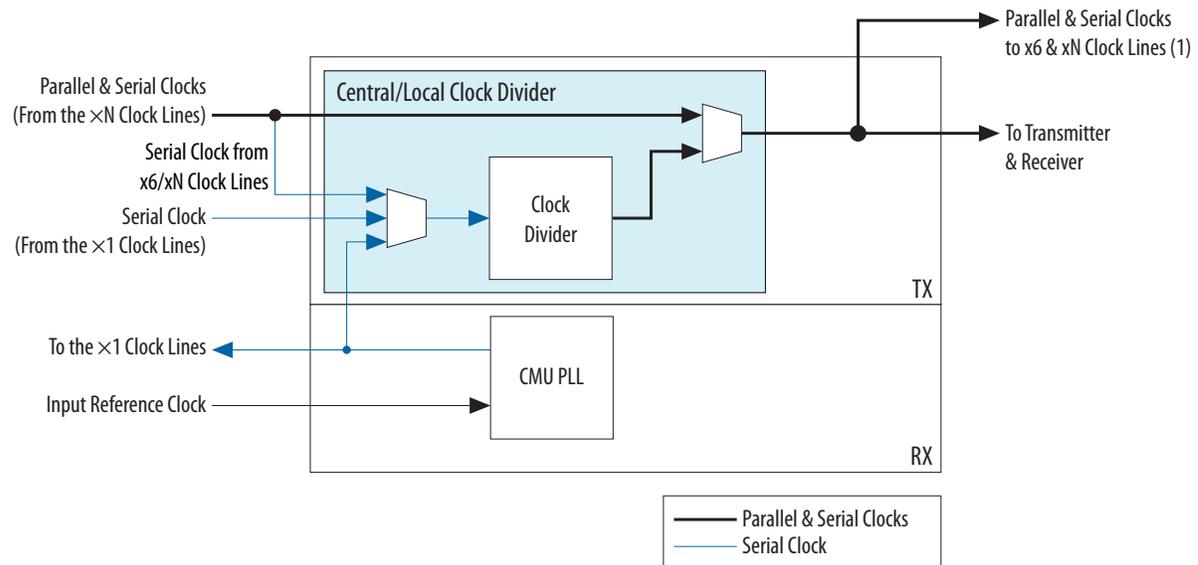
- [Stratix V Device Datasheet](#)
- [Transceiver Configurations in Stratix V Devices](#)
- [Transceiver Architecture in Stratix V Devices](#)

Clock Dividers

Each transmitter channel has a local clock divider. Some of the clock dividers that drive the x6 and xN clock lines are called central clock dividers.

Central clock dividers are located in channels 1 and 4 of the GX transceiver bank. The clock dividers generate the parallel and serial clock sources for the transmitter and optionally for the receiver PCS. The central clock dividers can feed the clock lines used to bond channels.

Figure 2-12: Clock Dividers



Note: (1) This is available only for the central clock divider in channels 1 and 4.

Note: For more information about clock dividers and the division factors supported, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

Related Information

[Transceiver Architecture in Stratix V Devices](#)

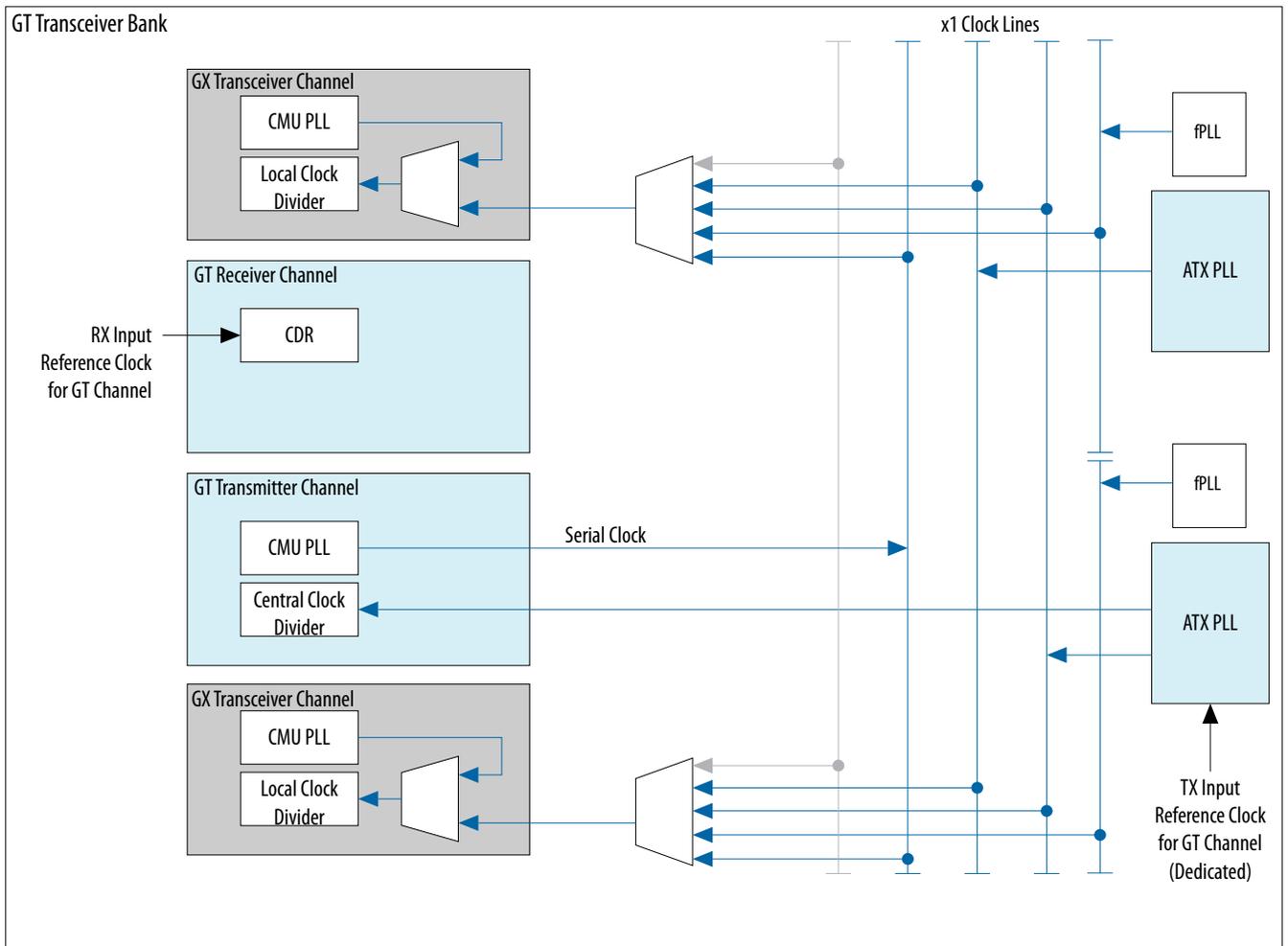
Transmitter Clock Network in Stratix V GT Transceiver Channels

The x1 clock lines in GT transceiver banks route the serial clock from the ATX PLL to the central clock divider of the GT transmitter channel. The x1 clock lines can also route the serial clock from the GT transmitter channel CMU PLL and the ATX PLLs to the local clock dividers of the GX transceiver channels.

Figure 2-13: x1 Clock Lines Used by Stratix V GT Transmitter Channels

The following figure shows the x1 clock lines used by a GT transmitter channel and two GX transceiver channels in a GT transceiver bank. There is one GT transmitter channel per GT transceiver bank. The transmitter channel must receive the clock from the bottom ATX PLL of a GT transceiver bank.

Note: The channel PLL in the GT receiver channel is always used as a CDR.

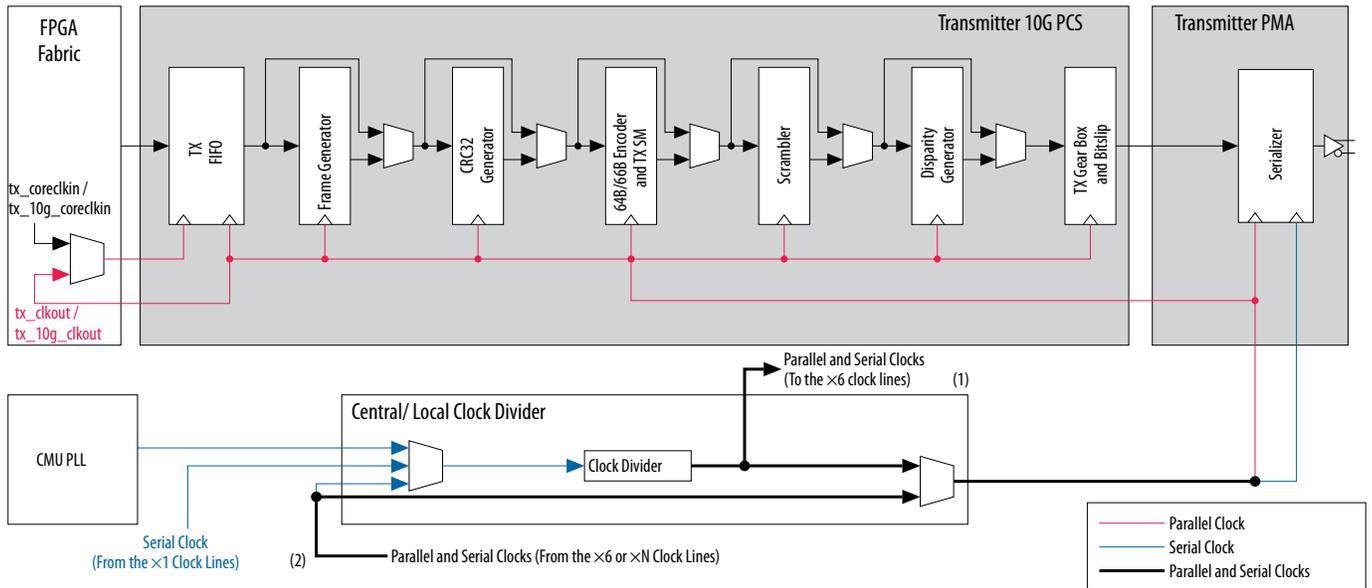


Transmitter Clocking

Transmitter clocking refers to the clocking architecture internal to the transmitter channel of a transceiver.

Transmitter 10G PCS Clocking

Figure 2-14: Transmitter 10G PCS and PMA Clocking

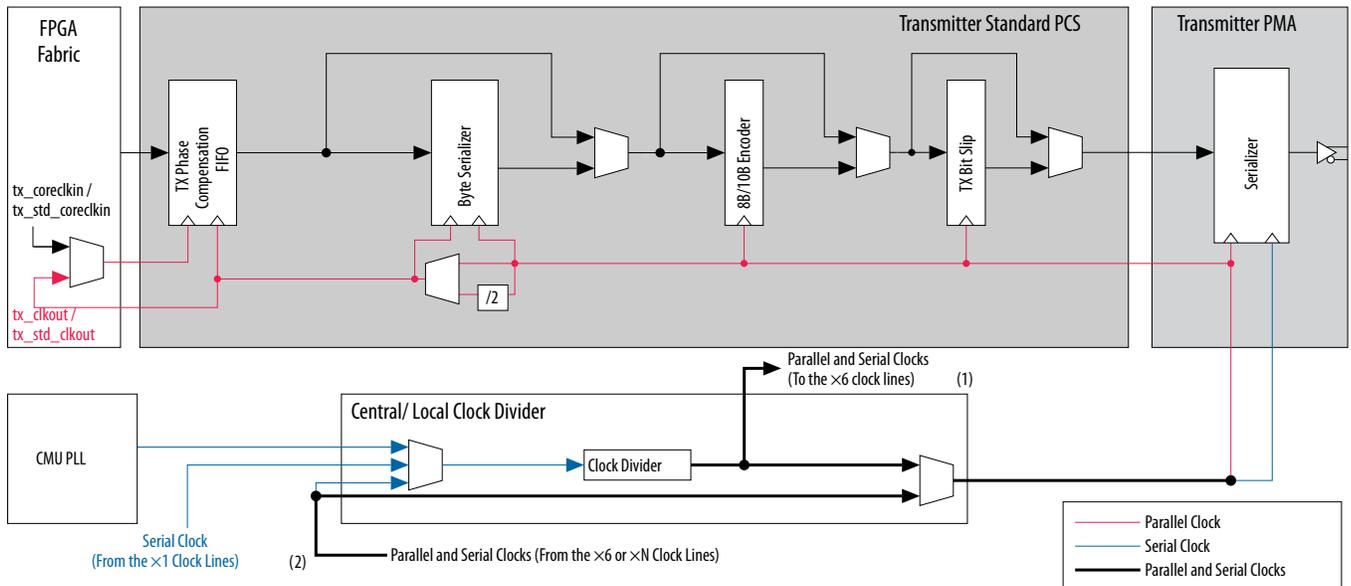


- Notes:**
- (1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.
 - (2) x1 clock lines can be driven by a CMU PLL, ATX PLL, or a fractional PLL.

Transmitter Standard PCS Clcking

Figure 2-15: Transmitter Standard PCS and PMA Clcking

The clock divider block provides the serial clock to the serializer of the transmitter PMA and the parallel clock to the transmitter PCS.



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.

In the 10G PCS channel, the parallel clock is used by all the blocks up to the read side of the transmitter (TX) FIFO.

In the standard PCS channel, the parallel clock is used by all the blocks up to the read side of the TX phase compensation FIFO in all configurations that do not use the byte serializer block. For configurations that use the byte serializer block, the clock is divided by a factor of two for the byte serializer and the read side of the TX phase compensation FIFO. The clock used to clock the read side of the TX phase compensation FIFO is also forwarded to the FPGA fabric to provide an interface between the FPGA fabric and the transceiver.

Note: For more information about clocking schemes used in different configurations, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

Related Information

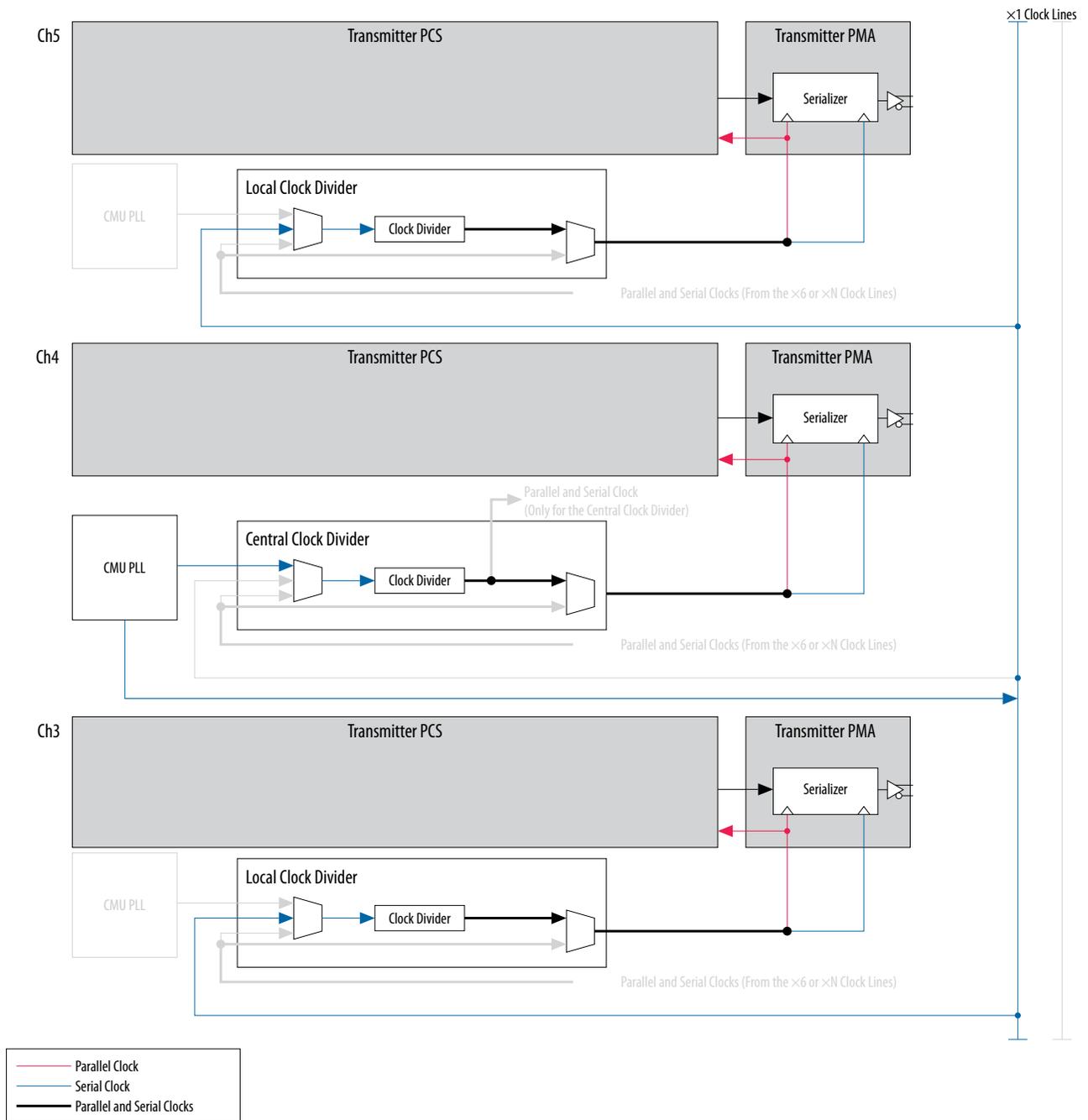
[Transceiver Configurations in Stratix V Devices](#)

Non-Bonded Channel Configurations Using the x1 Clock Network

In non-bonded channel configurations using the x1 clock networks, the parallel clock is generated by the clock divider of individual channels.

Figure 2-16: Three Transmitter-Only Channels Configured in x1 Non-Bonded Configuration

The figure shows three transmitter-only channels in non-bonded configuration driven by the channel PLL of channel 4 configured as a CMU PLL driving the x1 clock line. The clock divider block of each channel generates its own parallel clock by dividing the serial clock from the x1 clock line.

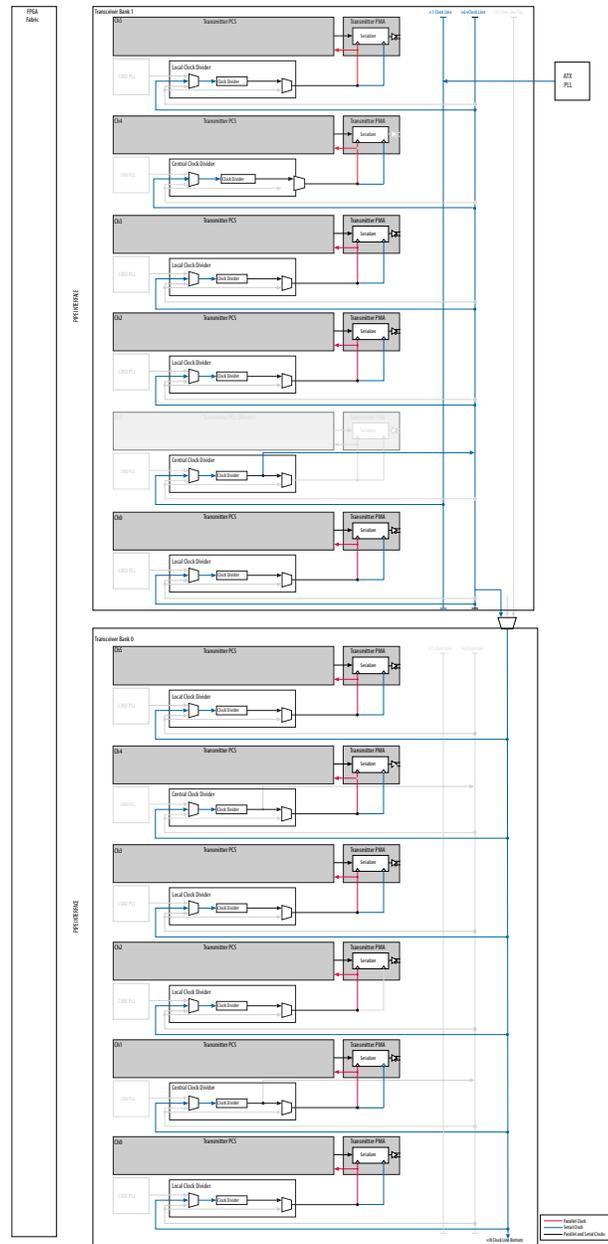


Non-Bonded Channel Configurations Using the xN Clock Network

In non-bonded channel configurations using the xN clock network, the parallel clock is generated by the clock divider of the individual channels.

Figure 2-17: Transmitter Channels in xN Non-Bonded Configuration

The figure shows 11 transmitter channels in non-bonded configuration. These channels are driven by the ATX PLL of the transceiver bank 1 which drives the x6 clock line through the central clock divider of the transceiver channel 1 in bank 1. The local clock divider block of each channel generates its own parallel clock by dividing the serial clock from the xN clock line. The channel where the central clock divider resides cannot generate the parallel clock and therefore it cannot be used as a data channel.



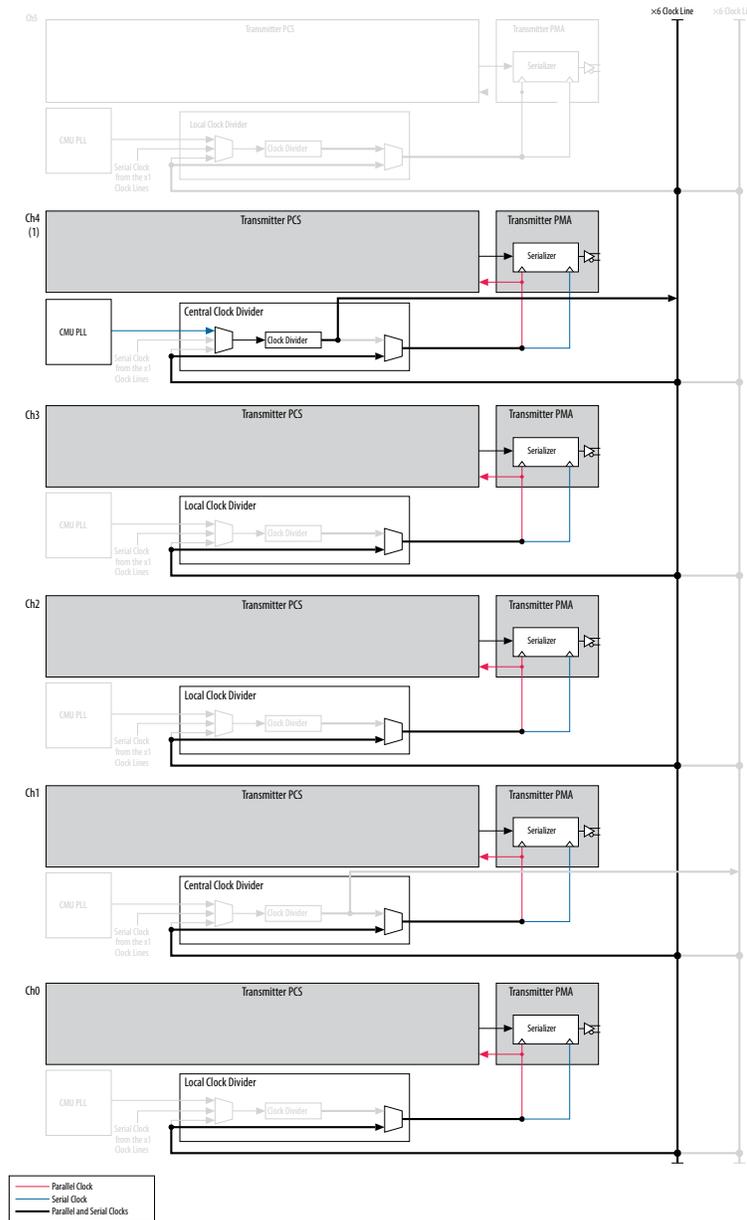
Bonded Channel Configurations

In bonded configurations, both the parallel clock and serial clock are sourced from either the x6 or xN clock line.

The central clock dividers source the serial clock from a transmitter PLL from the same transceiver bank using the x1 clock line. The central clock divider generates the parallel clock and drives both the serial clock and parallel clock on the x6 clock line, which can drive the xN clock line.

Figure 2-18: Five Transmitter-Only Channels Configured in Bonded Configuration

The figure shows five transmitter-only channels configured in a bonded configuration and driven by the channel PLL of channel 4 configured as a CMU PLL. The central clock divider of channel 4 generates a parallel clock and drives both the serial clock and parallel clock on the x6 clock line. All bonded channels source both serial and parallel clocks from the x6 clock line.



Note: (1) Channel 4 cannot be used as a receiver channel because the CMU PLL is used to generate clocks for other channels.

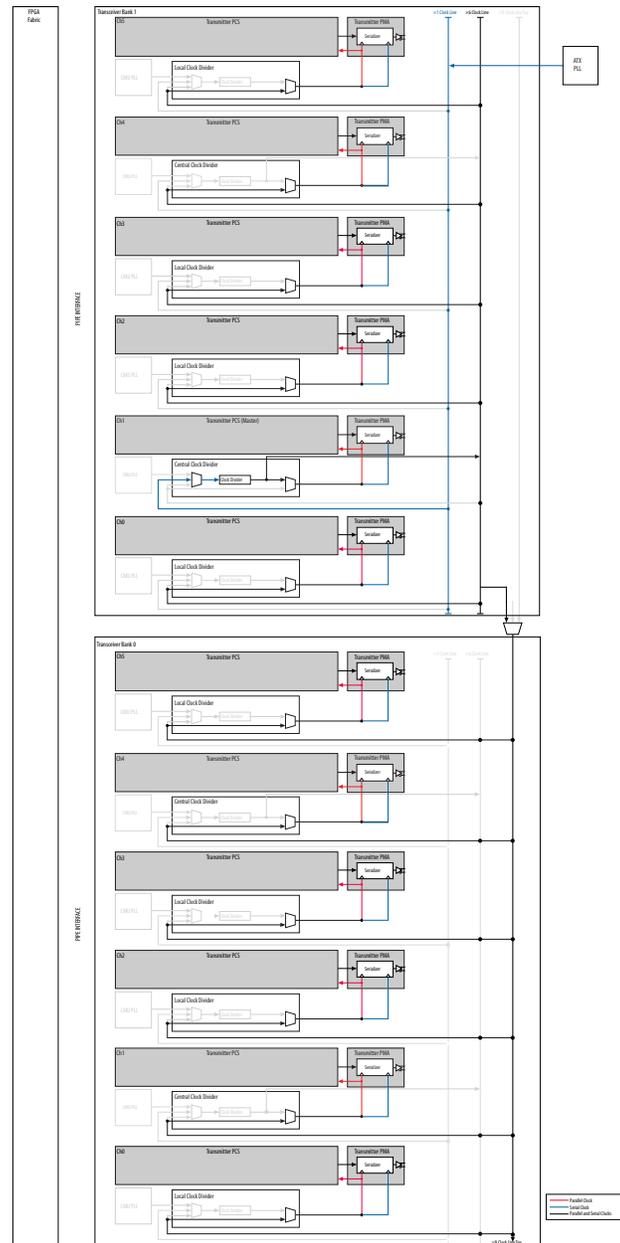
Related Information**Transceiver Configurations in Stratix V Devices**

For an example of using the xN clock lines, refer to the PCIe x8 configuration in the PCI Express (PCIe)—Gen1, Gen2, and Gen3 section of the *Transceiver Configurations in Stratix V Devices* chapter.

Bonded Channel Configurations Using the xN Clock Network

Figure 2-19: Transmitter Channels in xN Bonded Configuration

The following figure shows 12 transmitter channels configured in a bonded configuration and driven by the ATX PLL of the transceiver bank 1. The ATX PLL drives the central clock divider of channel 1 in transceiver bank 1. The central clock divider of channel 1 generates a parallel clock and drives both the serial and parallel clocks on the x6 clock line. All bonded channels in transceiver bank 1 source both the serial and parallel clocks from the x6 clock line. The x6 clock line in transceiver bank 1 also drives the xN clock lines in transceiver bank 0. All bonded channels in transceiver bank 0 source both the serial and parallel clocks from the xN clock line.



Bonded Channel Configurations Using the PLL Feedback Compensation Path

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.

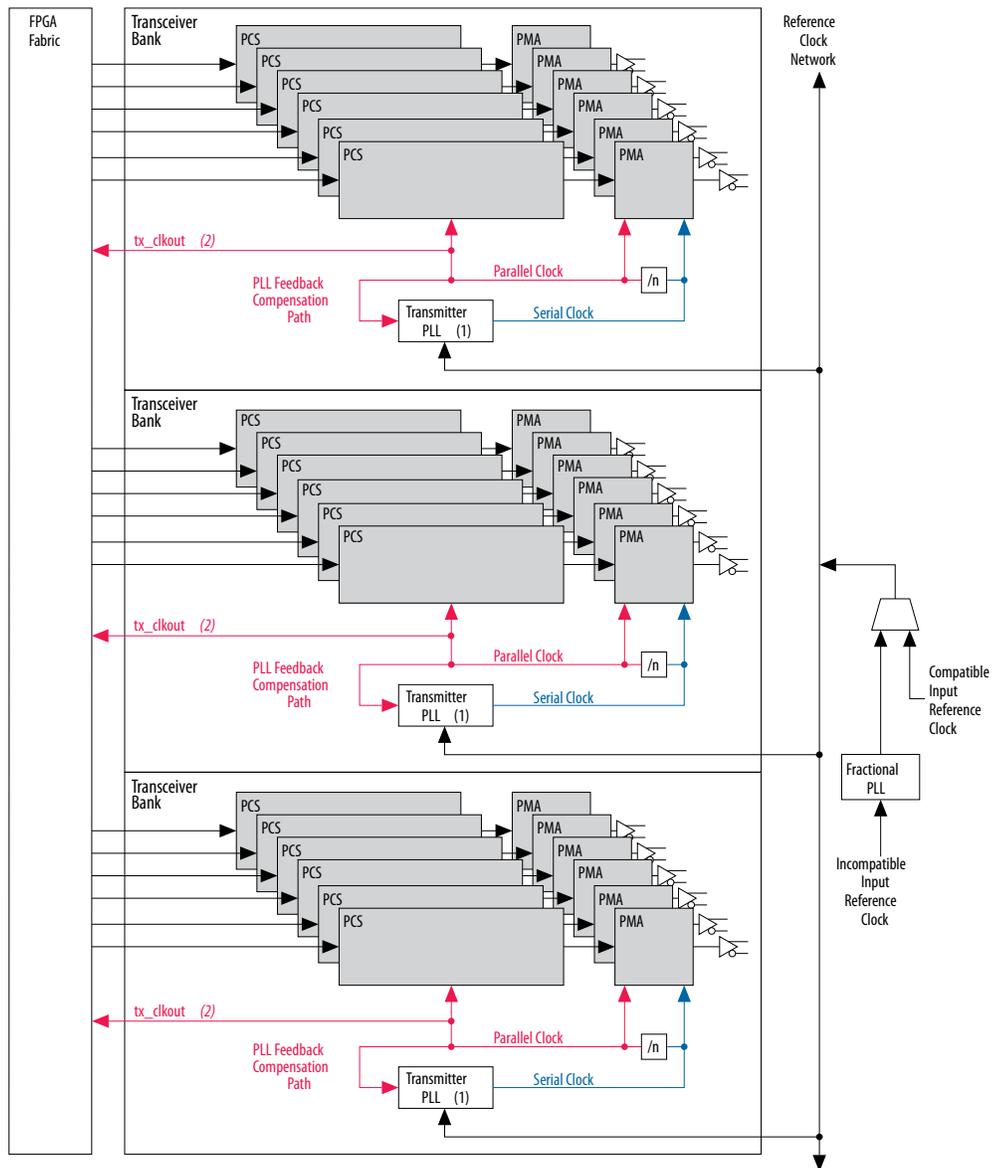
To bond channels using the PLL feedback compensation path, the input reference clock frequency used by the transmitter PLL must be the same as the parallel clock that clocks the PCS of the same channel. If the input reference clock frequency is not equal to the parallel clock frequency, use a fractional PLL to synthesize an input reference clock with the same frequency as the parallel clock.

Notes:

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

Figure 2-20: Three Transceiver Bank Channels Bonded Using the PLL Feedback Compensation Path

The figure shows 18 transmitter channels within three transceiver banks that are bonded using the PLL feedback compensation path. A fractional PLL is used to synthesize a clock that has the same frequency as the parallel clocks, which are looped back to the transmitter PLL.



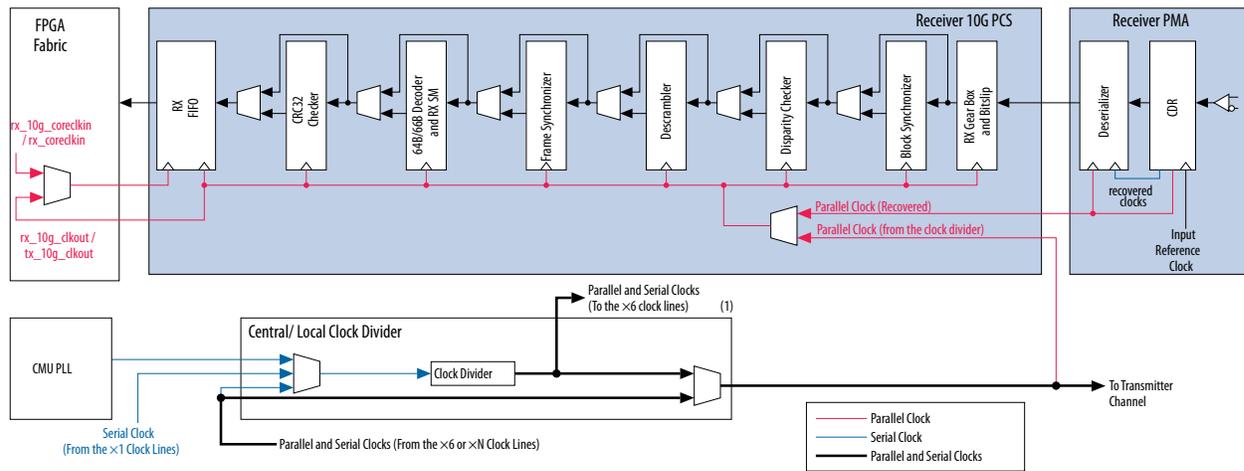
Notes: (1) The transmitter PLL can be an ATX PLL or a CMU PLL. You can have up to six channels per bank with an ATX PLL and five channels per bank with a CMU PLL.
(2) tx_clkout (2) from any of the banks can be used with the FPGA fabric-transceiver interface for all the bonded channels.

Transmitter GT Channel Clocking

In a Stratix V GT transmitter channel the central clock divider block provides the serial and the parallel clocks to the serializer.

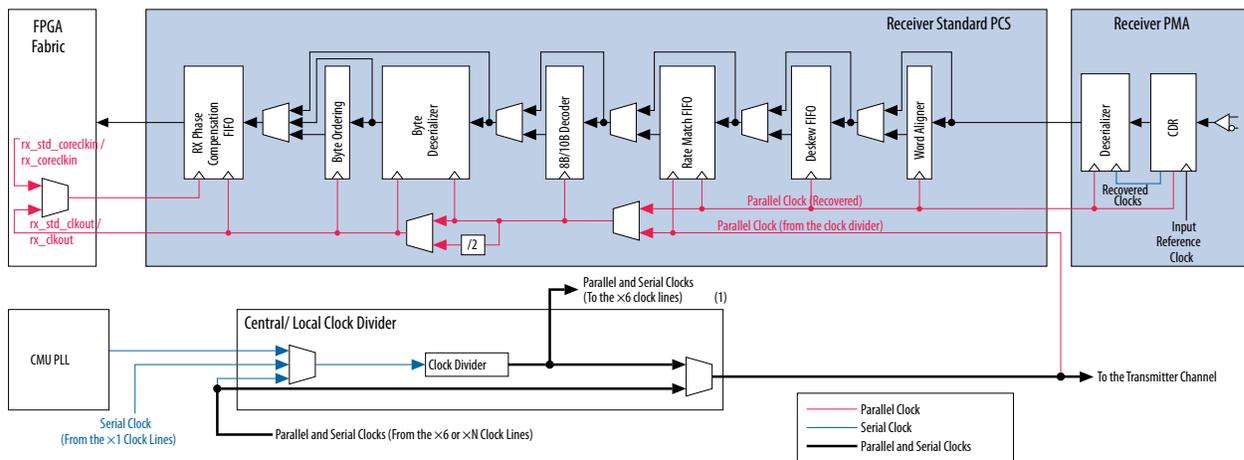
GX Channel Receiver Clocking

Figure 2-22: Receiver 10G PCS and PMA Clocking



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

Figure 2-23: Receiver Standard PCS and PMA Clocking



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

The CDR in the PMA of each channel recovers the serial clock from the incoming data. The CDR also divides the serial clock (recovered) to generate the parallel clock (recovered). Both clocks are used by the deserializer. 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 used by the transmitter PCS for that channel

Table 2-4: Clock Sources for All Receiver PCS Blocks

PCS	Block	Clock Source
Standard	Word aligner	Parallel clock (recovered)
	Rate match FIFO	Write side: parallel clock (recovered) Read side: parallel clock from the clock divider
	8B/10B decoder	If rate matcher is not used: parallel clock (recovered) If rate matcher is used: parallel clock from the clock divider
	Byte deserializer	Write side: <ul style="list-style-type: none"> If rate matcher is not used: parallel clock (recovered) If rate matcher is used: parallel clock from the clock divider Read side: 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 Side: Parallel clock (divided). This clock is also forwarded to the FPGA fabric Read Side: Clock sourced from the FPGA fabric
10G	All PCS blocks	Regular mode: parallel clock (recovered) Loopback mode: parallel clock from the clock divider ⁽⁷⁾

Related Information

[Transceiver Loopback Support in Stratix V Devices](#)

Non-Bonded Channel Configurations

In non-bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider.

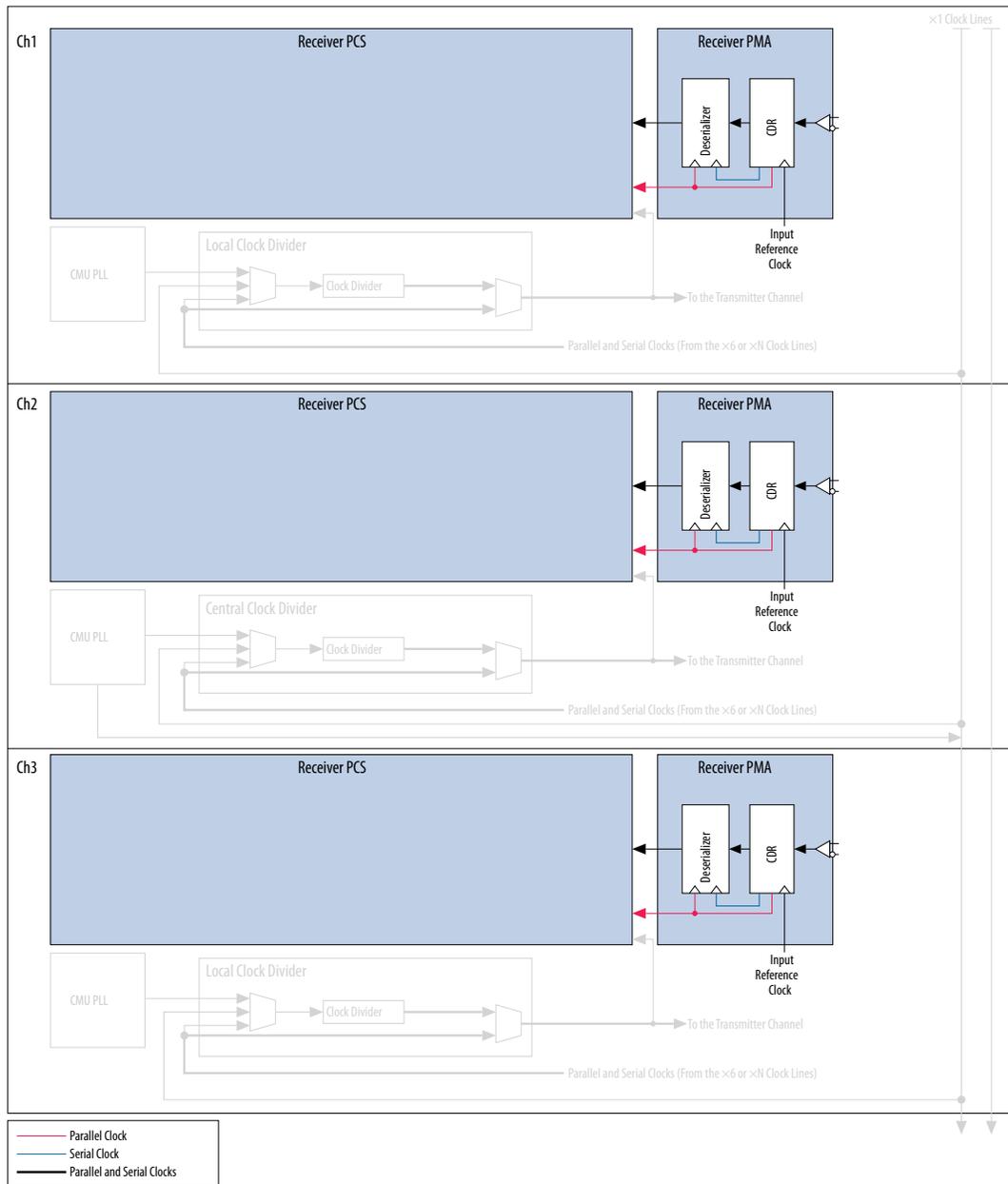
Depending on the configuration, the receiver PCS may require the parallel clock from the clock divider that is used for the transmitter PCS.

Note: In non-bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

⁽⁷⁾ For more information about loopback mode, refer to the *Transceiver Loopback Support in Stratix V Devices* chapter.

Figure 2-24: Three Channels Configured in Non-Bonded Configuration

The figure shows three channels configured in non-bonded configuration. They use the receiver standard PCS that does not use a rate match FIFO. The CDR of each channel recovers the serial clock (recovered) from the incoming data and generates the parallel clock (recovered) by dividing the serial clock (recovered). Depending on the configuration, the receiver PCS may also use the parallel clock from the clock divider that is generated by the local clock divider for the transmitter.



Note: For more information about the clocking scheme used in different configurations, refer to the Transceiver Configurations in Stratix V Devices chapter.

Related Information

[Transceiver Configurations in Stratix V Devices](#)

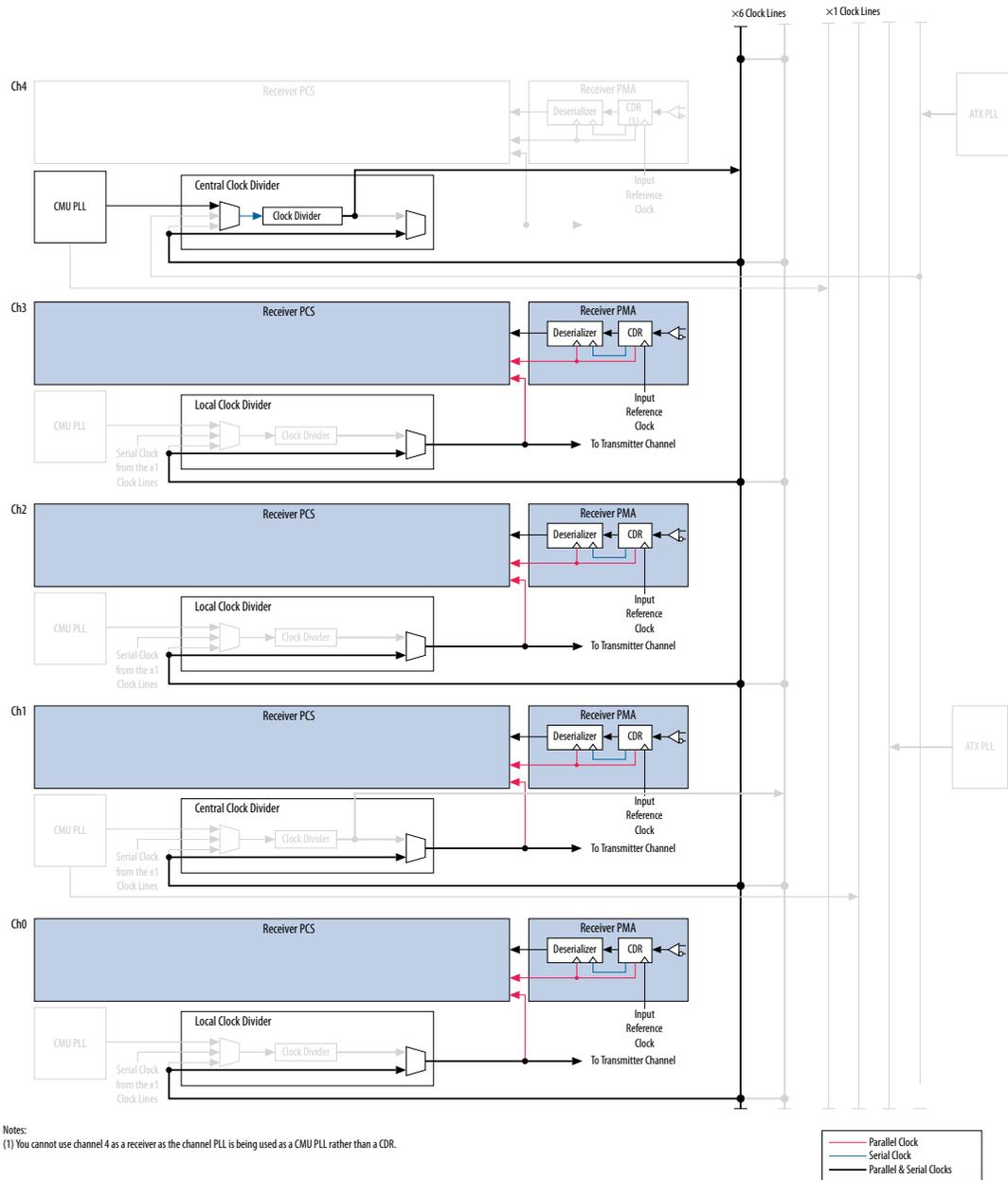
Bonded Channel Configurations

In bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider.

In bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2-25: Four Receiver Channels Configured in Bonded Duplex Configuration

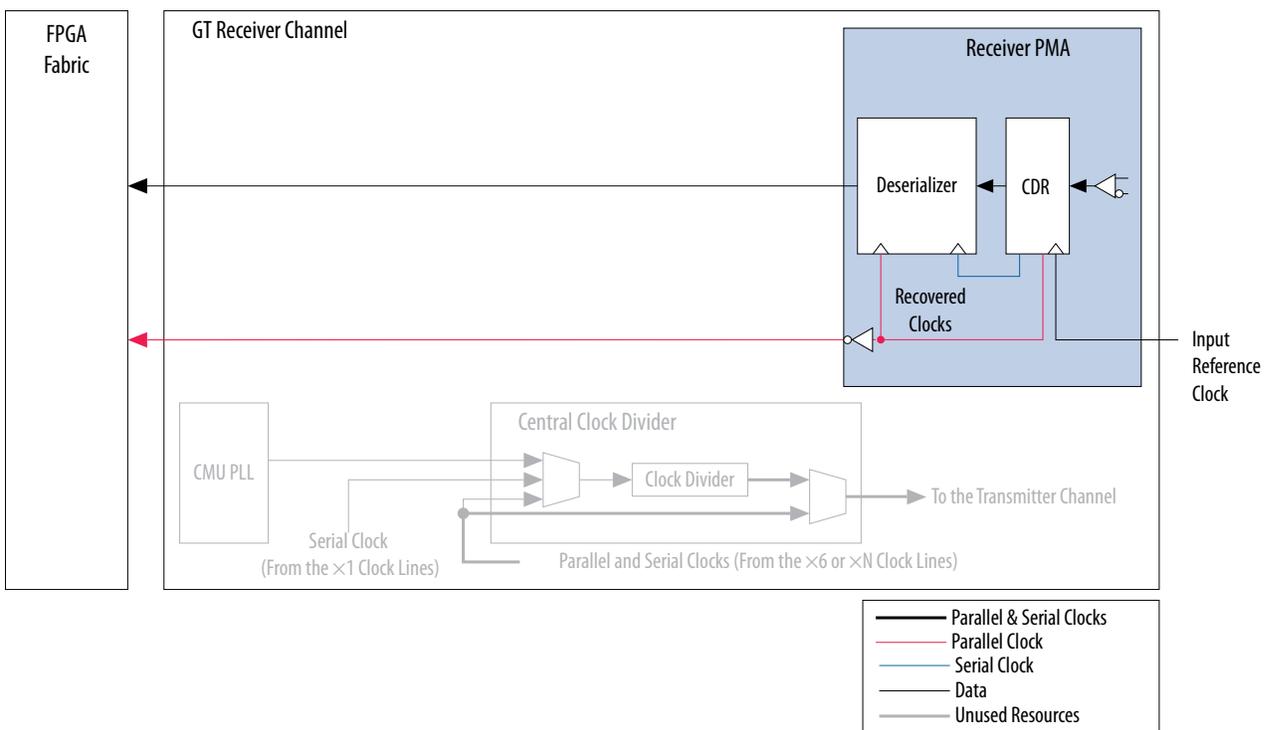
The figure shows four channels in a transceiver bank configured in bonded configuration, using the receiver standard PCS. The receiver PCS uses both the parallel clock (recovered) and parallel clock from the clock divider. The parallel clock from the clock divider is generated by the central clock divider for the transmitter PCS. It also drives some blocks in the receiver PCS, depending on the configuration you use.



Related Information**Transceiver Configurations in Stratix V Devices****GT Channel Receiver Clocking**

The CDR in the PMA of the GT receiver channel recovers the serial clock from the incoming data and is driven by an input reference clock or clock from the reference clock network in the same GT transceiver bank.

The CDR also divides the serial clock (recovered) to generate the parallel clock (recovered). Both clocks are used by the deserializer. The parallel clock (recovered) is forwarded to the FPGA fabric to interface the FPGA fabric with the transceiver. All PCS functions, such as word alignment, rate matching, decoding, and byte ordering, must be implemented in the FPGA core because the PCS is unavailable in the GT receiver channel.

Figure 2-27: GT Channel Receiver Clocking**FPGA Fabric-Transceiver Interface Clocking**

The FPGA fabric-transceiver interface clocks consist of clock signals from the FPGA fabric to the transceiver blocks and clock signals from the transceiver blocks to the FPGA fabric. These clock resources use the clock networks in the FPGA core, including the global (GCLK), regional (RCLK), and periphery (PCLK) clock networks.

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

- Input reference clocks—Refer to **Input Reference Clock Sources** on page 2-2. The input reference clock can be an FPGA fabric-transceiver interface clock when it is also forwarded to the FPGA fabric to clock the logic in the FPGA fabric.
- 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—Used to form a part of the FPGA fabric-transceiver interface clocks as follows:

`phy_mgmt_clk`—Avalon[®]-MM interface clock used for controlling the transceivers, dynamic reconfiguration, and calibration

`fixed_clk`—125 MHz fixed-rate clock used in the PCIe (PIPE) receiver detect circuitry

Table 2-5: FPGA Fabric-Transceiver Interface Clocks

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization
pll_refclk, rx_cdr_refclk	A transceiver PMA TX PLL and CDR reference clock, sourced by dedicated differential pins of the device.	Input	GCLK, RCLK, PCLK
tx_clkout, tx_pma_clkout	Clock forwarded by the transceiver for clocking the transceiver datapath interface. The value of tx_clkout / tx_pma_clkout is derived by dividing the data rate by the serialization factor. For example, a 3 Gbps link with a serialization factor of 20 will result in a tx_clkout of 150 MHz.	Transceiver-to-FPGA fabric	
rx_clkout, rx_pma_clkout	Clock forwarded by the receiver for clocking the receiver datapath interface. The value of rx_clkout / rx_pma_clkout is derived by dividing the data rate by the deserialization factor. For example, a 10 Gbps link with a deserialization factor of 40 will result in a rx_clkout of 250 MHz.		
tx_10g_coreclk/tx_std_coreclk	User-selected clock for clocking the transmitter datapath interface	FPGA fabric-to-transceiver	
rx_10g_coreclk / rx_std_coreclk	User-selected clock for clocking the receiver datapath interface		
fixed_clk	PCIe receiver detect clock		
phy_mgmt_clk ⁽⁸⁾	Avalon-MM interface management clock		

Note: You can forward the `pll_ref_clk`, `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 peripheral clocks, depending on your device and channel placement, and may require using a RGCLK or GCLK.

Table 2-6: Configuration Specific Port Names for `tx_clkout` and `rx_clkout`

Configuration	Port Name for <code>tx_clkout</code>	Port Name for <code>rx_clkout</code>
Custom	<code>tx_clkout</code>	<code>rx_clkout</code>
Native - 10G PCS	<code>tx_10g_clkout</code>	<code>rx_10g_clkout</code>
Native - Standard PCS	<code>tx_std_clkout</code>	<code>rx_std_clkout</code>
Native - PMA Direct	<code>tx_pma_clkout</code>	<code>rx_pma_clkout</code>
Interlaken	<code>tx_clkout</code>	<code>rx_clkout</code>
Low Latency	<code>tx_clkout</code>	<code>rx_clkout</code>
PCIe	<code>pipe_pclk</code>	<code>pipe_pclk</code>
XAUI	<code>xgmii_tx_clk</code>	<code>xgmii_rx_clk</code>

Note: For more information about the GCLK, RCLK, and PCLK resources available in each device, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter.

Related Information

[Clock Networks and PLLs in Stratix V Devices](#)

Transmitter Datapath Interface Clocking

The transmitter datapath interface consists of the following:

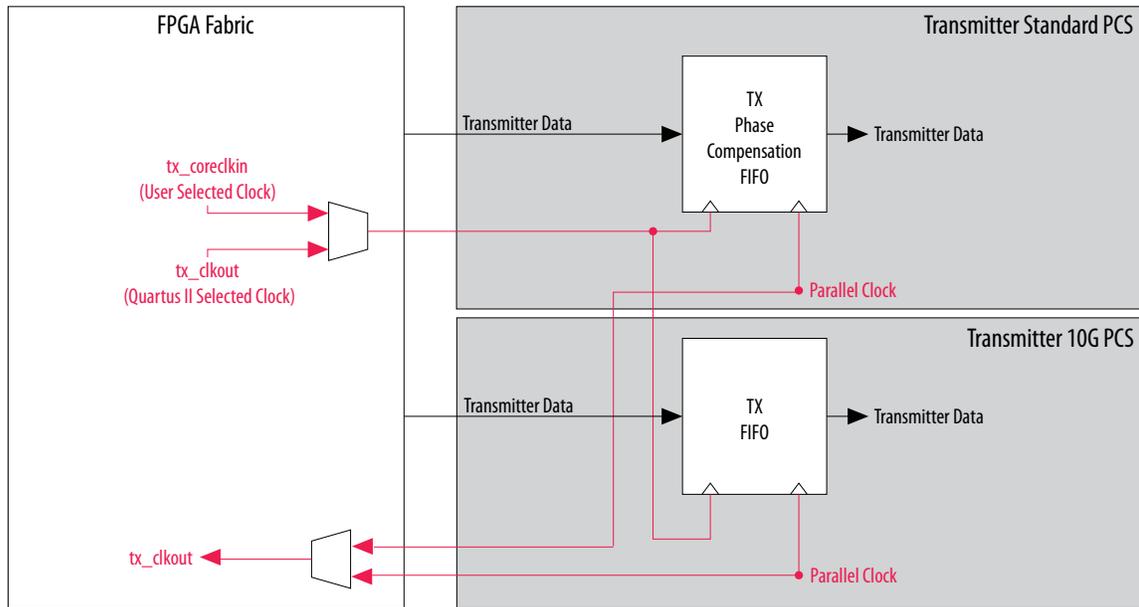
- Write side of the TX phase compensation FIFO—for configurations that use the standard PCS channel
- Write side of the TX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the transmitter datapath interface clock. The transmitter PCS forwards the following clocks to the FPGA fabric:

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

⁽⁸⁾ The `phy_mgmt_clk` is a free-running clock that is not derived from the transceiver blocks, except if `phy_mgmt_clk` is derived from the dedicated `refclk` pin.

Figure 2-28: Transmitter Datapath Interface Clocking



All configurations using the standard PCS channel must have a 0 parts per million (ppm) difference between the transmitter datapath interface clock and the read side clock of the TX phase compensation FIFO.

Note: For more information about interface clocking for each configuration, refer to the Transceiver Configurations in Stratix V Devices chapter.

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

- Quartus II-selected transmitter datapath interface clock
- User-selected transmitter datapath interface clock

Note: User selection allows you to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Related Information

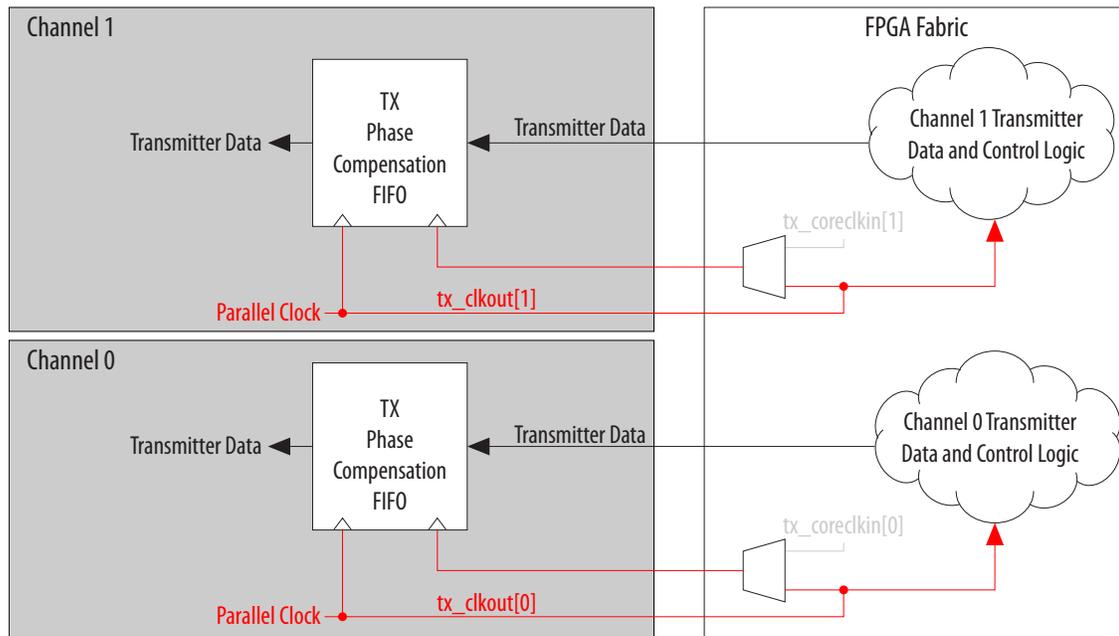
[Transceiver Configurations in Stratix V Devices](#)

Quartus II-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 2-29: Transmitter Datapath Interface Clocking for Non-Bonded Channels

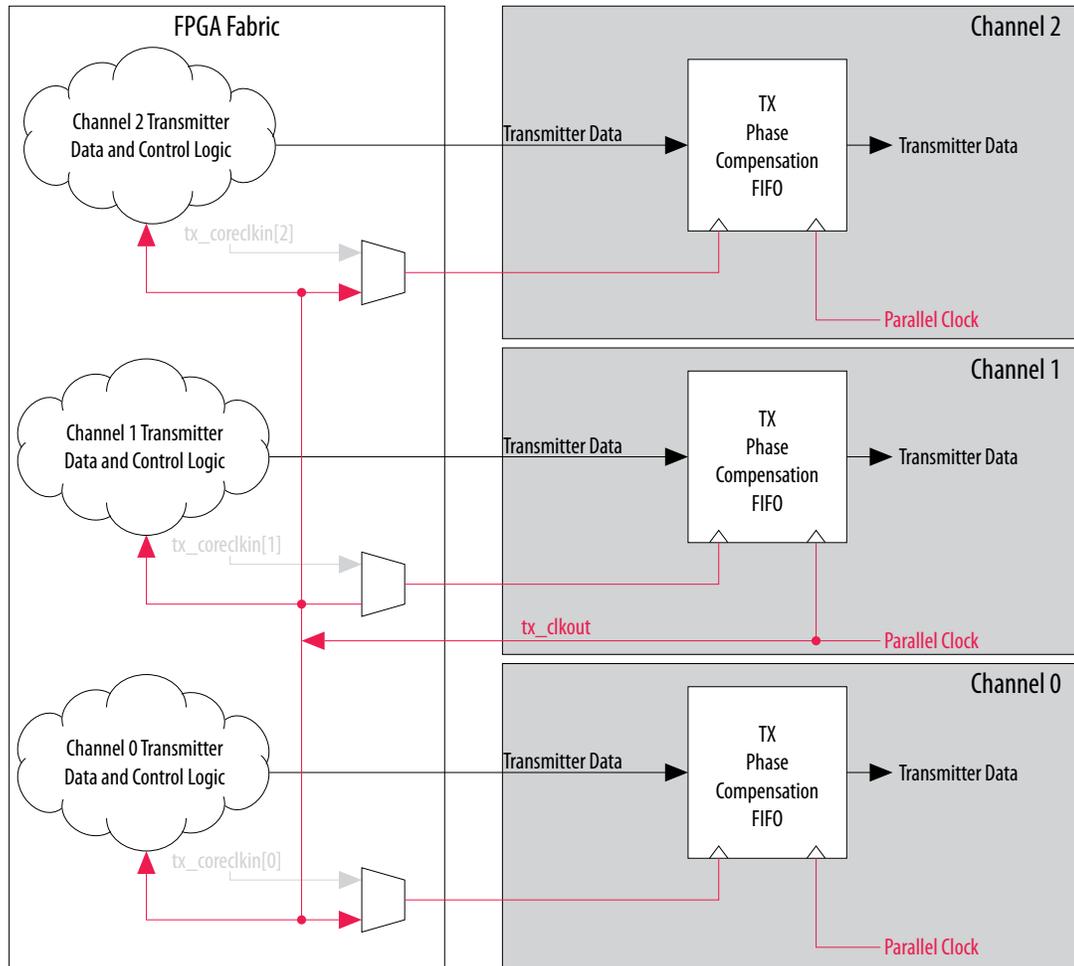
The figure shows the transmitter datapath interface of two non-bonded channels clocked by their respective transmitter PCS clocks that are forwarded to the FPGA fabric.



Note: The FPGA fabric-transceiver interface clocking for GT transmitter channels is similar to the non-bonded GX transmitter channel FPGA fabric-transceiver interface clocking.

Figure 2-30: Transmitter Datapath Interface Clocking for Three Bonded Channels

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



Selecting a Transmitter Datapath Interface Clock

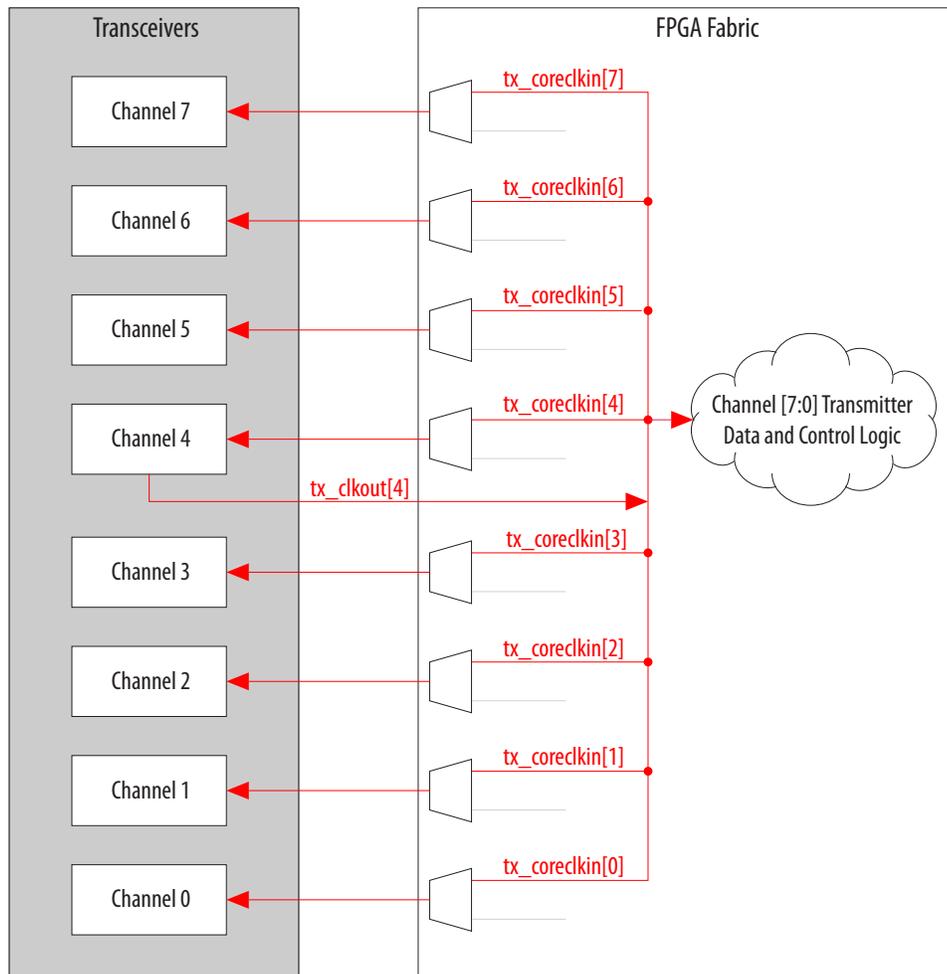
Multiple non-bonded transmitter channels use a large portion of GCLK, RCLK, and PCLK resources. Selecting a common clock driver for the transmitter datapath interface of all identical transmitter channels saves clock resources.

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 2-31: 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_coreclkkin` port for all the identical transmitter channels (`tx_coreclkkin[7:0]`).
2. Connect `tx_clkout[4]` to the `tx_coreclkkin[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
- When there is 0 PPM between `refclk` and `tx_clkout`

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

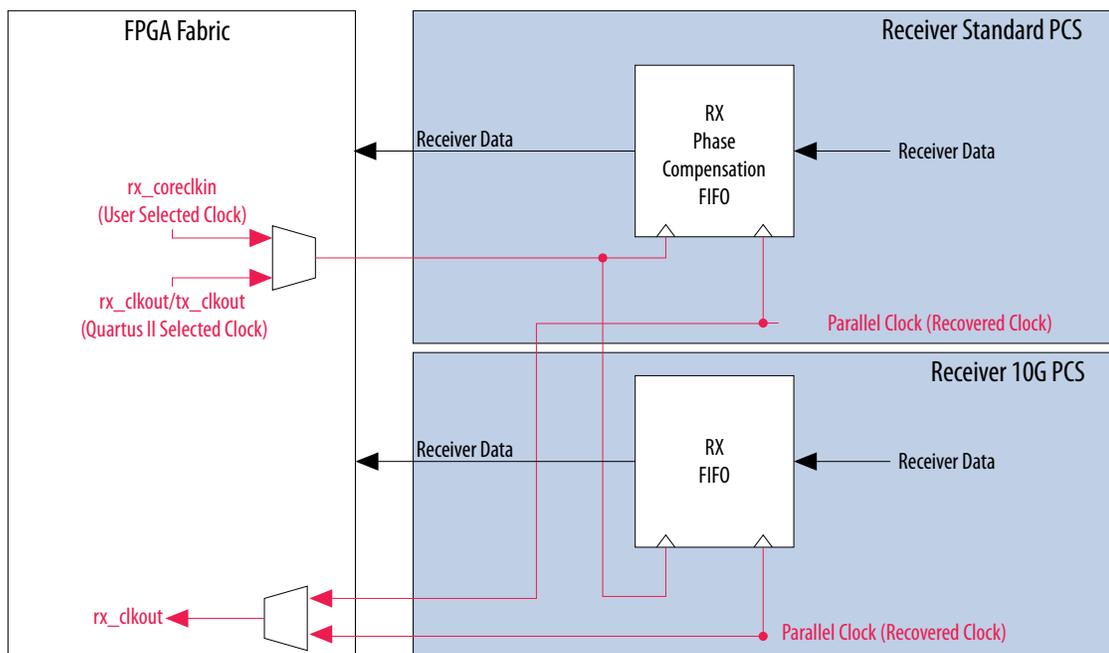
The receiver datapath interface consists of the following:

- Read side of the RX phase compensation FIFO—for configurations that use the standard PCS channel
- Read side of the RX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the receiver datapath interface clock. 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 2-32: Receiver Datapath Interface Clocking



All configurations that use the standard 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.

Note: For more information about interface clocking for each configuration, refer to the clocking sections for each configuration in the Transceiver Configurations in Stratix V Devices chapter.

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

- Quartus II-selected receiver datapath interface clock
- User-selected receiver datapath interface clock

Note: User-selection is provided to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Related Information

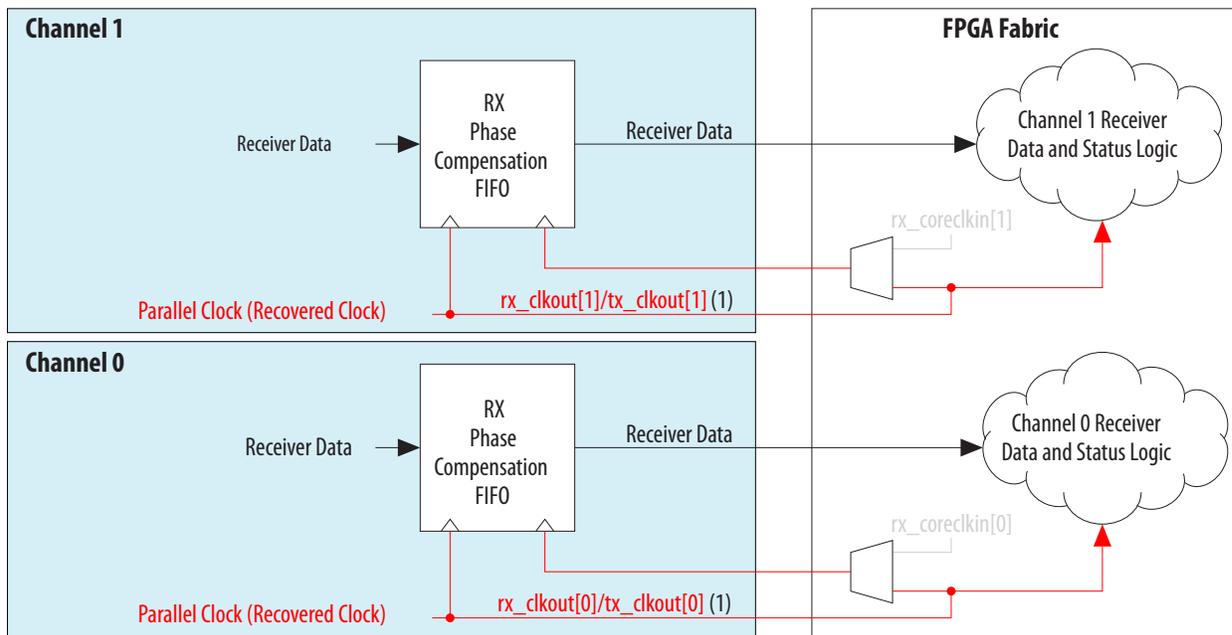
Transceiver Configurations in Stratix V Devices

Quartus II Software-Selected Receiver Datapath Interface Clock

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

Figure 2-33: Receiver Datapath Interface Clocking for Non-Bonded Channels

The figure shows the receiver datapath interface of two non-bonded channels that are clocked by their respective receiver PCS clocks and forwarded to the FPGA fabric.

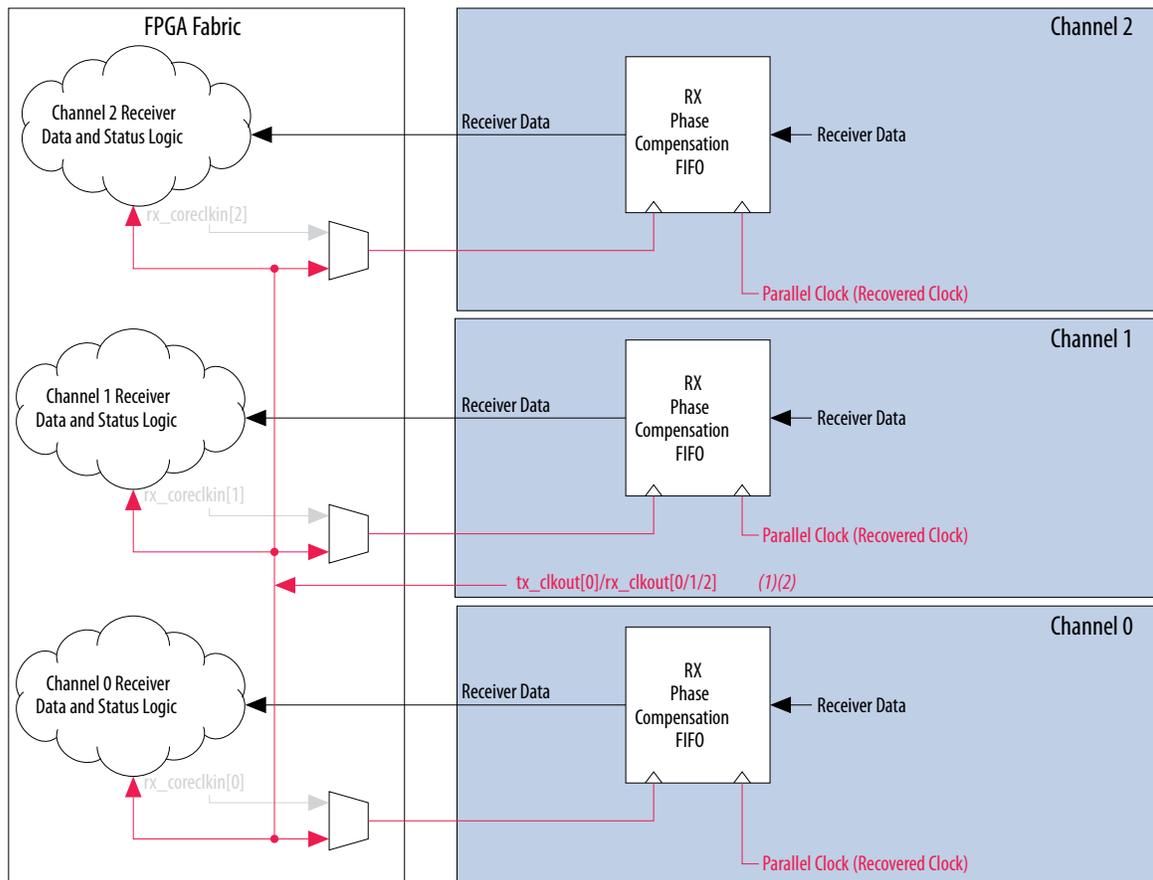


Note : (1) If you use a rate matcher, the tx_clkout clock is used.

Note: The FPGA fabric-transceiver interface clocking for GT receiver channels is similar to the non-bonded GX receiver channel FPGA fabric-transceiver interface clocking.

Figure 2-34: Receiver Datapath Interface Clocking for Three Bonded Channels

The following figure shows the receiver 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.

**Notes:**

(1) `tx_clkout[0]` can only be used if a rate matcher is used.

(2) `rx_clkout[0]`, `rx_clkout[1]` or `rx_clkout[2]` can be used to clock the read side of the rx phase compensation FIFO of all channels only if there is 0 ppm difference between the `rx_clkout[0]`, `rx_clkout[1]`, and `rx_clkout[2]`.

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

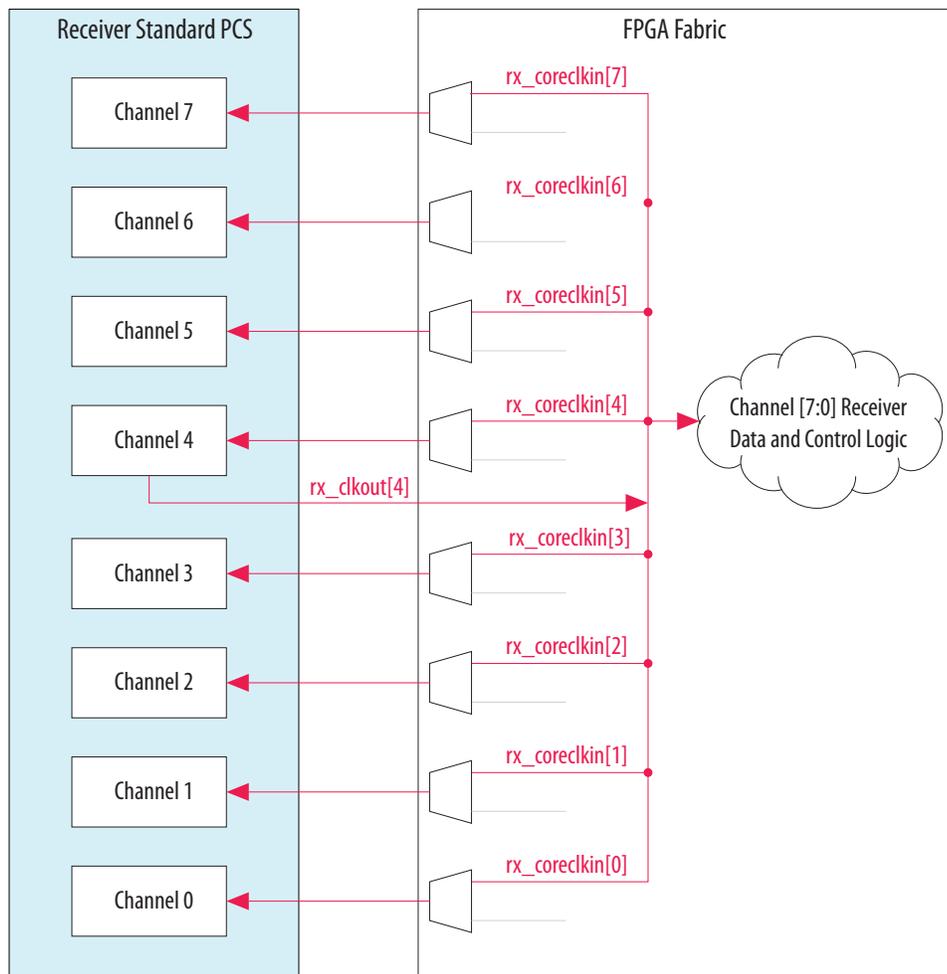
be PPM aligned with regards to their remote transmitters. These channels may have different analog settings, such as receiver common mode voltage (V_{ICM}), equalization, or DC gain setting.

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_coreclkkin` 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 2-35: 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_coreclkkin` port for all the identical receiver channels (`rx_coreclkkin[7:0]`).
- Connect `rx_clkout[4]` to the `rx_coreclkkin[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_coreclk` 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

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 2-7: 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 ⁽⁹⁾	Description
To	<code>tx_dataout/rx_datain</code> pins of all channels whose <code>tx/rx_coreclk</code> 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](#)

⁽⁹⁾ 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 2-8: Document Revision History

Date	Version	Changes
January 2016	2016.01.11	Added statement about powering up power pins to the "Dedicated refclk Pins" section.
September 2014	2014.09.30	<ul style="list-style-type: none"> Modified <i>Figure: Four Receiver Channels Configured in Bonded Duplex Configuration</i> in "Bonded Channel Configurations" section to indicate that CMU PLL can support only four channels in a bonded configuration. Modified <i>Figure: Input Reference Clock Sources to Transmit PLLs and CDR</i> and <i>Figure: Fractional PLL Input Clock Sources</i>. The reference clock network can be used as an input reference clock source and the dedicated reference clock pins feed the reference clock network. Modified <i>Figure: Transmitter Datapath Interface Clocking</i>. The tx_clkout multiplexer is implemented in the FPGA fabric and is not present in the PCS. Modified <i>Figure: Receiver Datapath Interface Clocking</i>. The rx_clkout multiplexer is implemented in the FPGA fabric and is not present in the PCS. 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 the definition of identical receiver channels in <i>Selecting a Receiver Datapath Interface Clock</i> section.
October 2013	2013.10.11	<ul style="list-style-type: none"> Updated "Dedicated refclk Pins" section.
May 2013	2013.05.06	<ul style="list-style-type: none"> Updated for Quartus II software version 13.0 feature support. Added table "Electrical Specifications for the Input Reference Clock". Added figure "Termination Scheme for a Reference Clock Signal When Configured as HCSL". Updated table "Data Rates and Spans Supported Using Stratix V Clock Sources and Clock Networks". Added information and figures for bonded and non-bonded channel configurations using the xN clock network. Added link to the known document issues in the Knowledge Base.

Date	Version	Changes
December 2012	2012.12.17	<ul style="list-style-type: none"> • Reorganized content and updated template. • Updated for the Quartus II software version 12.1. • Updated Figures 2-2, 2-3, 2-4, 2-5, 2-7, 2-8, 2-11, 2-21, and 2-26. • Added the “RX Pins Using the Reference Clock Network” section. • Updated table "Data Rates and Spans Supported Using Stratix V Clock Sources and Clock Networks." • Updated table "FPGA Fabric-Transceiver Interface Clocks" to address FB #60881. • Updated additional info in table "FPGA Fabric-Transceiver Interface Clocks" to address FB #65061. • Updated table "Configuration Specific Port Names for tx_clkout and rx_clkout."
June 2012	1.6	<ul style="list-style-type: none"> • Updated for the Quartus II software version 12.0 and reordered paragraphs. • Added Clock Divider section previously located in Architecture chapter. • Added information about GXB 0 PPM core clock assignment. • Updated Figures 2-4, 2-7, 2-17, 2-21, 2-23, and 2-30.
February 2012	1.5	<ul style="list-style-type: none"> • Updated document and figures for clarity. • Edited Figures 2-2, 2-3, and 2-4.
December 2011	1.4	<ul style="list-style-type: none"> • Updated document and figures for clarity. • Changed path for serial and parallel clocks in channel 4 in Figure 2–14, Figure 2–20, and Figure 2–21.
November 2011	1.3	<ul style="list-style-type: none"> • Added information about GT transceivers. • Added information about bonding channels across transceiver banks by using the PLL feedback compensation path. • x8 bonding using the xN clock lines is now available for PCIe Gen3. • Added information about the transceiver clocks used in the FPGA fabric. • Added information about fractional mode when using fractional PLLs. • Added information about using the FPGA fabric clocks as a reference clock to GX transceiver channels. • Added information about forwarding transceiver clocks to a fractional PLL so that the fractional PLL can synthesize a clock for the FPGA logic.

Date	Version	Changes
May 2011	1.2	<ul style="list-style-type: none">Added information about fractional PLLs as they provide an input reference clock in “Input Reference Clocking.”Chapter moved to Volume 3.
December 2010	1.1	<ul style="list-style-type: none">Updated clock names.Updated figures for more accurate depiction of transceiver clocking.Added information about ATX PLLs.
July 2010	1.0	Initial release.

2016.01.11

SV52004



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.

The Stratix V transceiver reset sequence is mandatory to initialize the physical coding sublayer (PCS) and physical medium attachment (PMA) blocks. Multiple reset options are available to reset the analog and digital portions of the transmitter and receiver.

Altera provides an embedded reset controller, but you can also provide your own user-coded reset controller.

Table 3-1: Stratix V Reset Control Options

Transceiver PHY IP Core	Embedded Reset Controller	User-Coded Reset Controller	Transceiver PHY Reset Controller IP	Avalon Memory-Mapped Reset Registers
XAUI	Yes			Yes
PCI Express	Yes			Yes
10GBASE-R	Yes	Yes	Yes	Yes
Interlaken	Yes			Yes
Custom Configuration	Yes	Yes	Yes	Yes
Low Latency	Yes	Yes	Yes	Yes
Deterministic Latency	Yes	Yes	Yes	Yes
Native PHY		Yes	Yes	

Related Information

[Stratix V Device Handbook: Known Issues](#)

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

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.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

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.

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

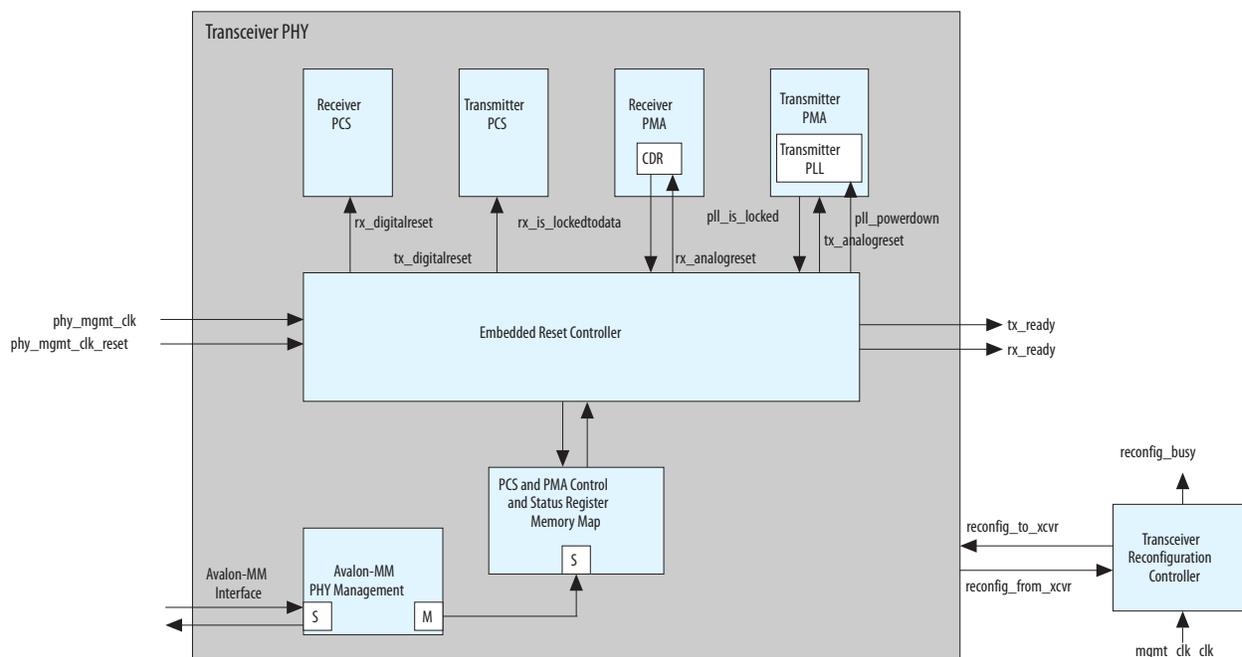


Table 3-2: Embedded Reset Controller Reset Control and Status Signals

Signal Name	Signal	Description
<code>phy_mgmt_clk</code>	Control Input	Clock for the embedded reset controller.

Signal Name	Signal	Description
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.

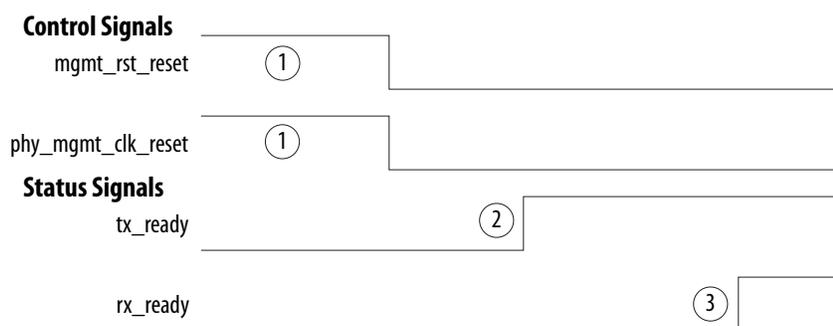
Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Power-Up

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

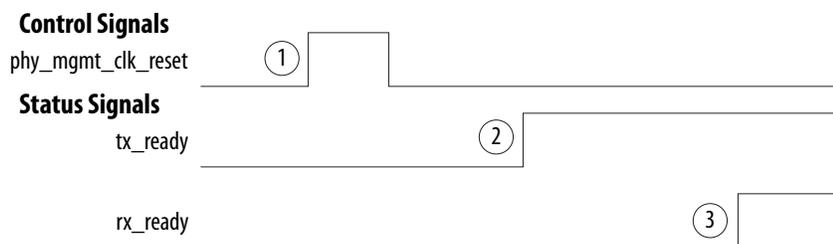
Resetting the Transceiver with the PHY IP Embedded Reset Controller During Device Operation

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.

Related Information

- [Transceiver Reset Using Avalon Memory Map Registers](#) on page 3-11
- [Transceiver Reset Control Signals Using Avalon Memory Map Registers](#) on page 3-11

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.

Note: If you do not follow the Stratix V power sequence, you might encounter transceiver functionality issues, especially with regard to the `pll_locked` signal.

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.

This reset controller comes with a clear text Verilog file that you modify based on your requirements.

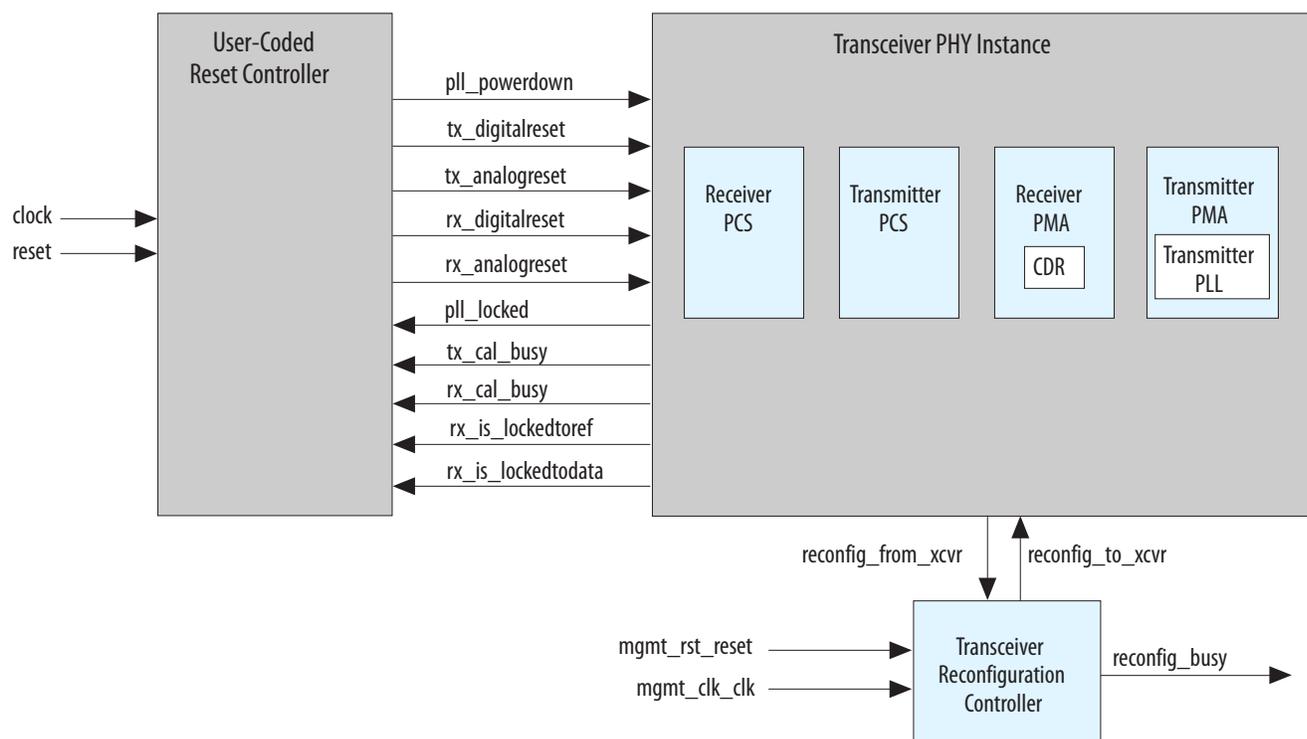
Related Information

["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
<code>mgmt_clk_clk</code>	Clock	Clock for the Transceiver Reconfiguration Controller. This clock must be stable before releasing <code>mgmt_rst_reset</code> .
<code>mgmt_rst_reset</code>	Reset	Reset for the Transceiver Reconfiguration Controller
<code>pll_powerdown</code>	Control	Resets the TX PLL when asserted high
<code>tx_analogreset</code>	Control	Resets the TX PMA when asserted high
<code>tx_digitalreset</code>	Control	Resets the TX PCS when asserted high
<code>rx_analogreset</code>	Control	Resets the RX PMA when asserted high
<code>rx_digitalreset</code>	Control	Resets the RX PCS when asserted high
<code>reconfig_busy</code>	Status	A high on this signal indicates that reconfiguration is active
<code>tx_cal_busy</code>	Status	A high on this signal indicates that TX calibration is active
<code>rx_cal_busy</code>	Status	A high on this signal indicates that RX calibration is active

Signal Name	Signal Type	Description
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

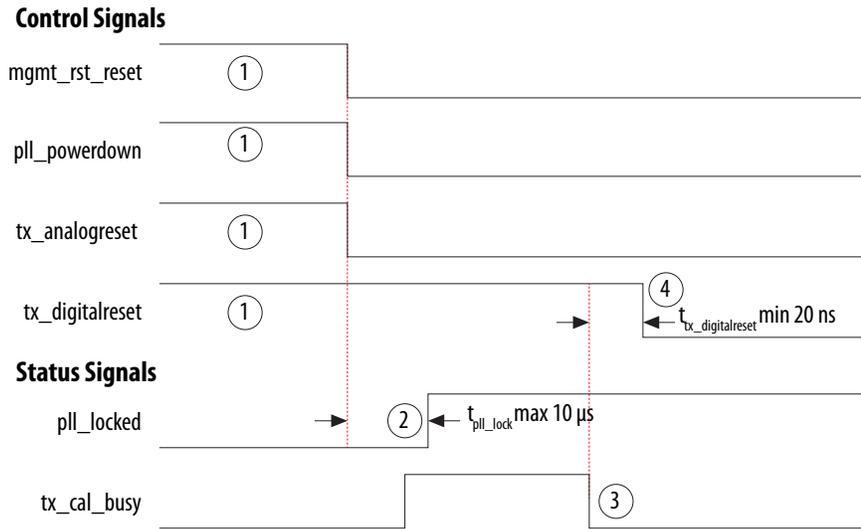
Follow this reset sequence when designing your User-Coded Reset Controller to ensure a reliable transmitter initialization after the initial power-up.

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 μ s ($t_{pll_powerdown}$). If you use ATX PLL calibration, 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

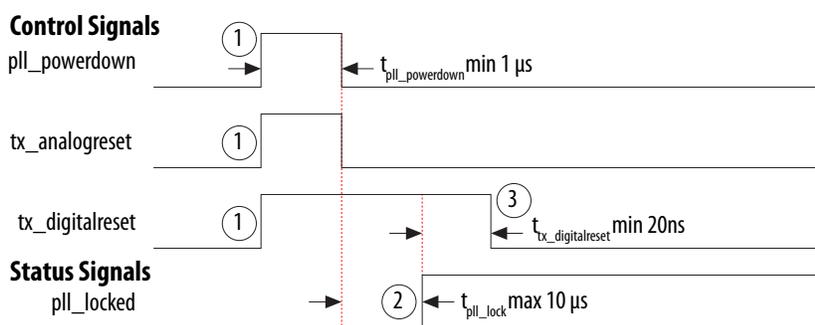
Follow this reset sequence if you want to reset the PLL, or analog or digital blocks of the transmitter at any point during device operation. This might be necessary for re-establishing a link or after certain dynamic reconfigurations.

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

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



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`.

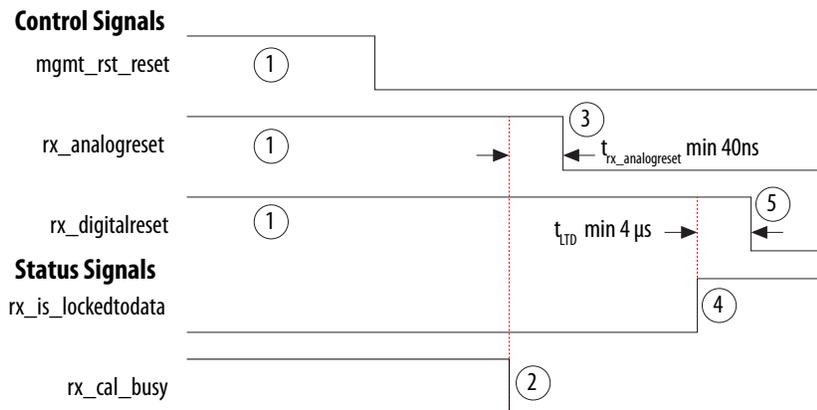
The receiver is now out of reset and ready for operation.

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 Stratix V Devices](#)

For information about CDR lock modes.

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

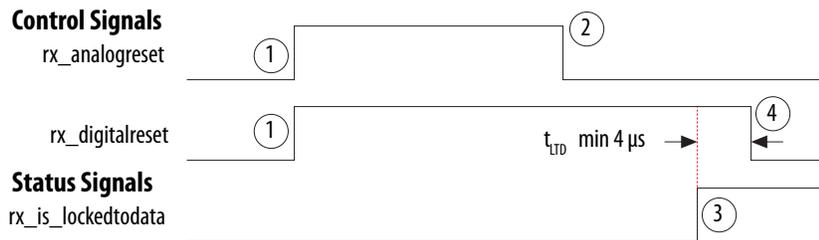
Follow this reset sequence to reset the analog or digital blocks of the receiver at any point during the device operation. This might be necessary for re-establishing a link or after certain dynamic reconfigurations.

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 Stratix 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
<code>pma_rx_set_locktodata</code>	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.
<code>pma_rx_set_locktoref</code>	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 <code>pma_rx_set_locktodata</code> is not asserted. The default is low when both registers have the CDR in auto lock mode.

Register Name	Description
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 Altera 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

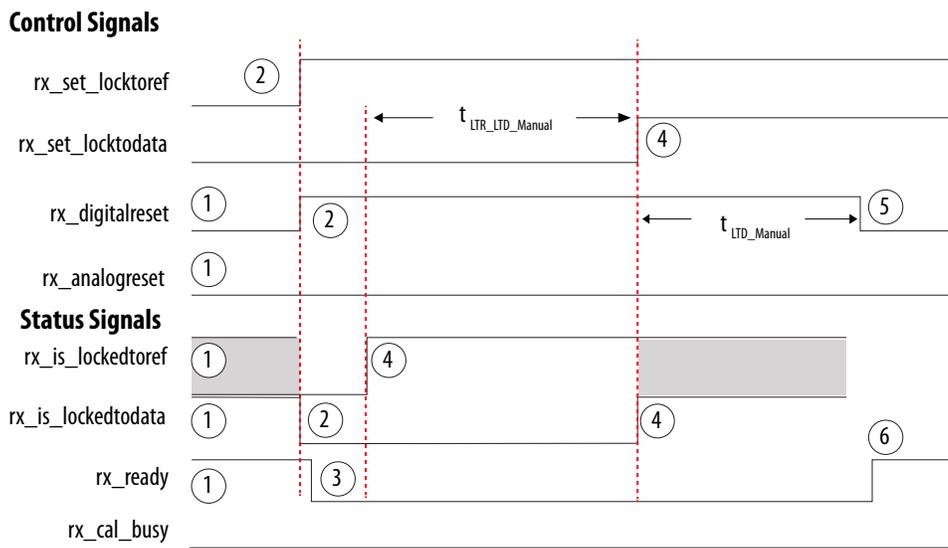
rx_set_locktoref	rx_set_locktodata	CDR Lock Mode
0	0	Automatic
1	0	Manual-RX CDR LTR
X	1	Manual-RX CDR LTD

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_locktoref` 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 μ s) after the CDR is locked to reference. `rx_is_locktoref` should be high and stable for a minimum $t_{LTR_LTD_Manual}$ (15 μ s), 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 μ s).
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



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_analogreset	tx_digitalreset	tx_analogreset
PLL					
CMU PLL	Yes	—	—	—	—
ATX PLL	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	—	—	—

Transceiver Block	pll_ powerdown	rx_digital- reset	rx_analogr- reset	tx_digitalreset	tx_analogreset
Receiver 10G PCS					
Receiver Gear Box	—	Yes	—	—	—
Receiver Block Synchronizer	—	Yes	—	—	—
Receiver Disparity Checker	—	Yes	—	—	—
Receiver Descrambler	—	Yes	—	—	—
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					
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					

Transceiver Block	pll_powerdown	rx_digital-reset	rx_analogr-eset	tx_digitalreset	tx_analogreset
Transmitter Central/Local Clock Divider	—	—	—	—	Yes
Serializer	—	—	—	—	Yes
Transmitter Buffer	—	—	—	—	Yes

Document Revision History

Date	Version	Changes
January 2016	2016.01.11	Added third note to the "User-Coded Reset Controller" section.
September 2014	2014.09.30	<ul style="list-style-type: none"> Added statement at the top of the chapter that an embedded reset controller is provided, but you can provide your own reset controller instead. Added information about deasserting signals to the "Resetting the Transceiver with the PHY IP Embedded Reset Controller during Device Power-Up" section. Added "Control Signals" and "Status Signals" labels to the timing diagrams. Added a link to the Related Links in the "Clock Data Recovery in Manual Lock Mode" section.
October 2013	2013.10.11	<ul style="list-style-type: none"> Changed term of User-Controlled Reset Controller to User-Coded Reset Controller.
May 2013	2013.05.06	<ul style="list-style-type: none"> Added information about ATX PLL calibration. Changes to figures to reflect correct values for t_{pll_lock}. Added link to the known document issues in the Knowledge Base.
February 2013	2013.02.21	<ul style="list-style-type: none"> Added information about reset options. Clarified content about resetting the transmitter with the user-controlled reset controller during power-up. Updated incorrect description of <code>pll_powerdown</code> when using Memory Map registers.
December 2012	2012.12.17	<ul style="list-style-type: none"> Added information about resetting the transceiver during power-up and device operation Restructured document

Date	Version	Changes
June 2012	3.3	<ul style="list-style-type: none">• Updated for the Quartus II software version 12.0.• Revised Figure 3-2 , Figure 3-4 , and Figure 3-6 .• Added new Table 3-1.• Added new steps describing wave forms.• Deleted power-up Figures 3-2 and 3-5 and changed text accordingly.

2017.02.15

SV52005



Subscribe



Send Feedback

Stratix® V 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 4-1](#).

Stratix V 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

- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the *Stratix V Device Handbook* chapters.
- [Upcoming Stratix V Device Features](#)
- [Altera Transceiver PHY IP Core User Guide](#)

Protocols and Transceiver PHY IP Support

Table 4-1: 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) ⁽¹⁰⁾	Standard and Gen3	Yes	Embedded

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered



Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
PCIe Gen2 x1, x2, x4, x8	PHY IP Core for PCIe (PIPE) ⁽¹⁰⁾	Standard	Yes	Embedded
PCIe Gen1 x1, x2, x4, x8	PHY IP Core for PCIe (PIPE) ⁽¹⁰⁾	Standard	Yes	Embedded
10GBASE-R	10GBASE-R	10G	Yes	Embedded
	Native PHY	10G	No	External Reset IP
10G/40/100G 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
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 ⁽¹¹⁾	10G	No	External Reset IP
OTU-4 (100G) via OIF SFI-S	Low Latency PHY	10G	Yes	External Reset IP
	Native PHY	10G	No	External Reset IP

⁽¹⁰⁾ Hard IP for PCI Express is also available as a core function.

⁽¹¹⁾ A Soft-PCS bonding IP is required.

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
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 +/SFP-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
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

Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
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
16/10G 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
SRIO 2.2/1.3 ⁽¹²⁾	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

⁽¹²⁾ Nx Multi-Alignment Deskew State Machine must be implemented in the core.

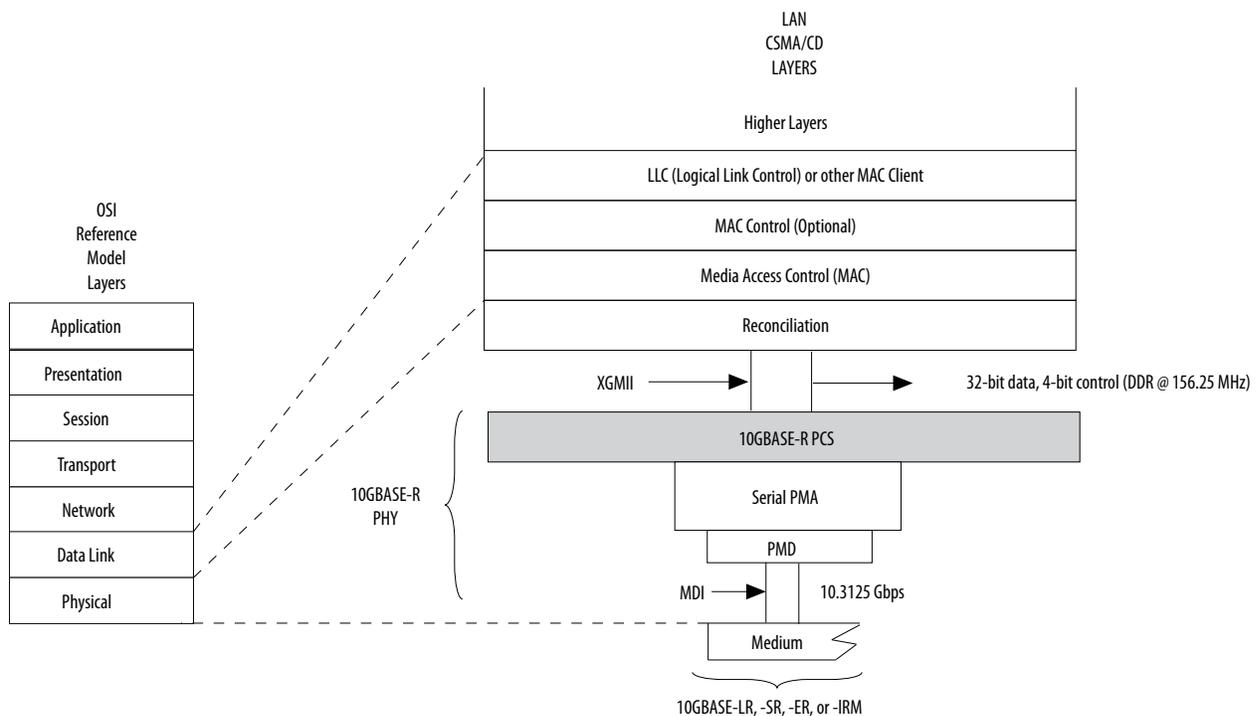
Protocol Standard	Transceiver IP	PCS Type	Avalon-MM Register Interface	Reset Controller
HiGig+/2+	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
JESD204A	Custom PHY	Standard	Yes	Embedded or External Reset IP
	Native PHY	Standard	No	External Reset IP
ASI	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

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 Stratix V 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.

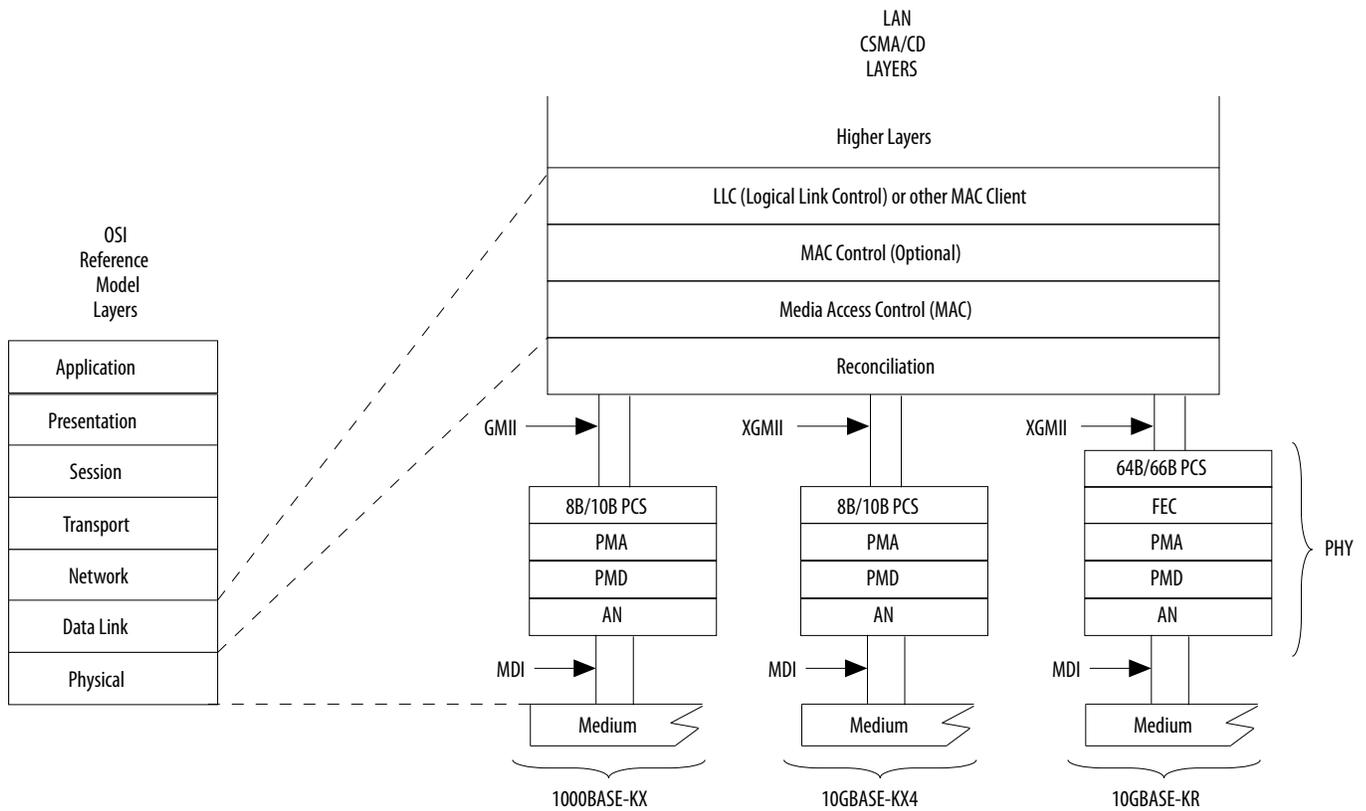
Figure 4-1: 10GBASE-R PHY Connection to IEEE802.3 MAC and RS



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 1 Gbps 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 1 Gbps 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 4-2: 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 4-3: 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.

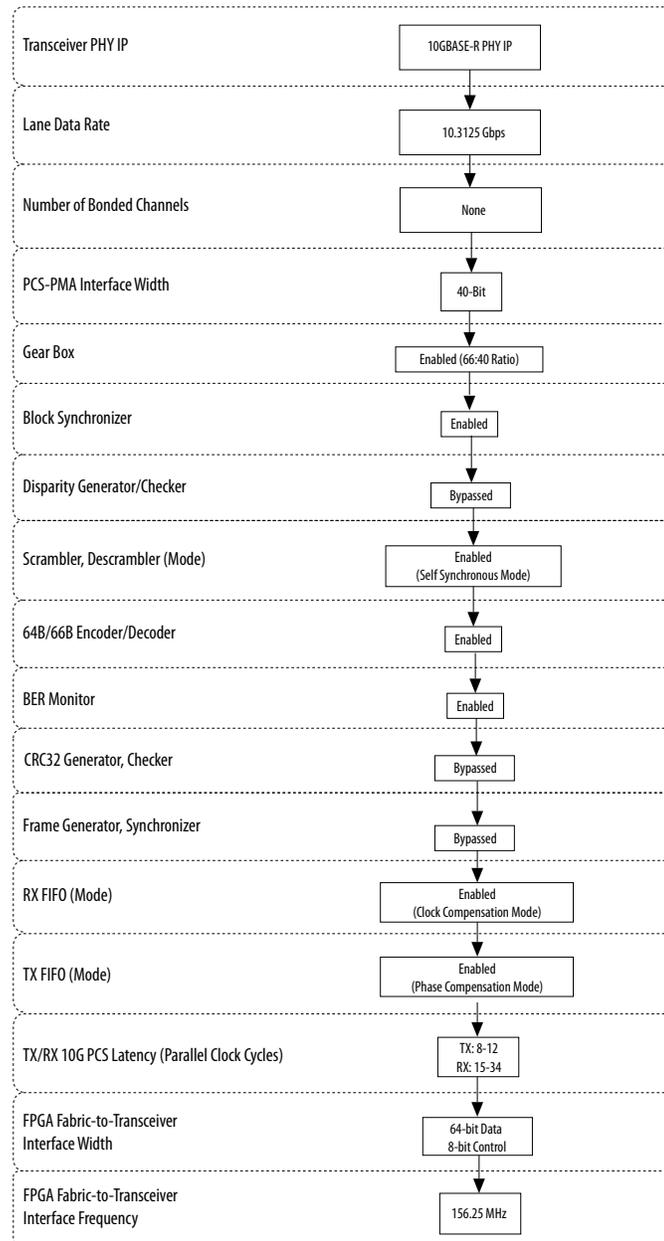
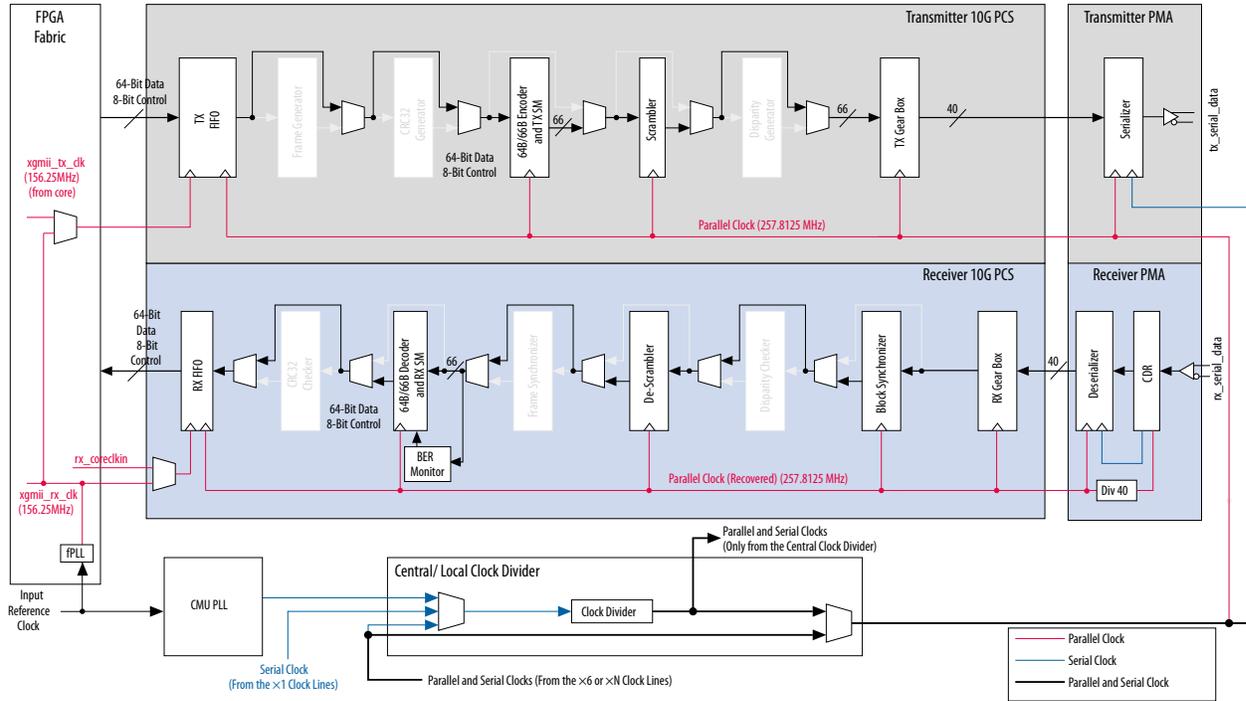


Figure 4-4: Transceiver Channel Datapath for a 10GBASE-R Configuration



10GBASE-KR

Figure 4-5: 10GBASE-R/KR and 1000Base-X/KX Datapath Configuration

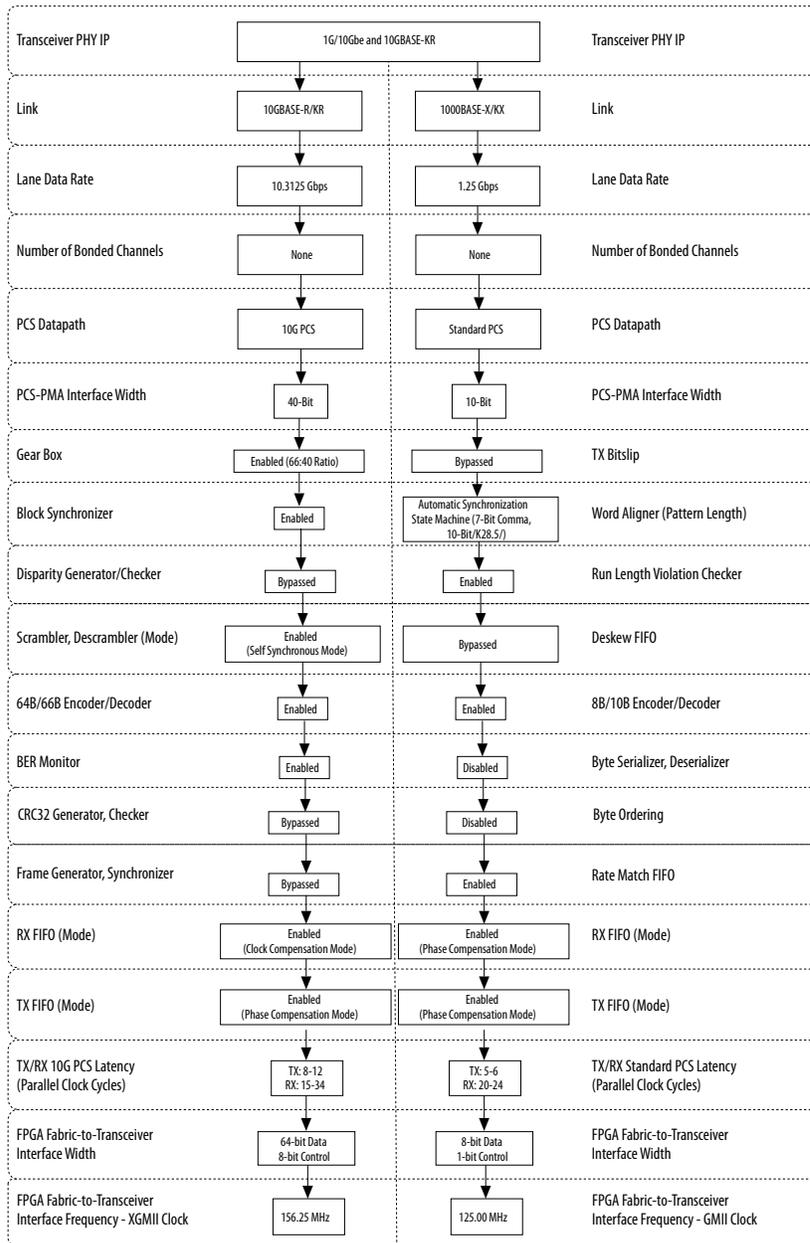
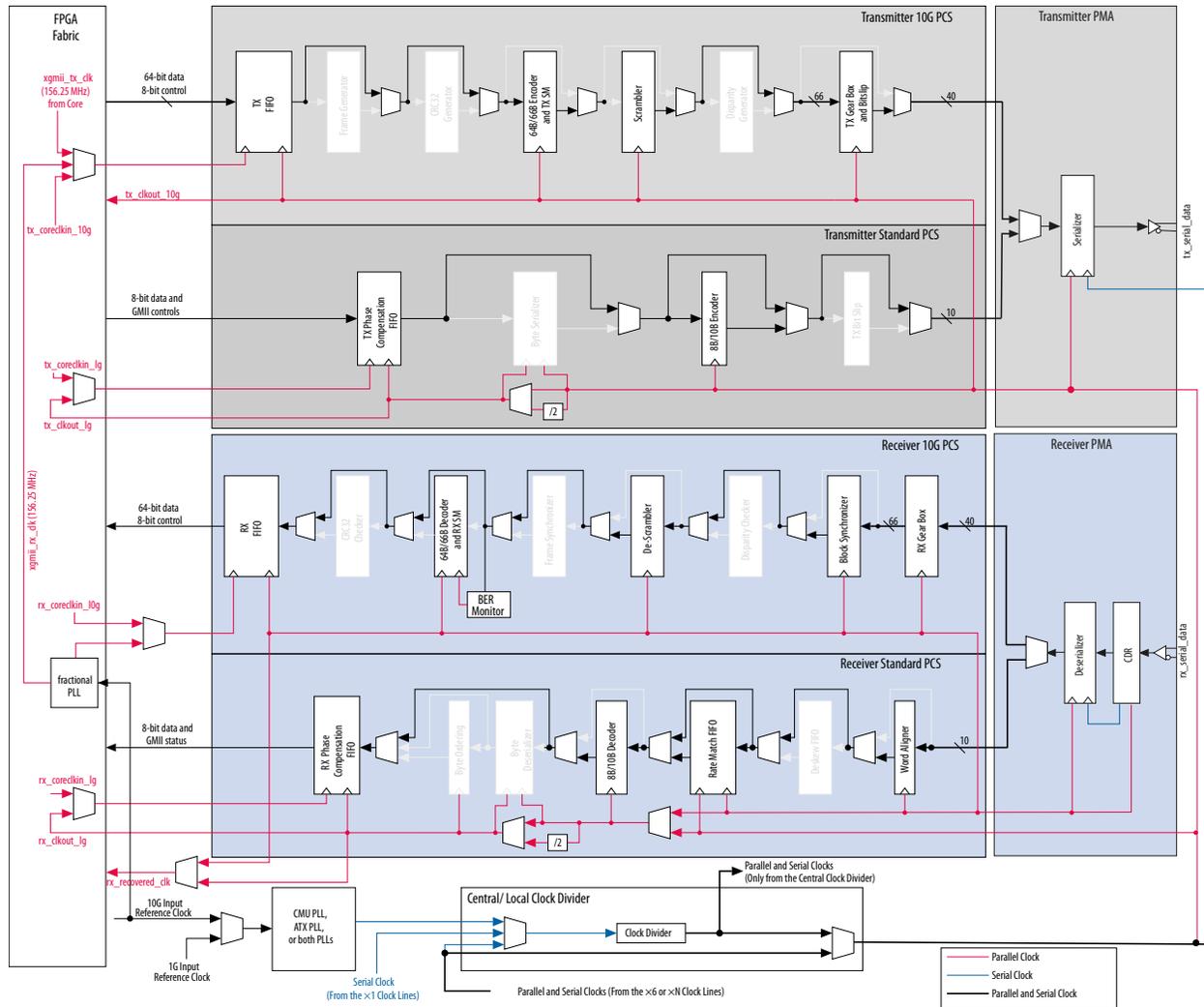


Figure 4-6: Transceiver Channel Datapath for 10GBASE-R/KR and 1000BASE-X/KX Configuration



10GBASE-R and 10GBASE-KR Supported Features

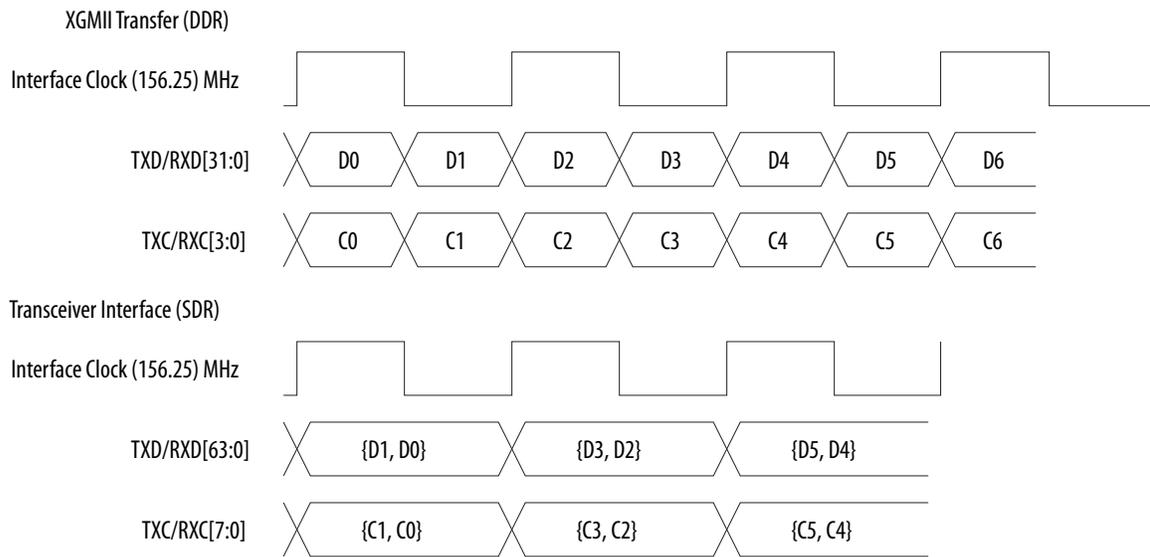
The following features are supported by the transceivers in 10GBASE-R and 10GBASE-KR configurations.

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 4-7: XGMII Interface (DDR) versus Stratix V 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 $1 + x^{39} + x^{58}$, 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 μ s 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.

Stratix V GX 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

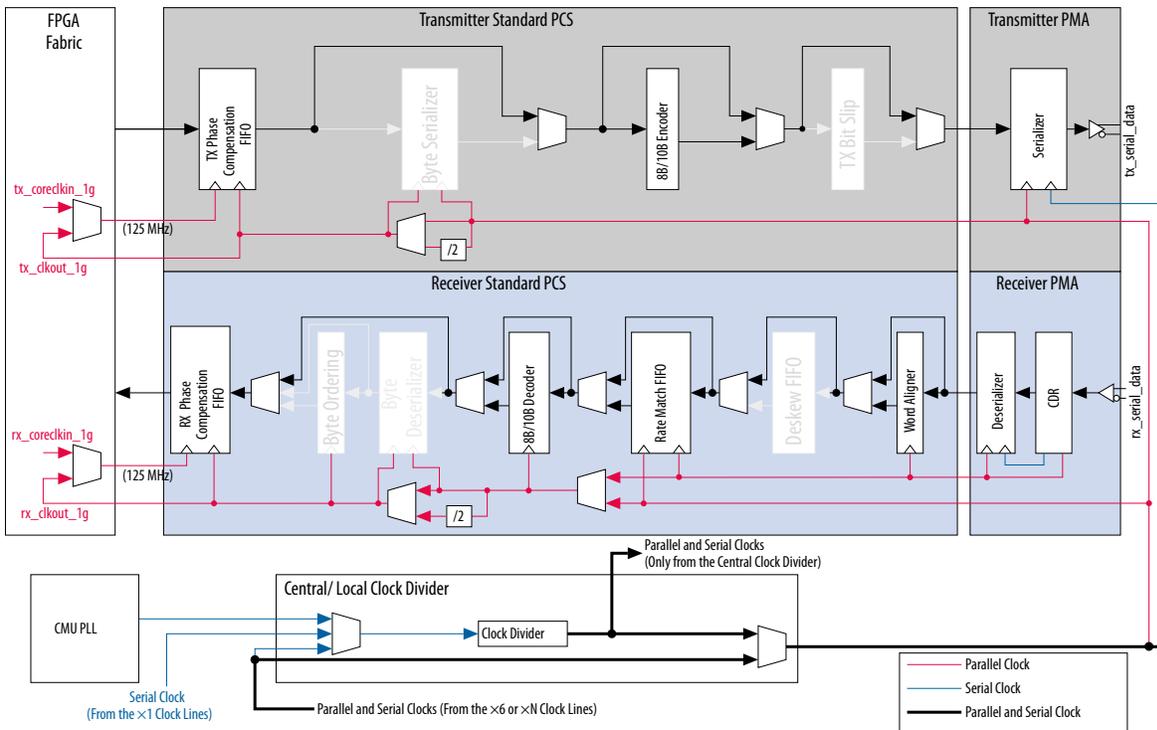
The FEC function defined in clause 74 of IEEE 802.3ap-2007 specification must be implemented in the core fabric. In Stratix V devices, the hard PCS does not support applications that require FEC function-

ality. 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.

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 4-8: 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.

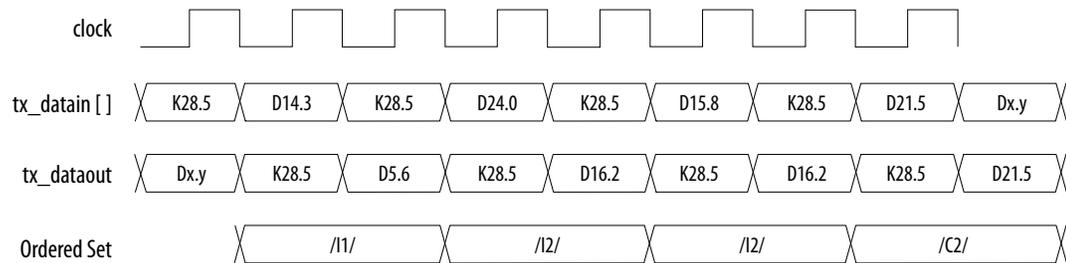
Idle Ordered-Set Generation in 1000BASE-X and 1000BASE-KX Configurations

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 4-9: Example of Automatic Ordered Set Generation



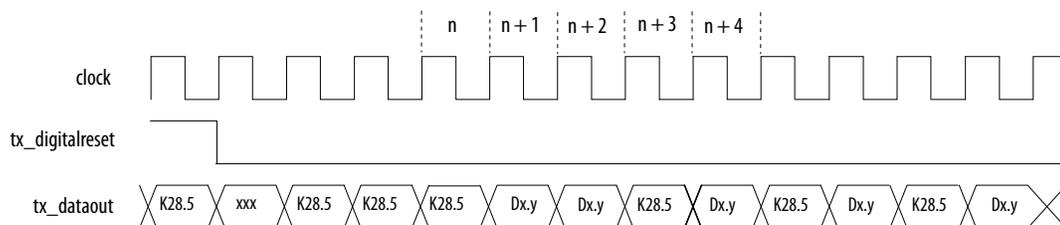
Reset Condition in 1000BASE-X and 1000BASE-KX Configurations

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 4-10: Example of Reset Condition in 1000BASE-X and 1000BASE-KX Configurations



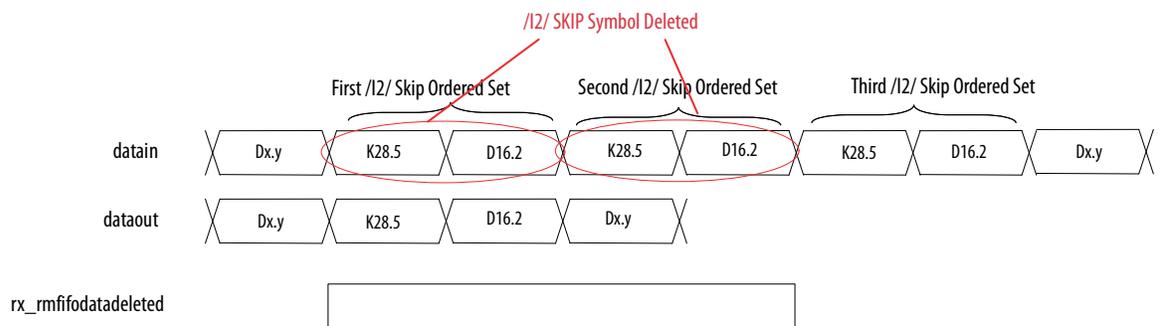
Rate Match FIFO in 1000BASE-X and 1000BASE-KX Configurations

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.

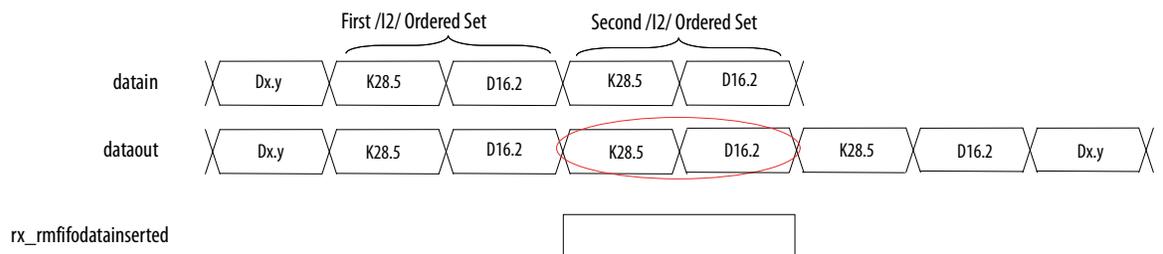
The following figure shows an example of rate match FIFO deletion where three symbols are required to be deleted. Because the rate match FIFO can only delete /I2/ ordered set, it deletes two /I2/ ordered sets (four symbols deleted).

Figure 4-11: 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 4-12: Example Rate Match Insertion in 1000BASE-X and 1000BASE-KX Configurations



Two register bits, `rx_rmifodatadeleted` and `rx_rmifodatainserted`, indicate rate match FIFO deletion and insertion events. Both the `rx_rmifodatadeleted` and `rx_rmifodatainserted` 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 4-2: 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 4-3: 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

PHY IP Type	PHY Type	Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
1G/10GbE and 10GBASE-KR PHY IP	1000BASE-X and 1000BASE-KX	125, 62.5	8-bit data, gmi i_ tx_en and gmi i_ 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.

Stratix V devices support a transmission speed of up to 14.1 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 4-13: 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.

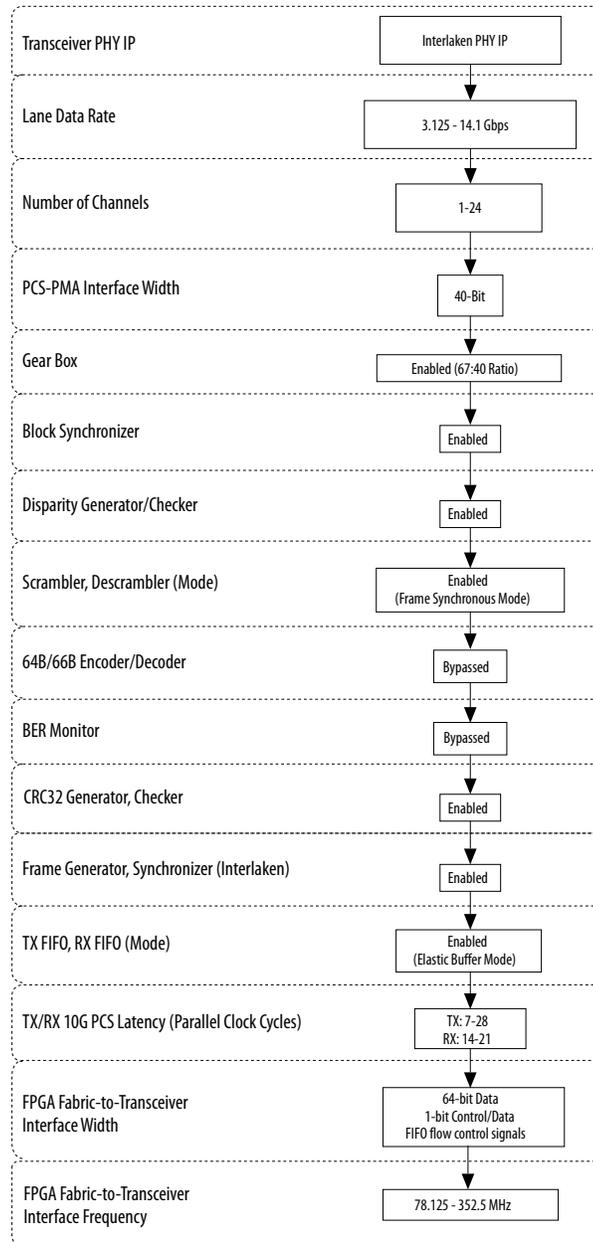
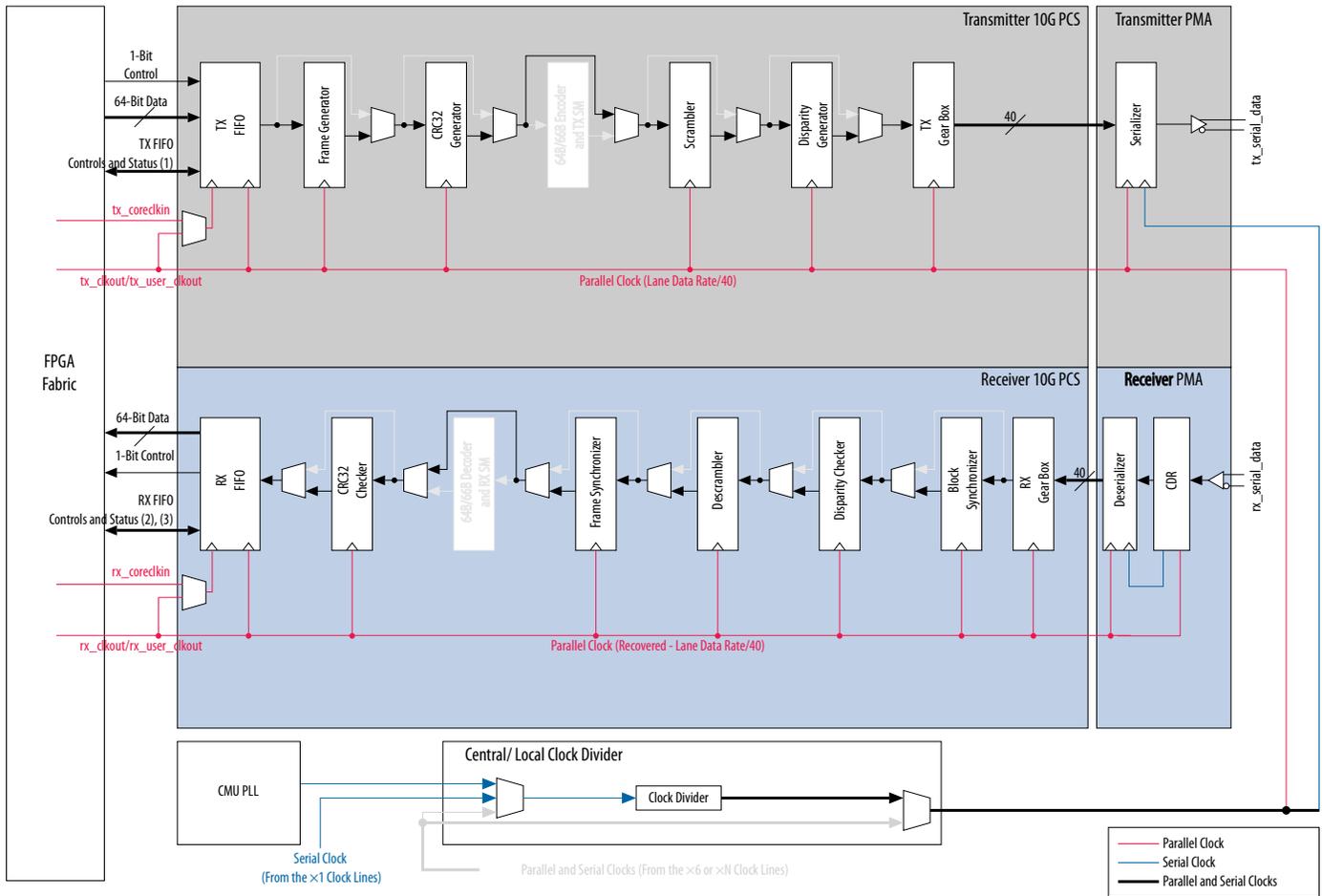


Figure 4-14: Transceiver Channel Datapath for Interlaken Configuration



- 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 4-4: 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 $x^{58} + x^{39} + 1$ 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 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.

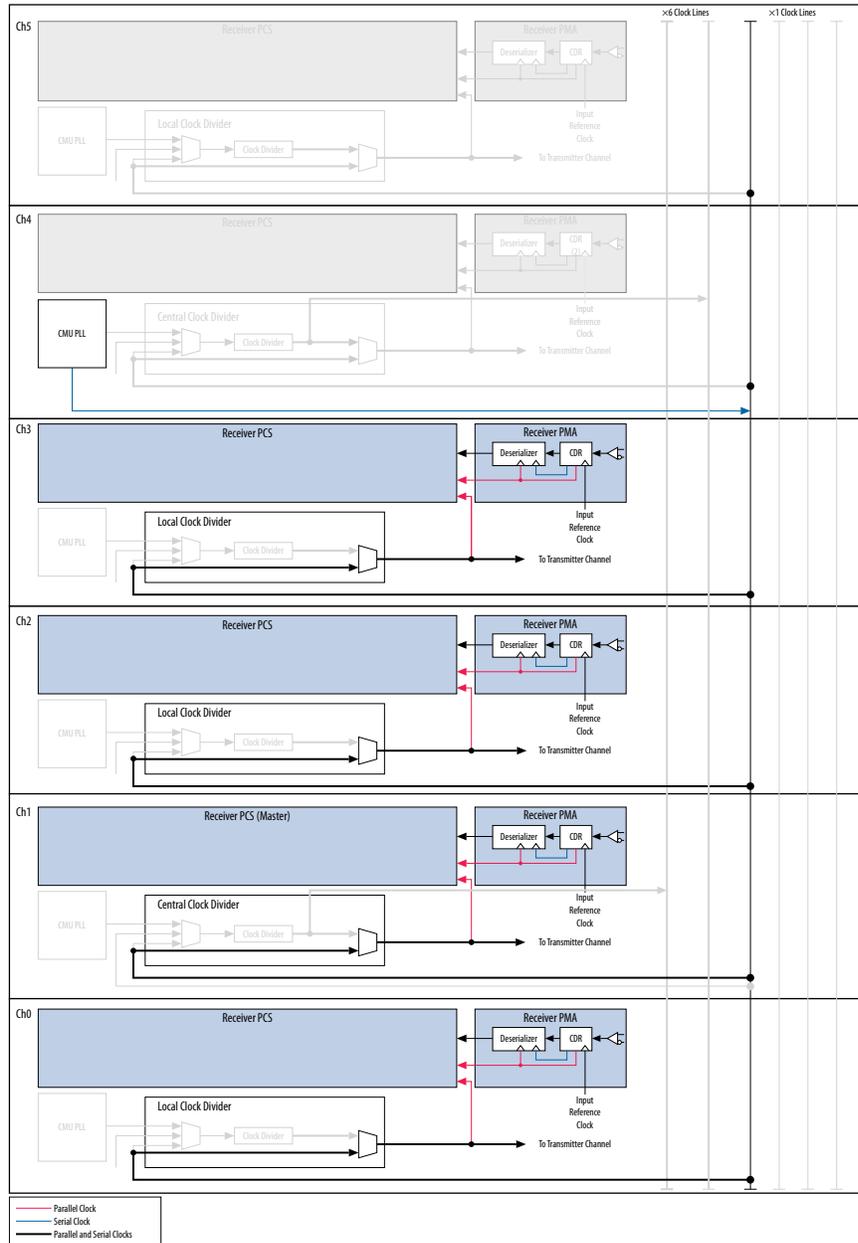
Related Information

- [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 4-15: 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 a Stratix V 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:

- Stratix V 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 4-5: Support for Transceivers

Support	Stratix V 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.

Stratix V 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

- [Stratix 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 4-16: PCIe Gen1 and Gen2 PIPE Datapath Configuration

This transceiver datapath configuration is for a configuration without Gen3 enabled.

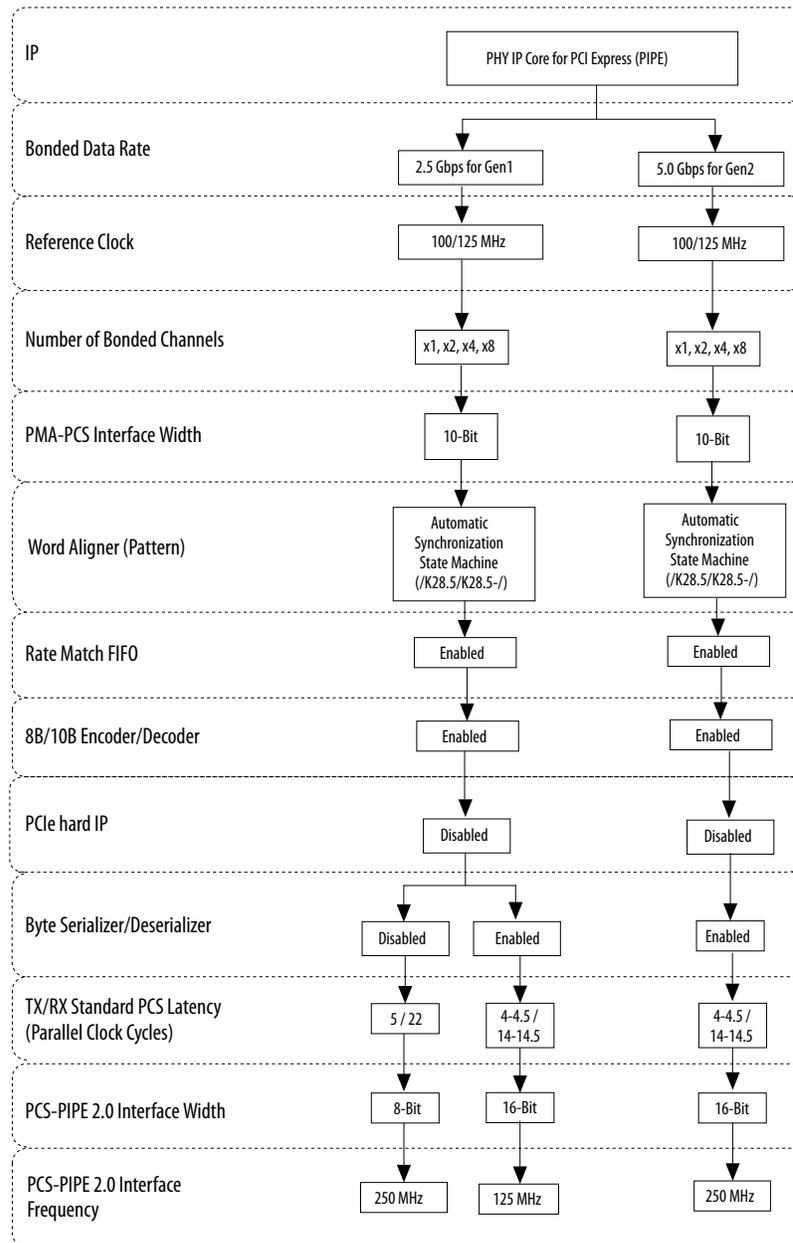
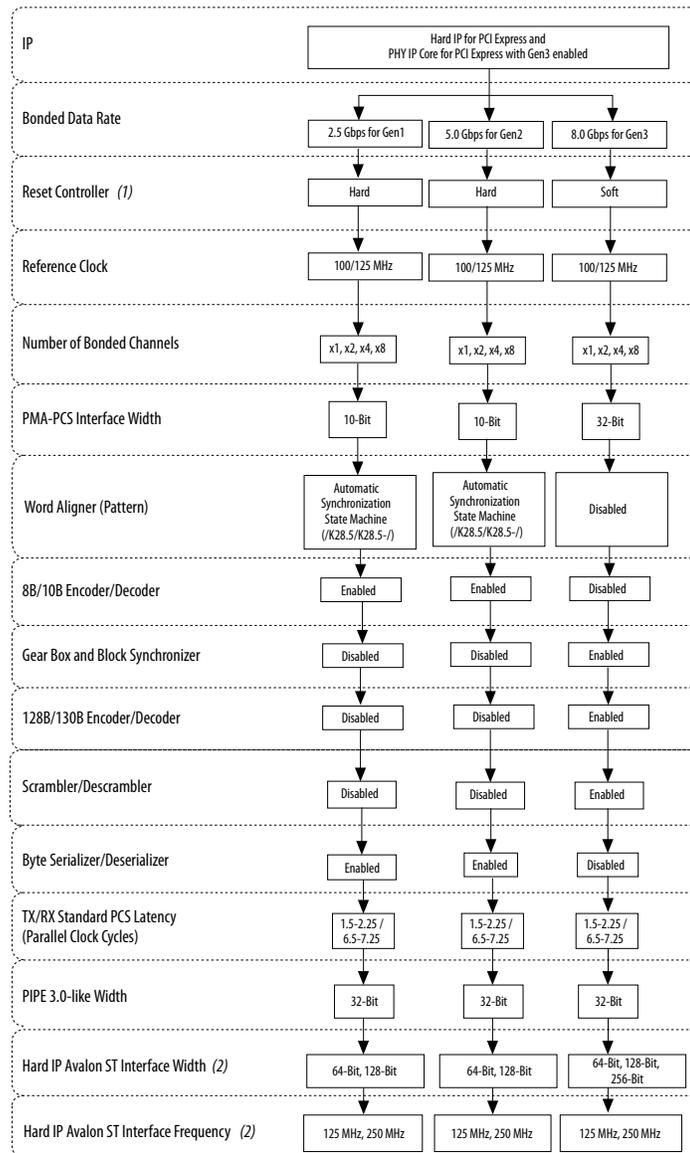


Figure 4-17: PCIe Gen1, Gen2, and Gen3 Hard IP and PHY IP Core for PCI Express Datapath Configuration

This transceiver datapath configuration is for a configuration with Gen3 enabled.



Notes:

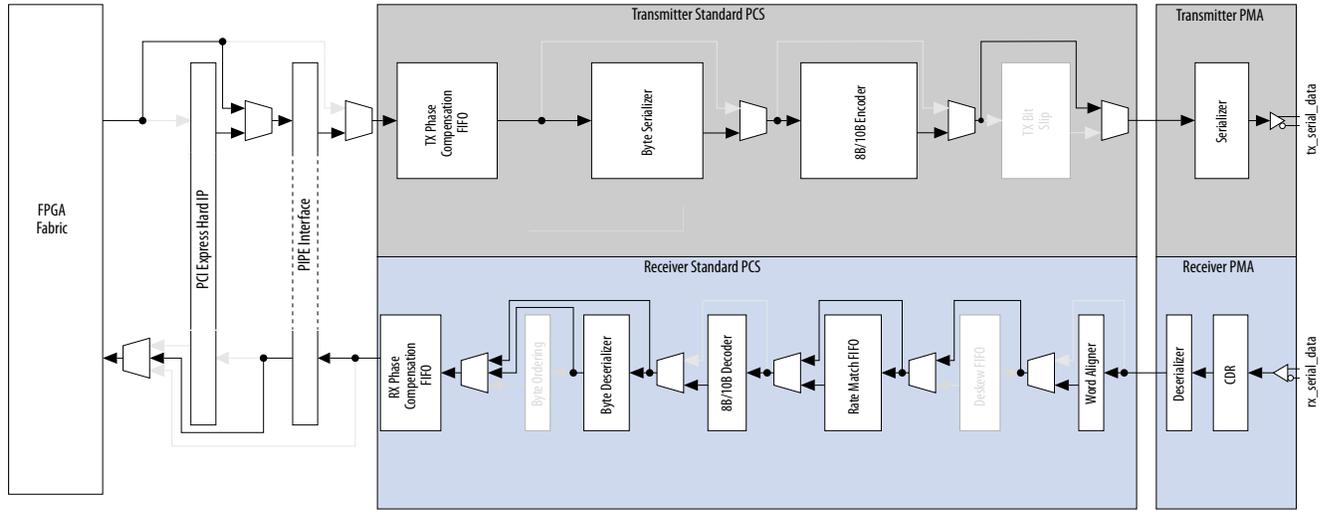
(1) The PHY IP Core for PCI Express (PIPE configuration) employs the Embedded Reset Controller IP. It does not use the Hard or Soft Reset Controller employed in the Hard IP for PCI Express (HIP configuration).

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

Transceiver Channel Datapath

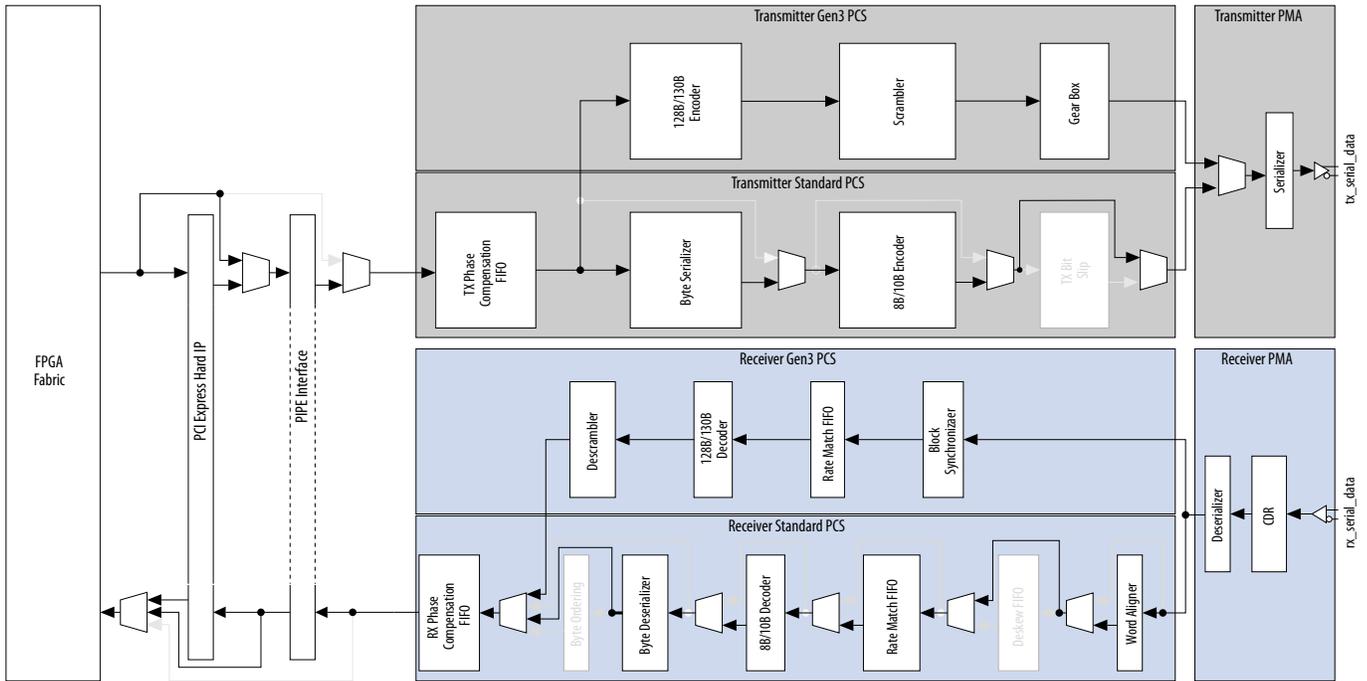
The following figure shows the Stratix V transmitter and receiver channel datapath for PCIe Gen1/Gen2 configurations when using PIPE configuration with Gen3 disabled. In this configuration, the transceiver connects to a PIPE 2.0 compliant interface.

Figure 4-18: Transceiver Channel Datapath for PCIe Gen1/Gen2 in PIPE Configuration with Gen3 Disabled



The following figure shows the Stratix V 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 4-19: Transceiver Channel Datapath for PCIe Gen1/Gen2/Gen3 Configurations



Related Information

[Transceiver Architecture in Stratix 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 4-6: 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 Stratix V 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 Stratix V 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. Stratix V

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 Stratix V 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, Stratix V 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.

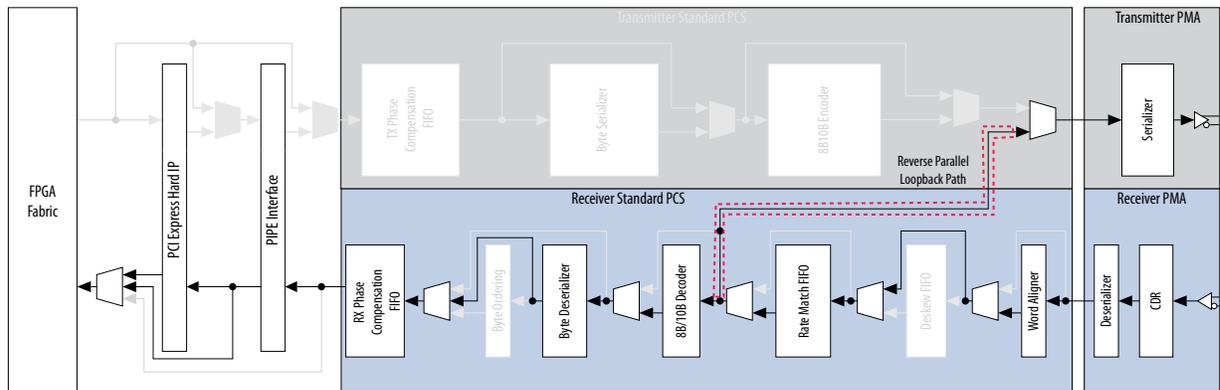
PCIe Reverse Parallel Loopback

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. Stratix V devices provide an input signal to enable this loopback mode.

Note: This is the only loopback option supported in PIPE configurations.

Figure 4-20: PCIe Reverse Parallel Loopback Mode Datapath

The grayed-out blocks are Inactive.



Related Information

- [Transceiver Clocking and Channel Placement Guidelines](#) on page 4-35
- [Refer to the PHY IP Core for PCI Express \(PIPE\) chapter in the Altera Transceiver PHY IP Core User Guide](#)
- [Refer to the “Standard PCS Architecture” section in the Transceiver Architecture in Stratix 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 Stratix V Hard IP for PCI Express IP or PHY IP Core for PCI Express.

In Stratix V 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 4-7: 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 μ s in Recovery.RCVRLOCK. The `rx_is_lockedto data` 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

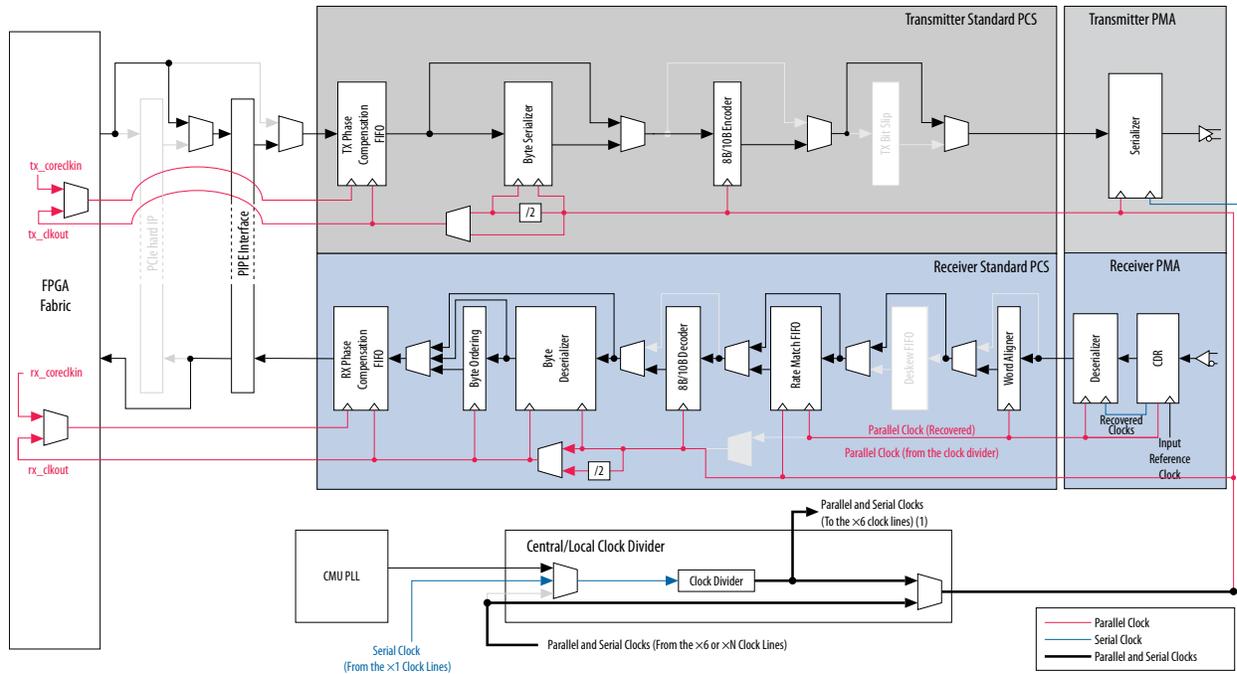
This section describes the transceiver clocking for Gen1 and Gen2 Hard IP and PIPE configurations. The channel placement guidelines are only described for Gen1 and Gen2 PIPE configuration. The channel placement guidelines for Gen1 and Gen2 Hard IP configuration are not included.

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.

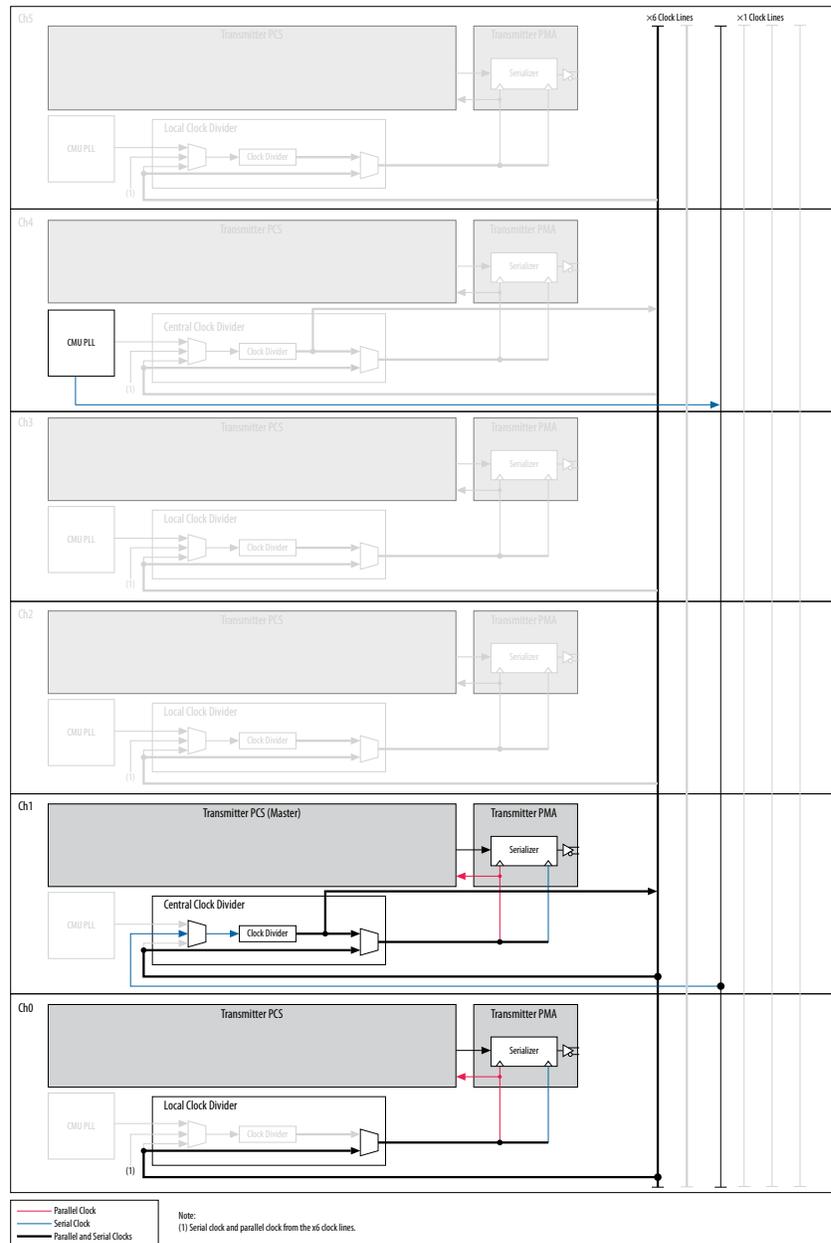
Figure 4-21: Transceiver Clocking in a Gen1/Gen2 PIPE x1 Configuration



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 4-22: Transmitter Clocking in a Gen1/Gen2 PIPE x2 Configuration



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 4-23: Transmitter Clocking in a Gen1/Gen2 PIPE x4 Configuration

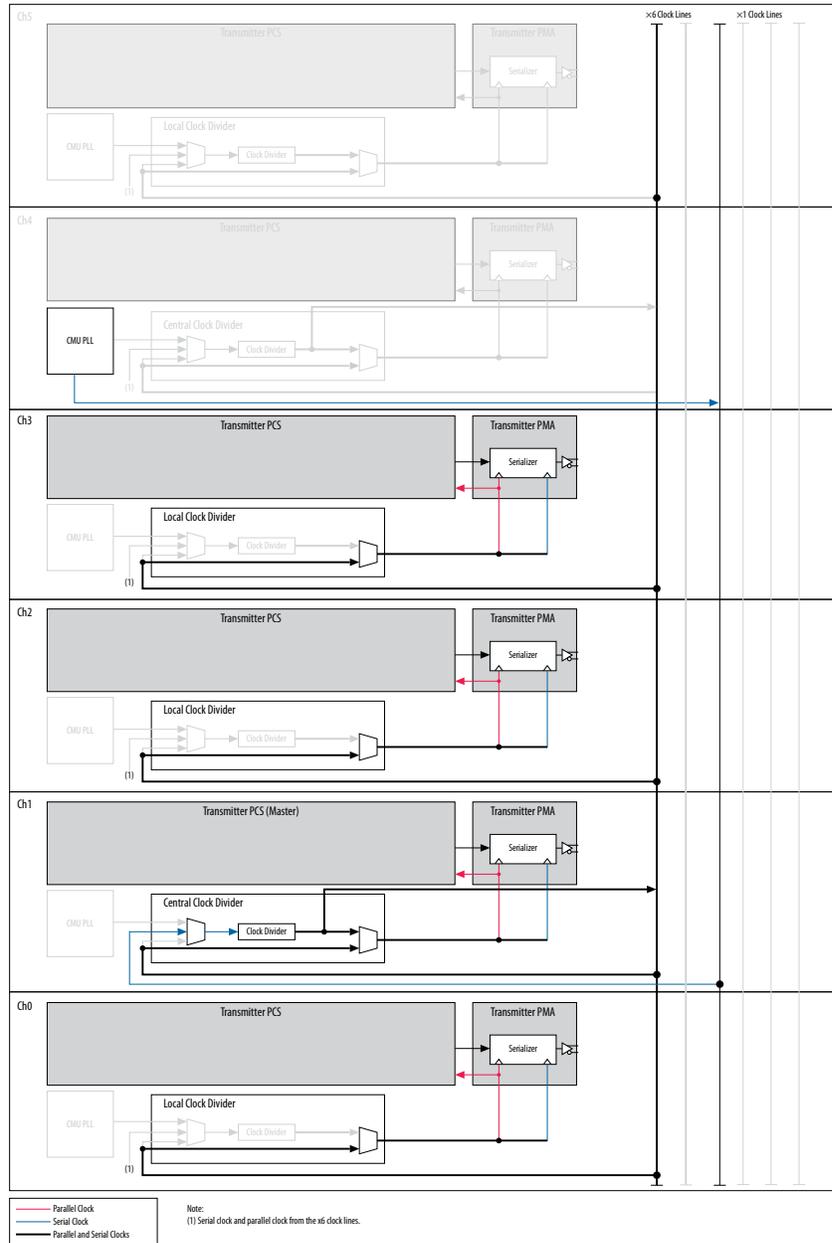
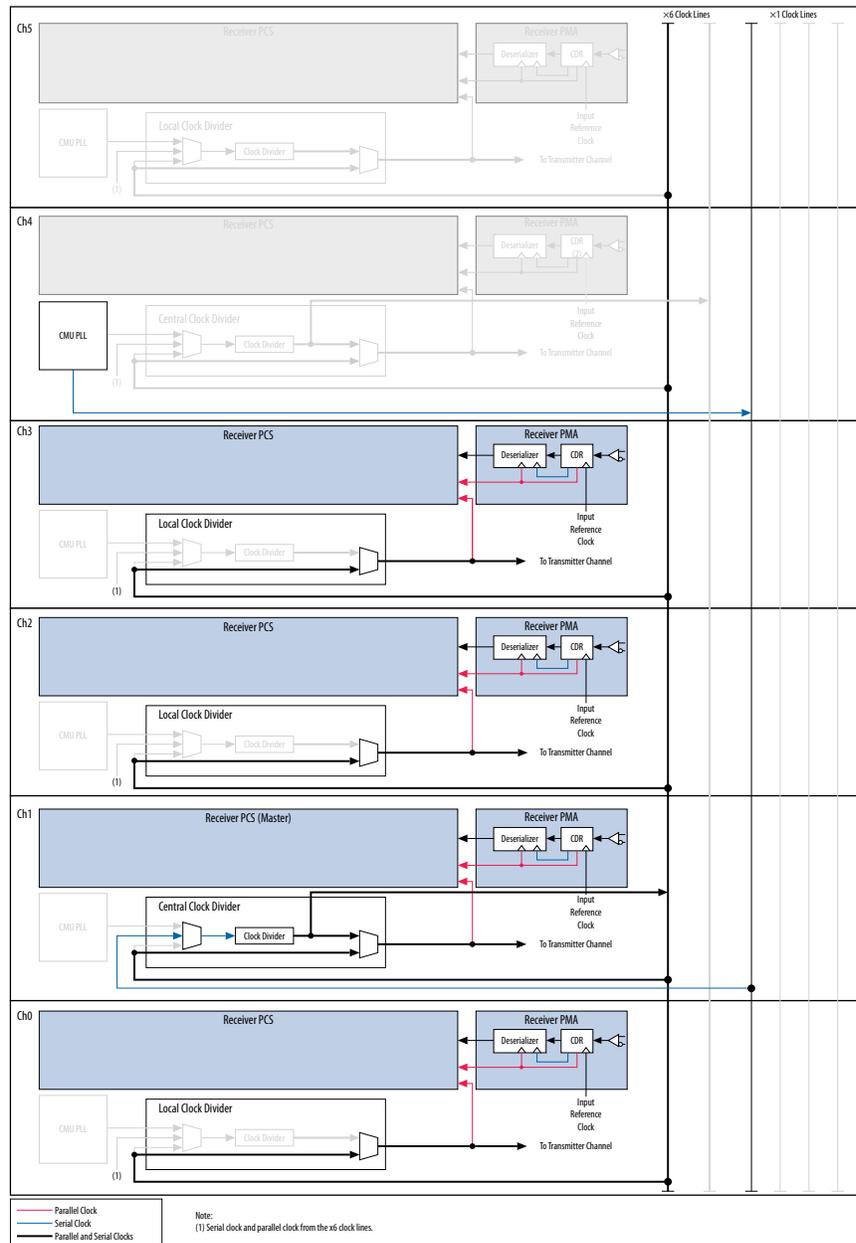


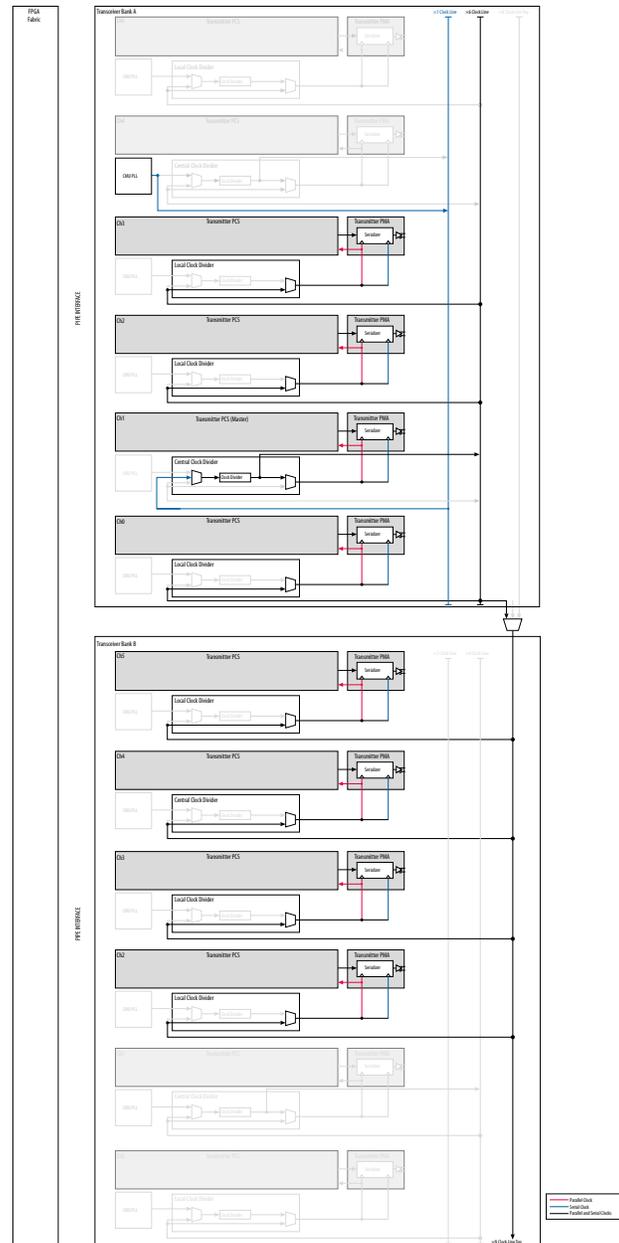
Figure 4-24: Receiver Clocking in a Gen1/Gen2 PIPE x4 Configuration



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 4-25: 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 4-8: 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 4-26: Example of PIPE x2 Gen1, Gen2, and Gen3 Channel Placement Using an ATX PLL, a CMU PLL, or Both

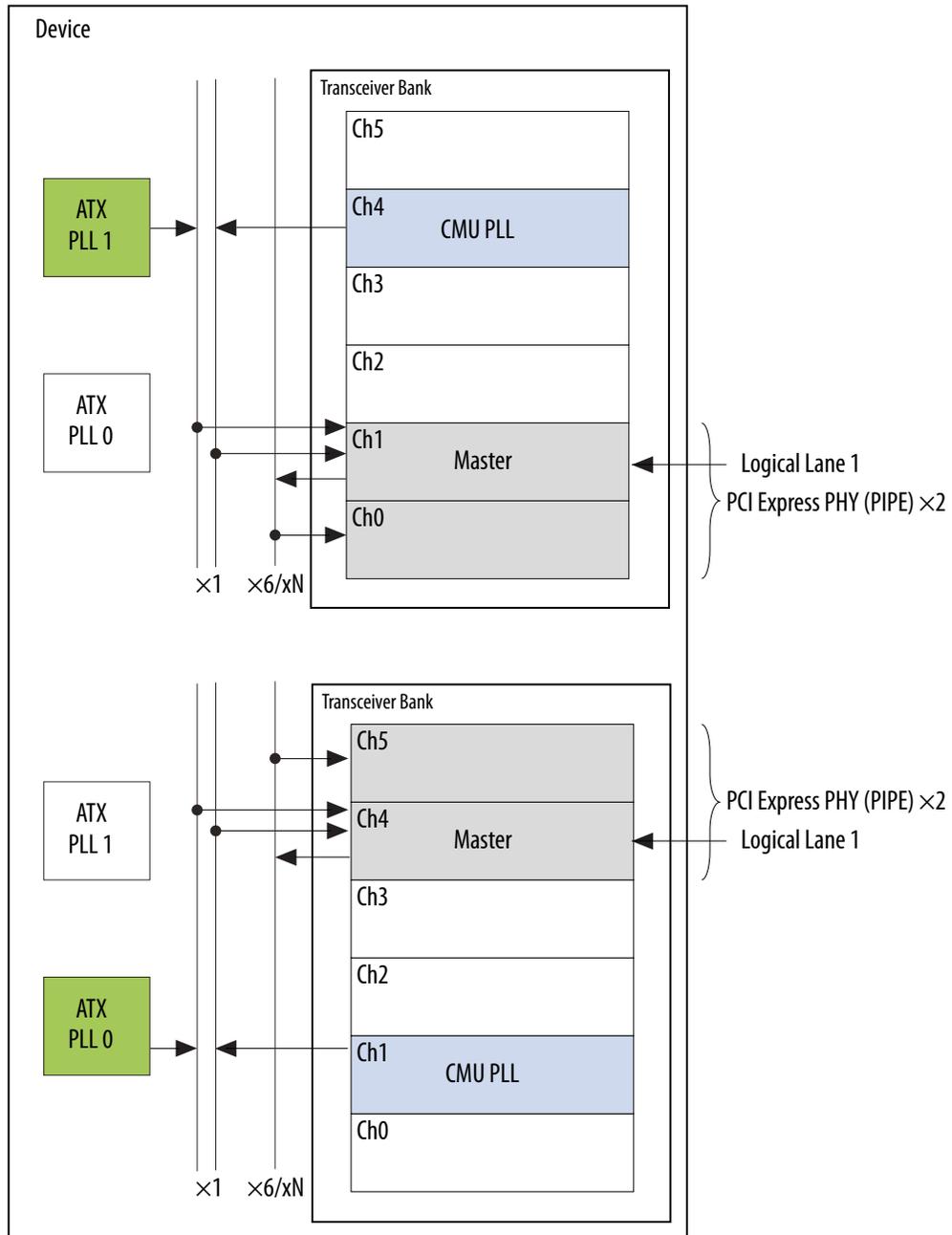
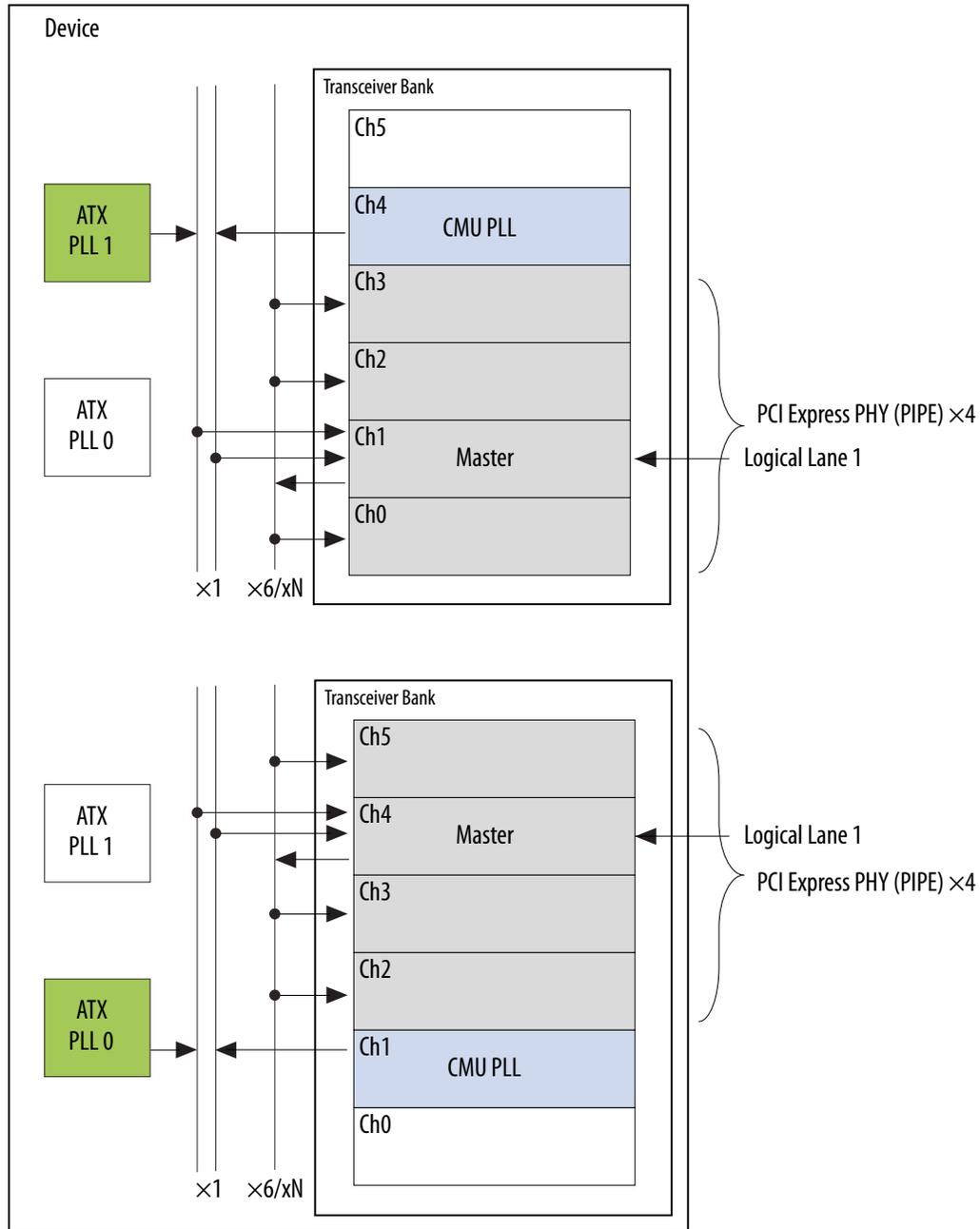


Figure 4-27: Example of PIPE x4 Gen1, Gen2, and Gen3 Channel Placement Using an ATX PLL, a CMU PLL, or Both

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.

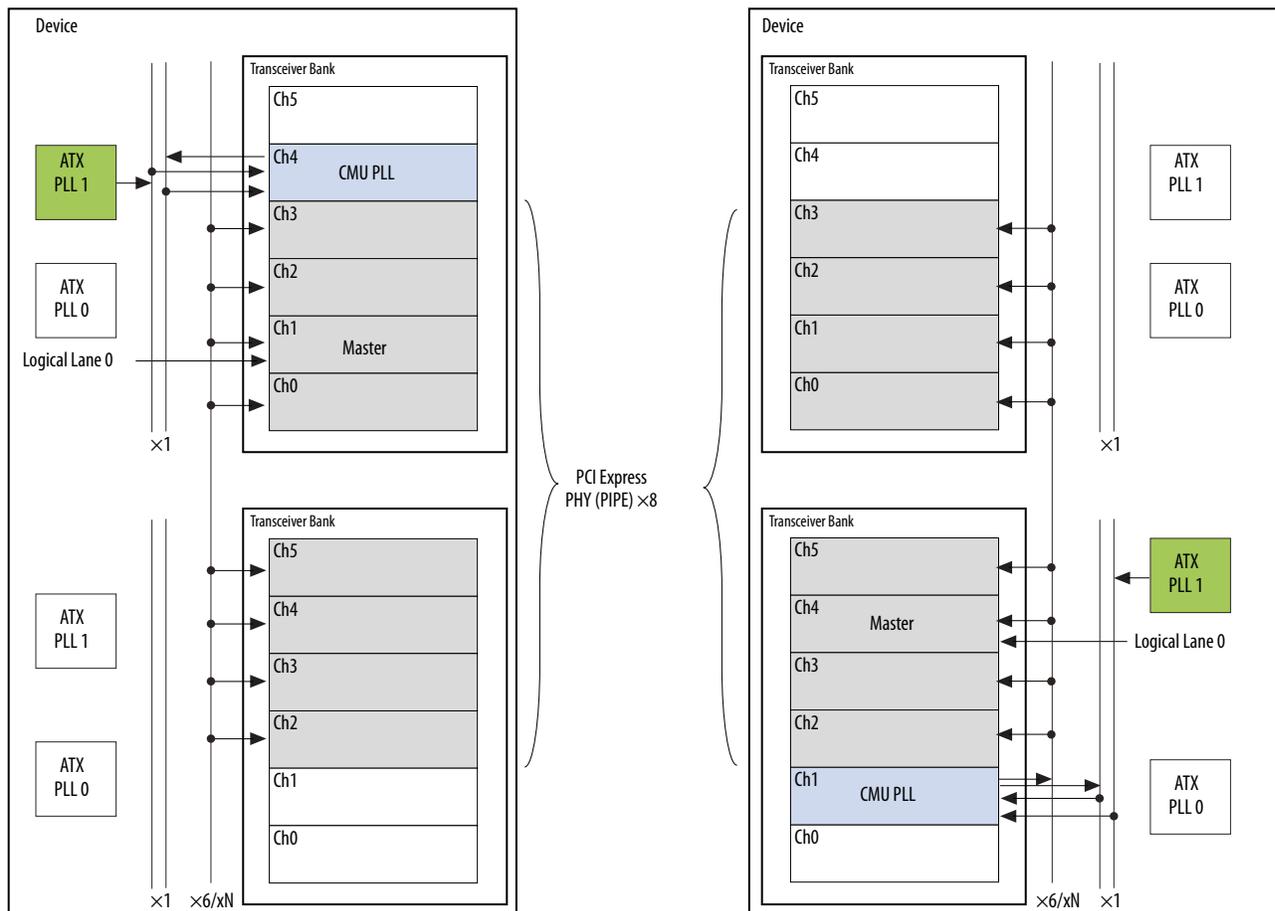
The Quartus II software automatically selects one of the following within a transceiver bank:

- 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 data rates and an ATX PLL is used for Gen3 data rates. The CMU PLL and/or ATX PLL must be within the same transceiver bank.

Figure 4-28: 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 [Stratix V Hard IP for PCI Express User Guide](#).

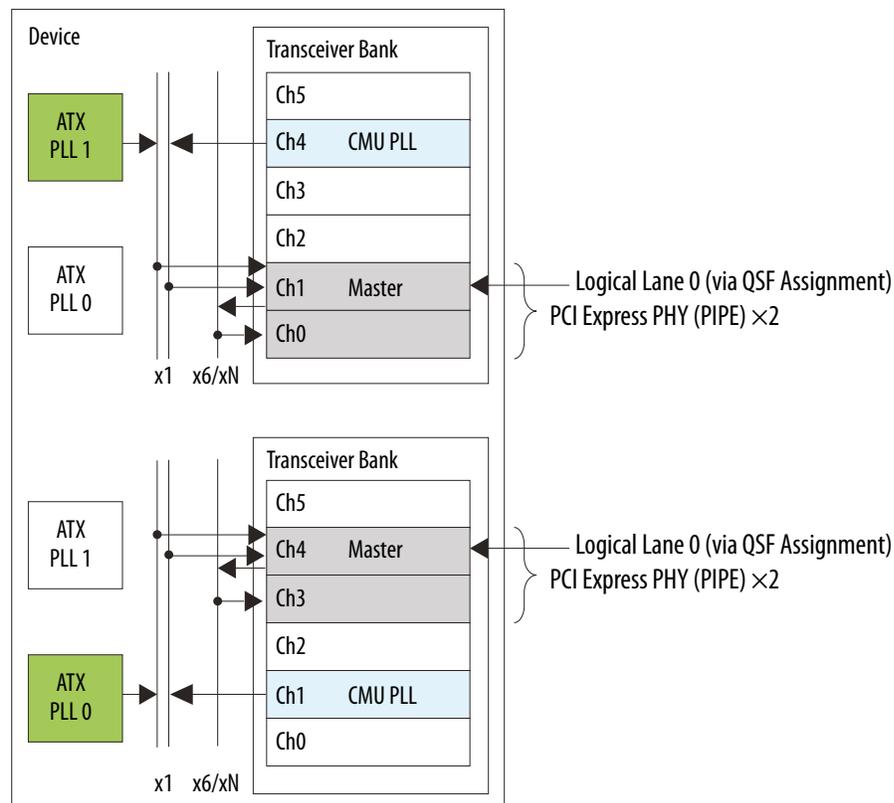
Advanced Channel Placement Guidelines for PIPE Configurations

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 data rates and the ATX PLL for Gen3 data rates 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 4-29: 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 4-30: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL in the Same Transceiver Bank

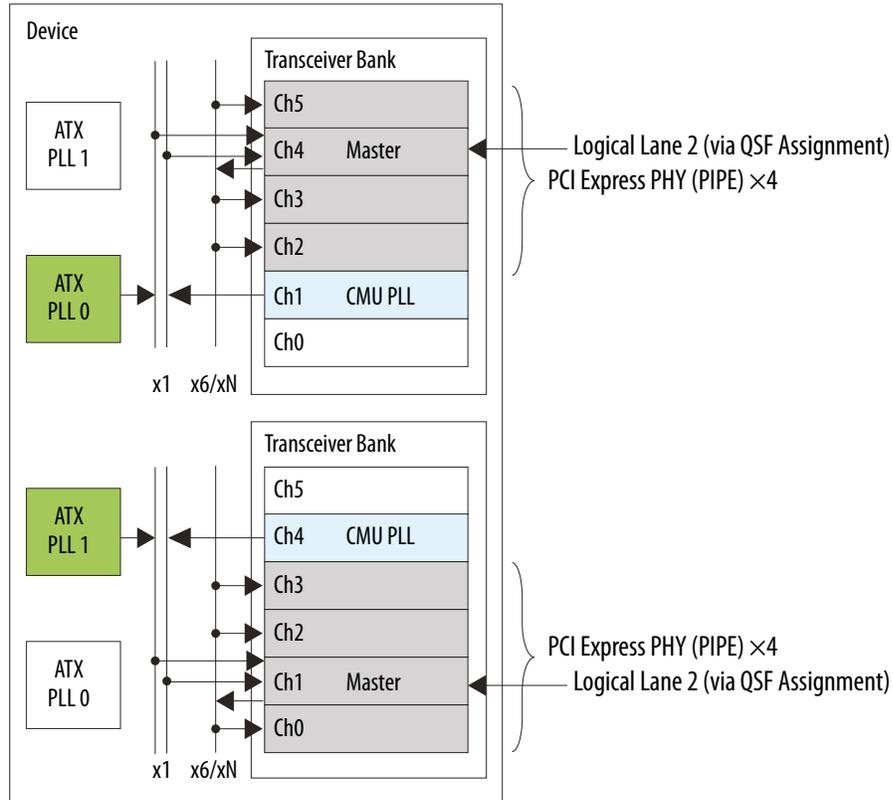


Figure 4-31: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL Across Two Transceiver Banks – example 1

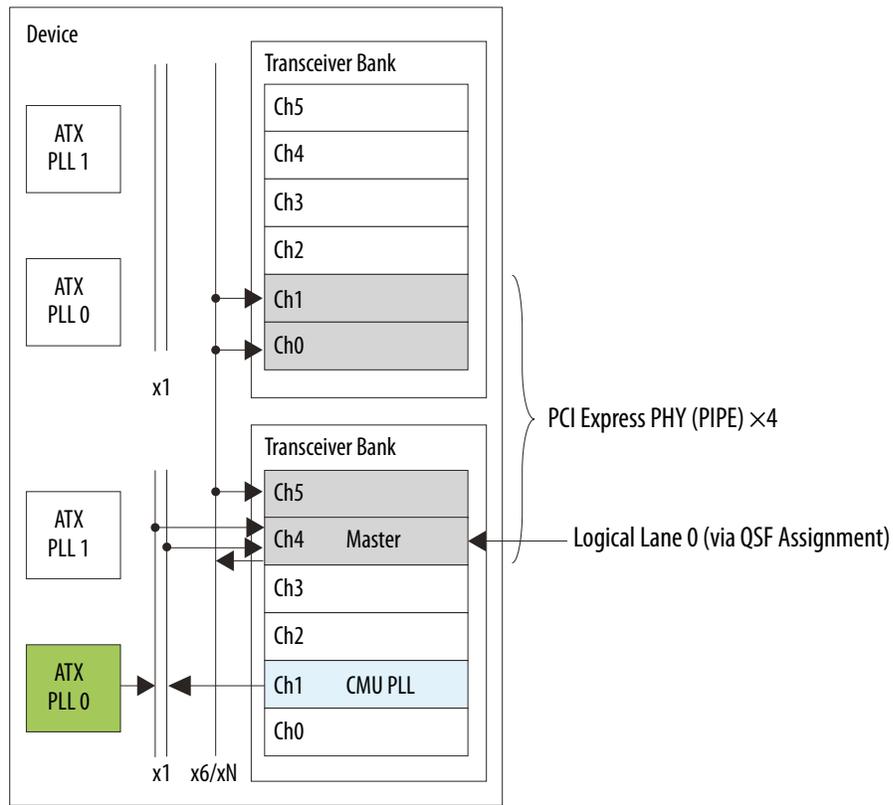
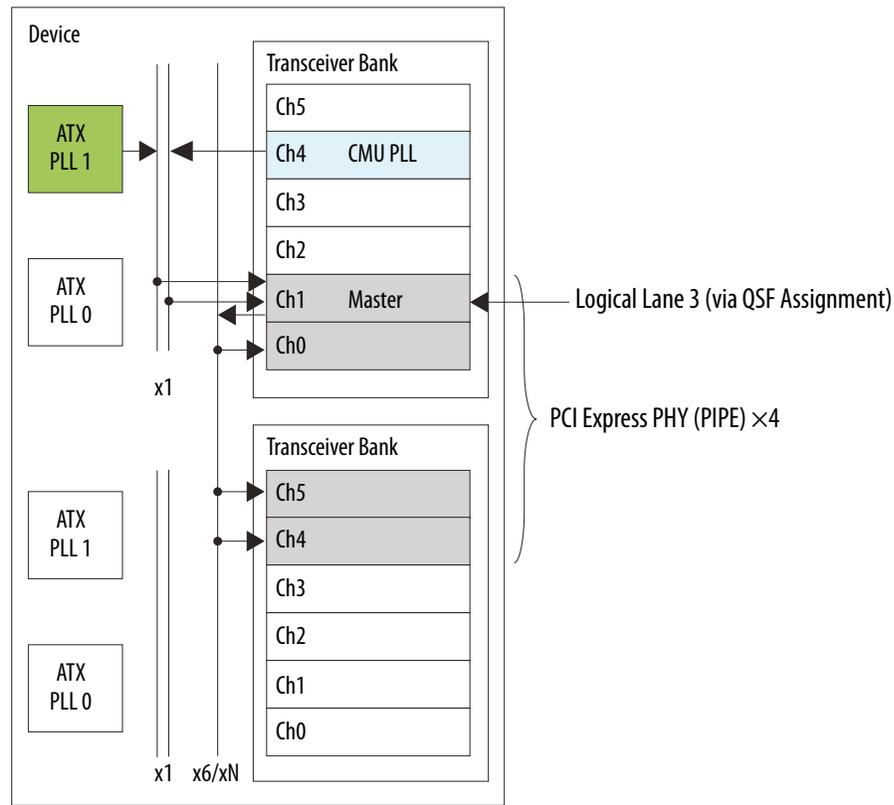


Figure 4-32: PIPE x4 Gen1, Gen2, and Gen3 Advanced Channel Placement Using CMU and/or ATX PLL Across Two Transceiver Banks – example 2



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 4-33: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement That is Compatible with HIP x8 Channel Placement

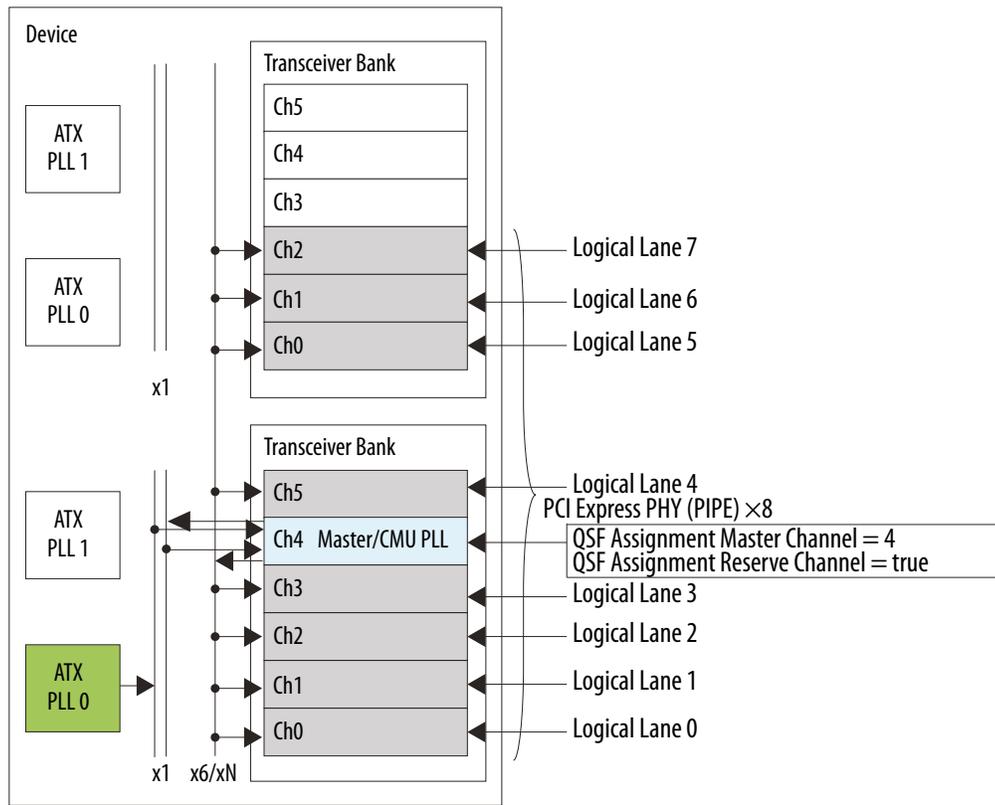
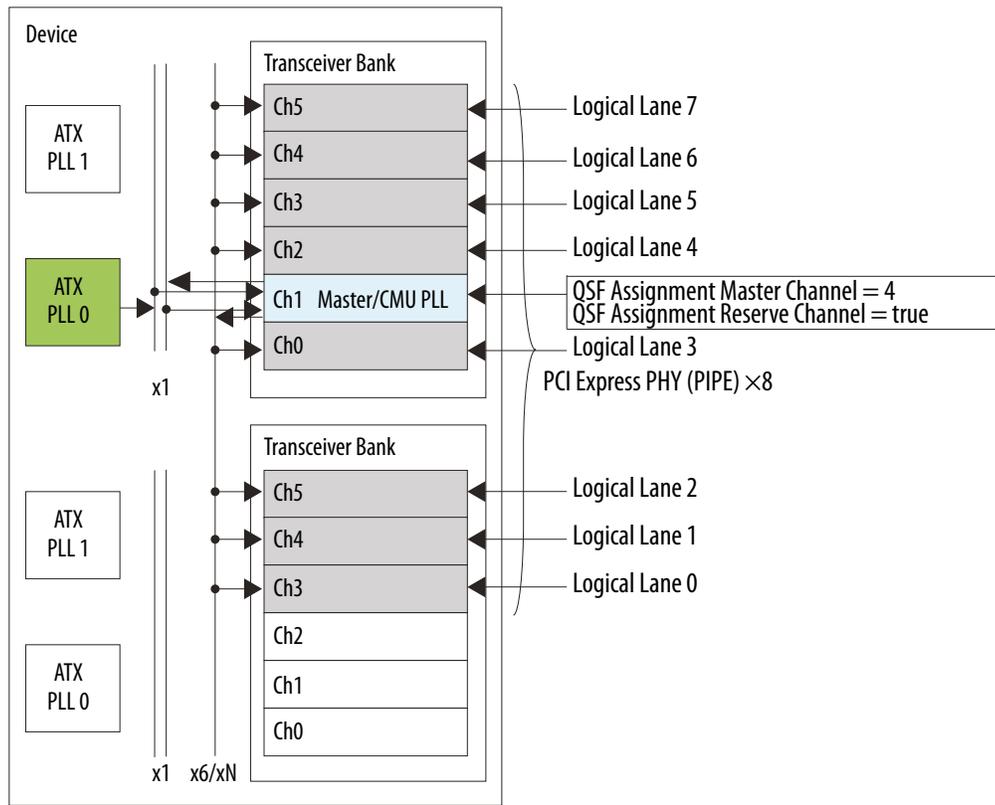


Figure 4-34: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement That is Not Compatible with HIP x8 Channel Placement



The following figures show PIPE x8 Gen1, Gen2, and Gen3 advanced channel placement that requires only a master channel QSF assignment.

Figure 4-35: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 1

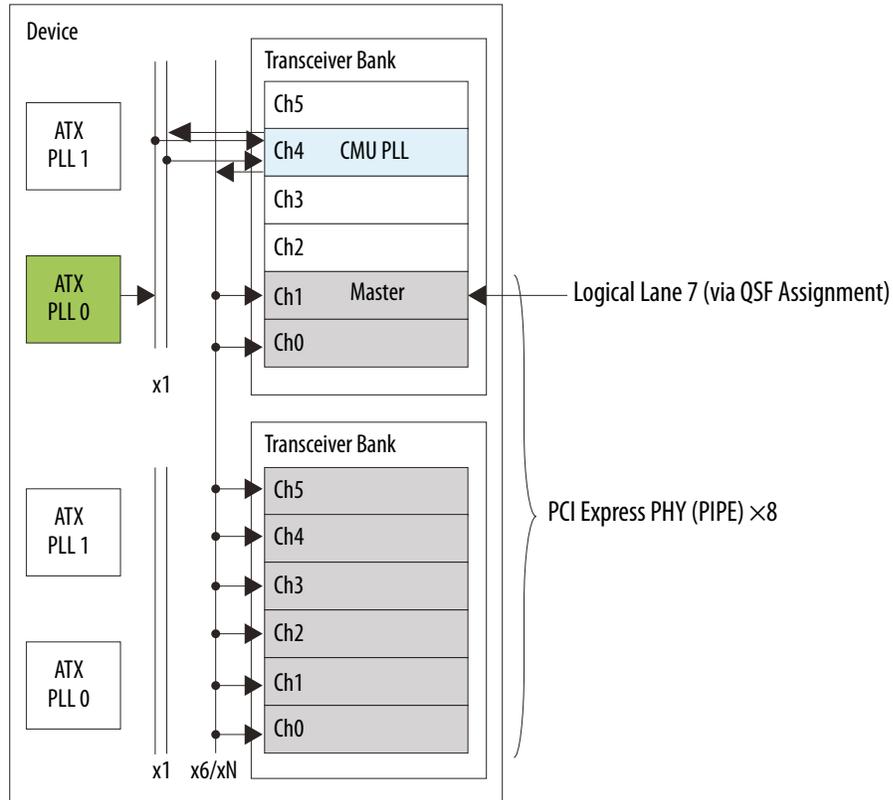


Figure 4-36: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 2

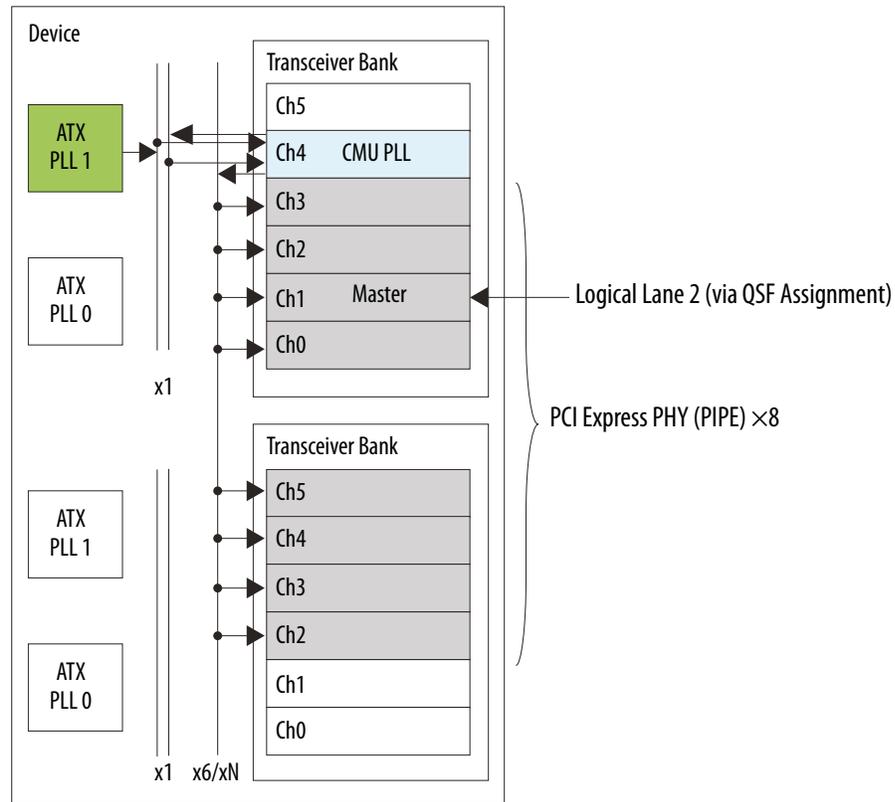
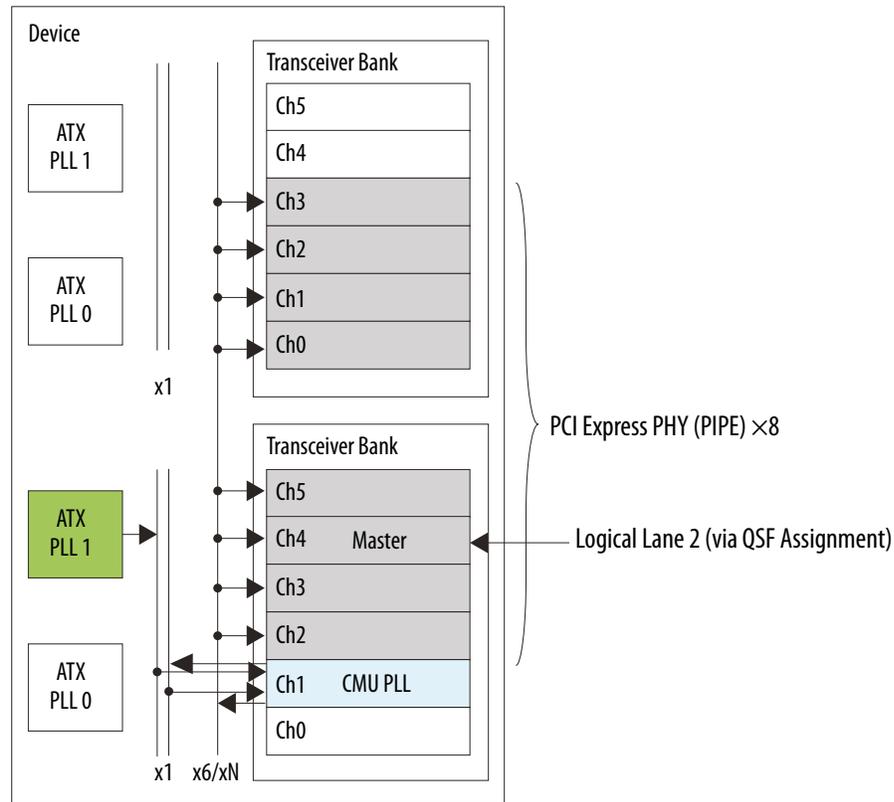


Figure 4-37: PIPE x8 Gen1, Gen2, and Gen3 Advanced Channel Placement – example 3



Transceiver Clocking for PCIe Gen3

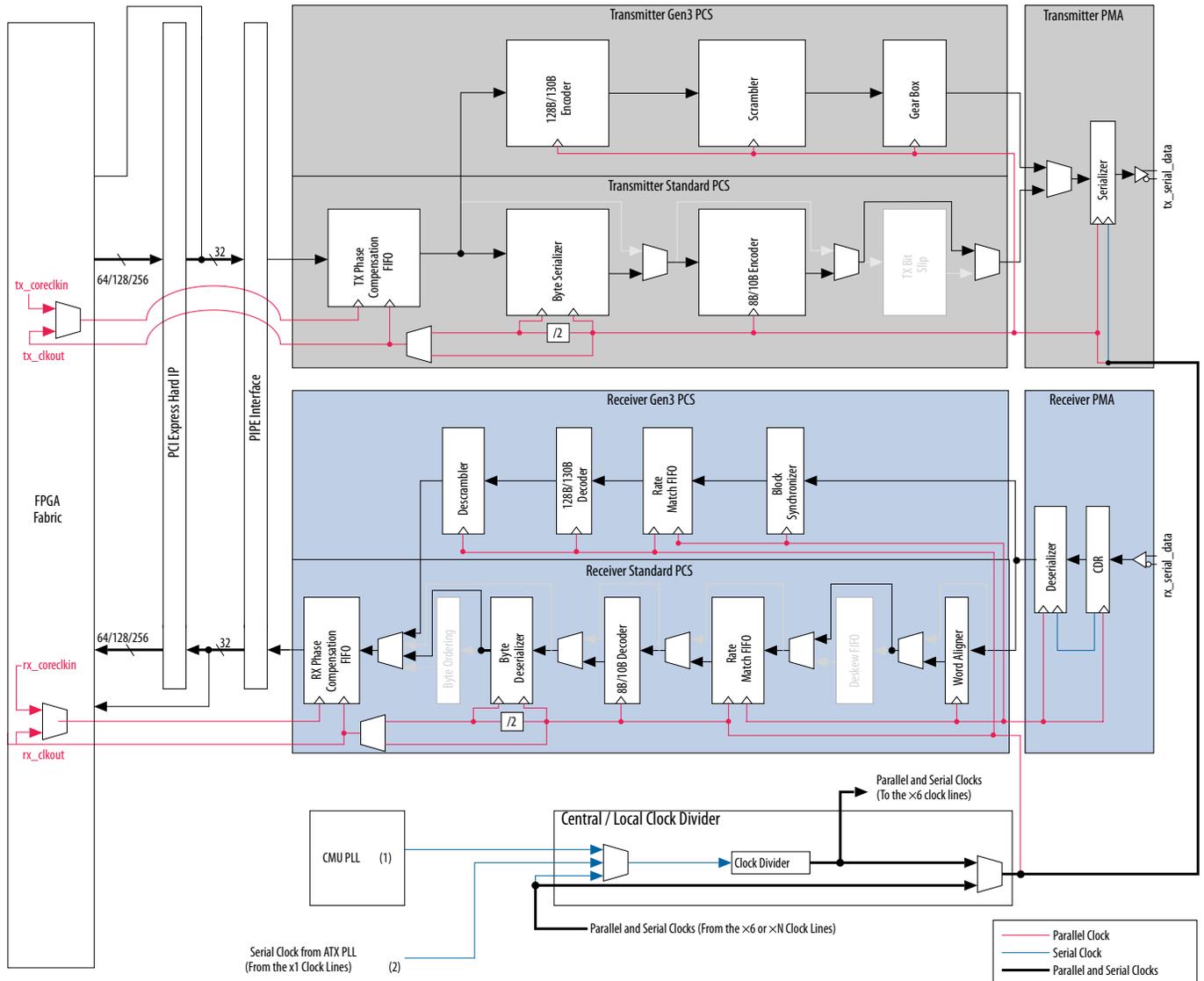
This section describes the transceiver clocking topology for both the PCIe Gen3 Hard IP and PIPE configuration.

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 4-38: 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.

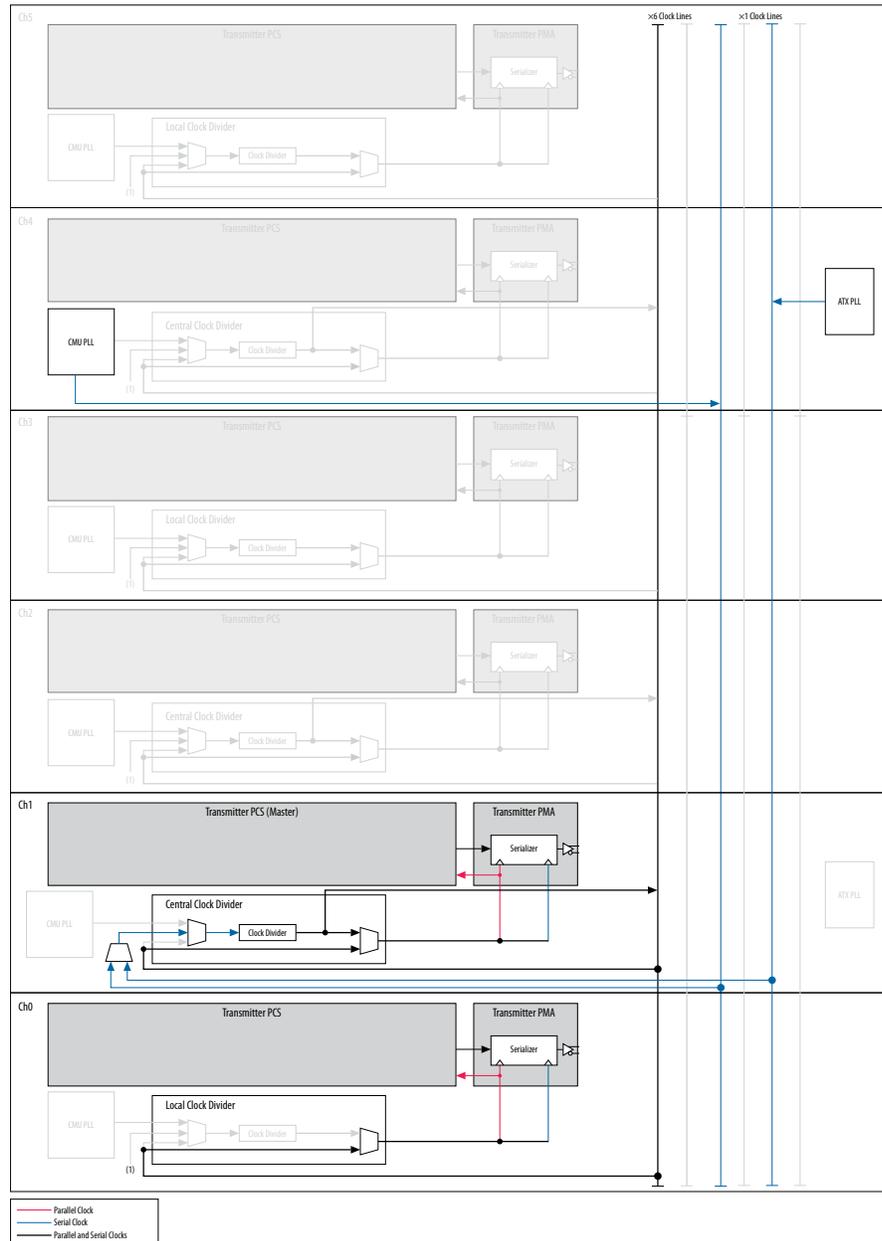
For **PCIe x1 Gen3** using PIPE configuration, the CMU PLL (transceiver physical channel 1 or 4) and the top or bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock for the transmitter datapath clock and the rate matcher side of the FIFO in the receiver datapath 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 4-39: 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 4-40: 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.

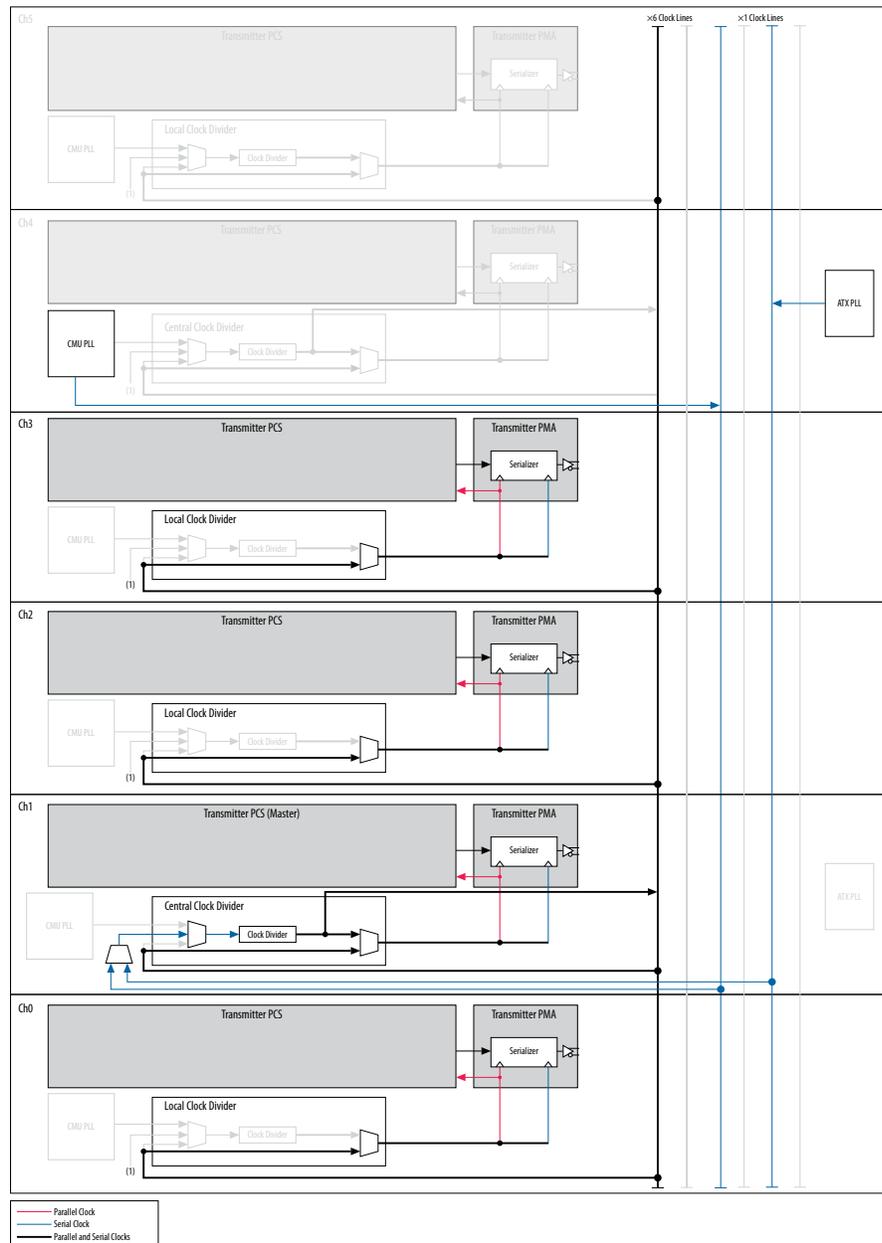
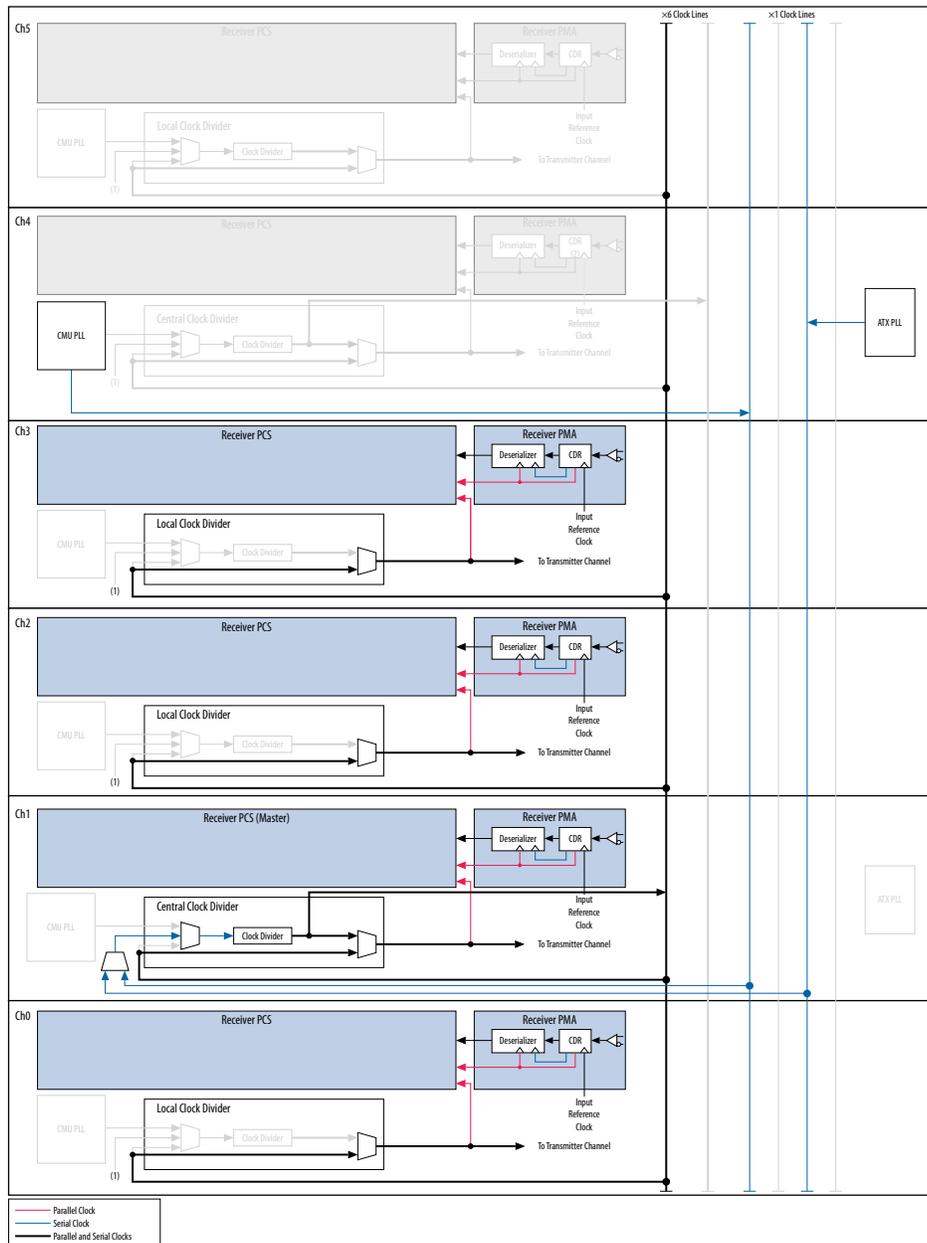


Figure 4-41: Receiver Clocking in a Gen1/Gen2/Gen3 PCIe x4 Hard IP and PIPE Configuration



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

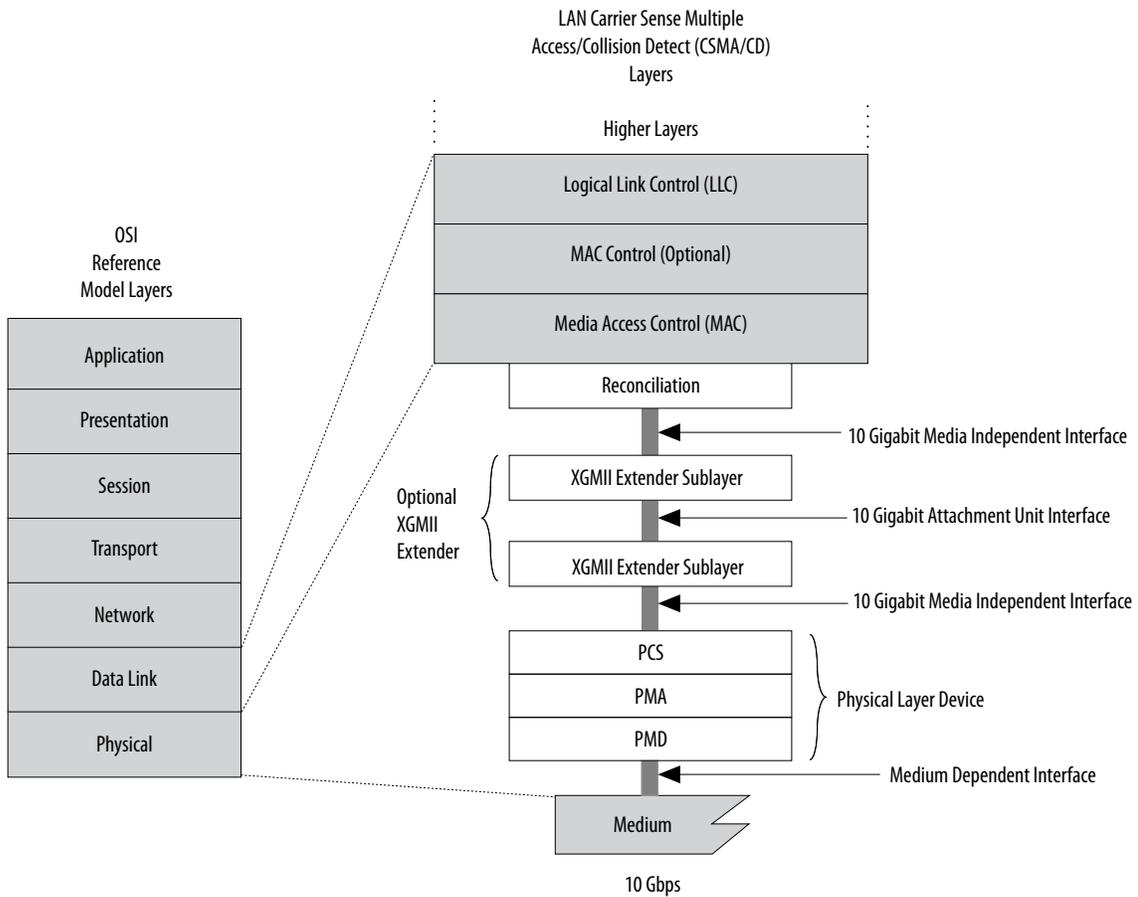
For **PCIe x8 Gen3**, the CMU PLL (transceiver physical channel 4) and the top or bottom ATX PLL of the lower transceiver bank are configured to generate the high-speed serial clock. A total of nine transceiver channels are required to implement PCIe x8 Gen3, including eight data channels and one channel 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 4-42: XAUI and XGMII Layers



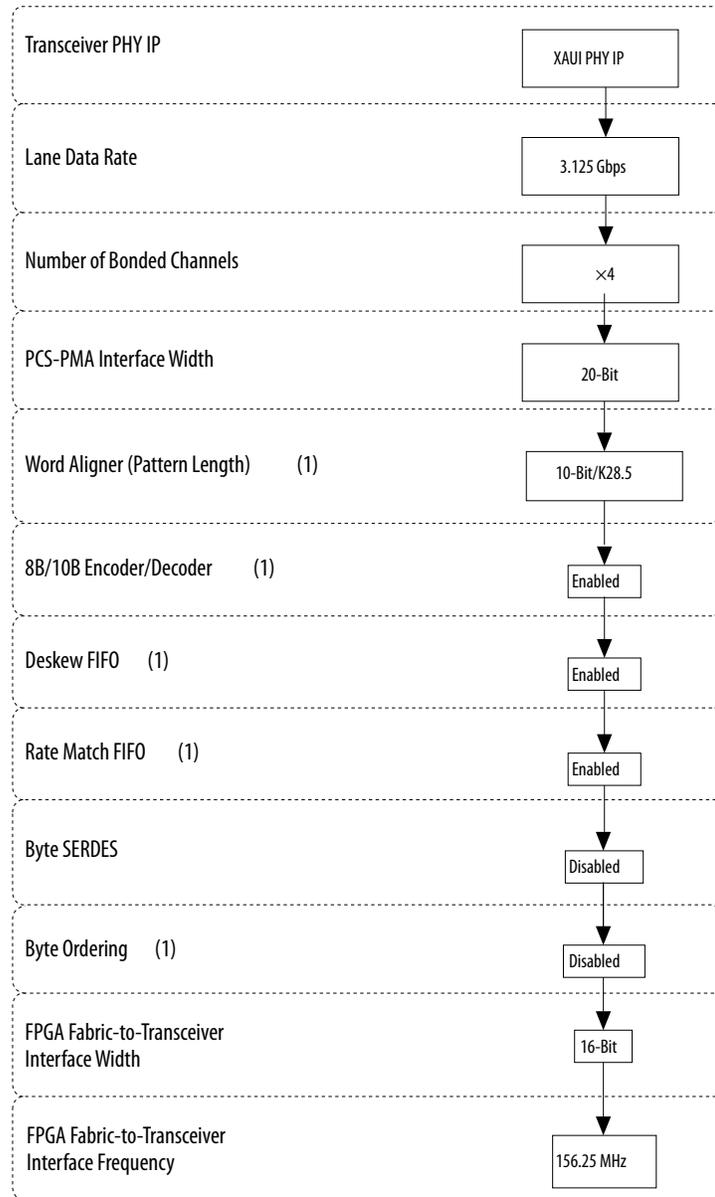
Related Information

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.

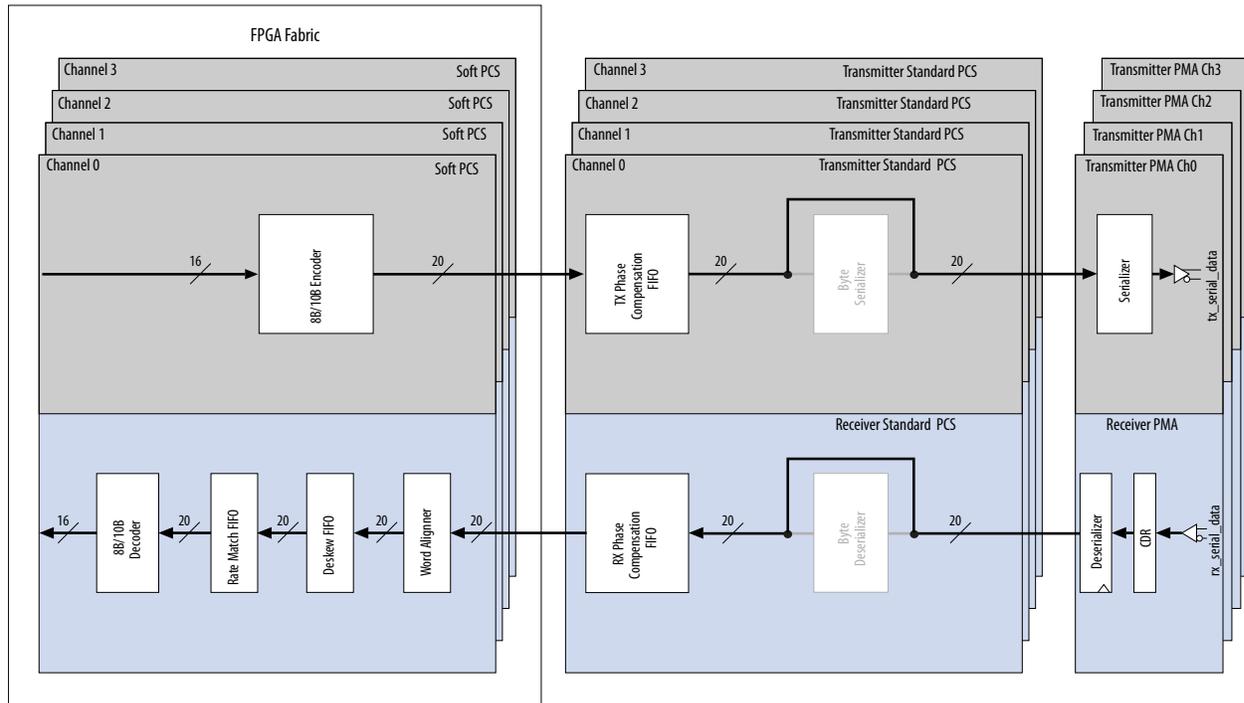
Figure 4-43: XAUI Datapath Configuration



(1) Implemented in soft logic.

Figure 4-44: 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.



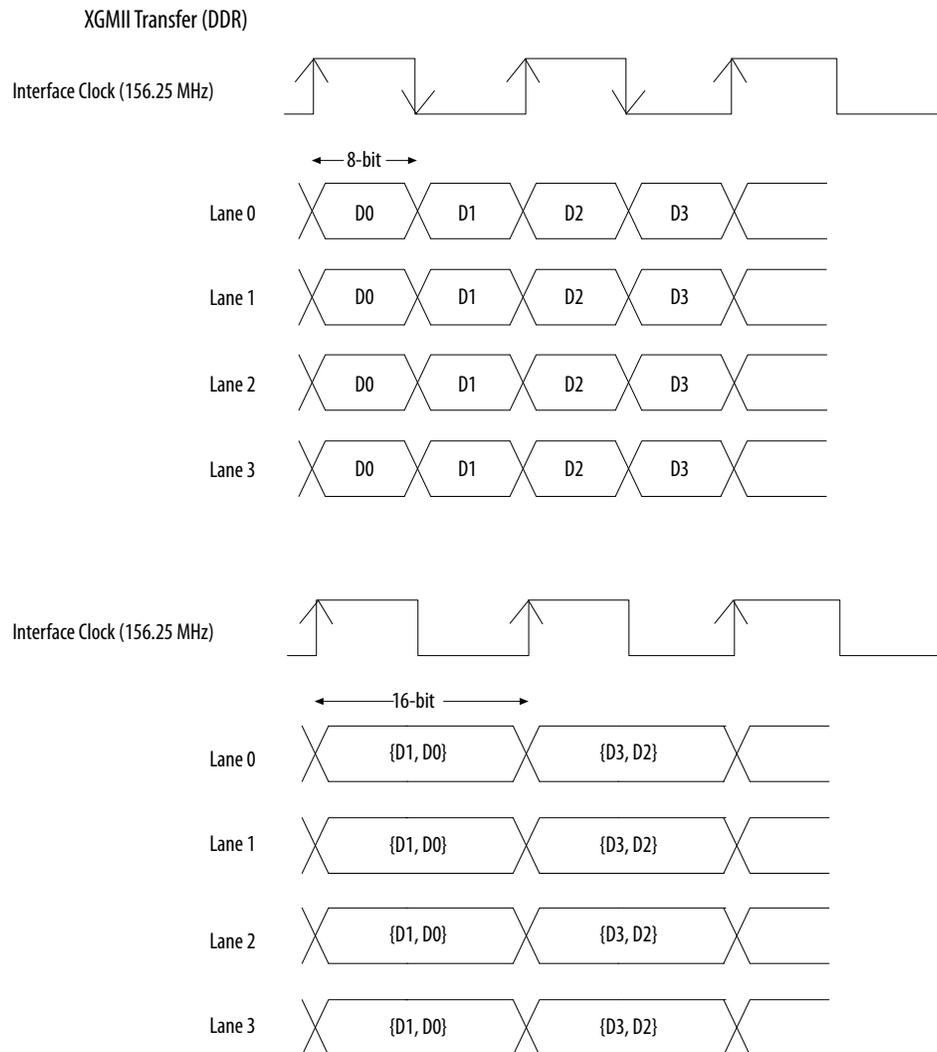
Supported Features

Stratix V 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.

Stratix V 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 4-45: Implementation of the XGMII Specification in Stratix V 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 Stratix V 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 4-46: 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.

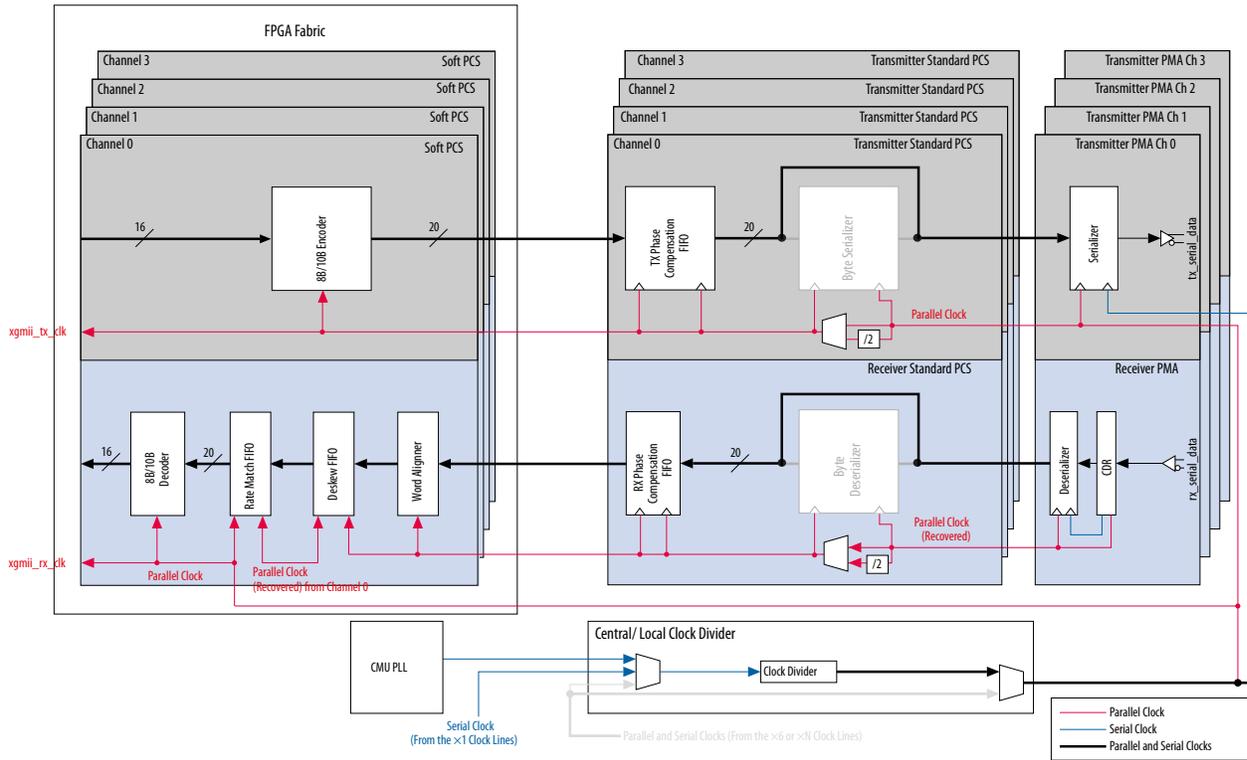


Table 4-9: Input Reference Clock Frequency and Interface Speed Specifications for XAUI Configurations

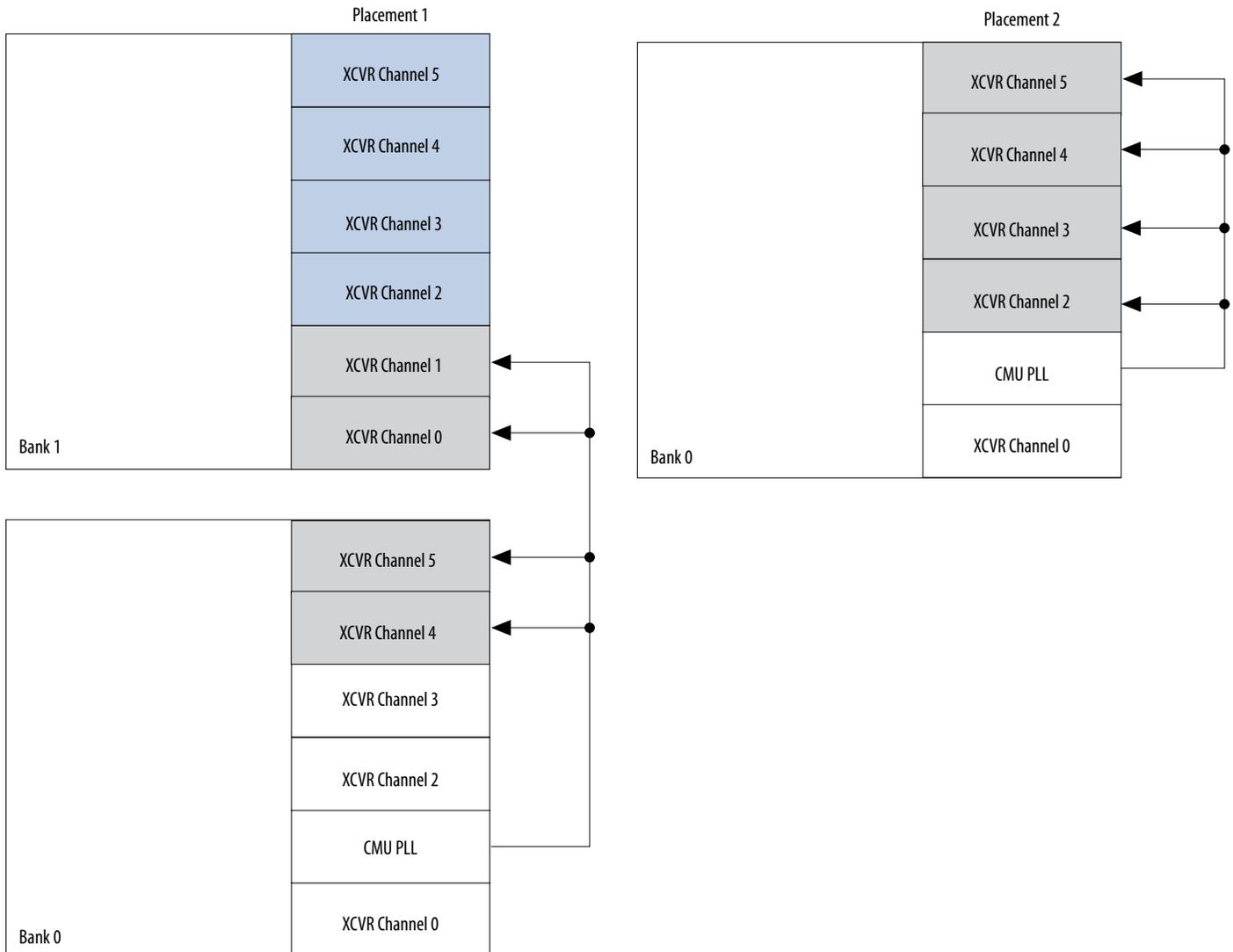
Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
156.25	16-bit data, 2-bit control	156.25

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 4-47: 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](#).

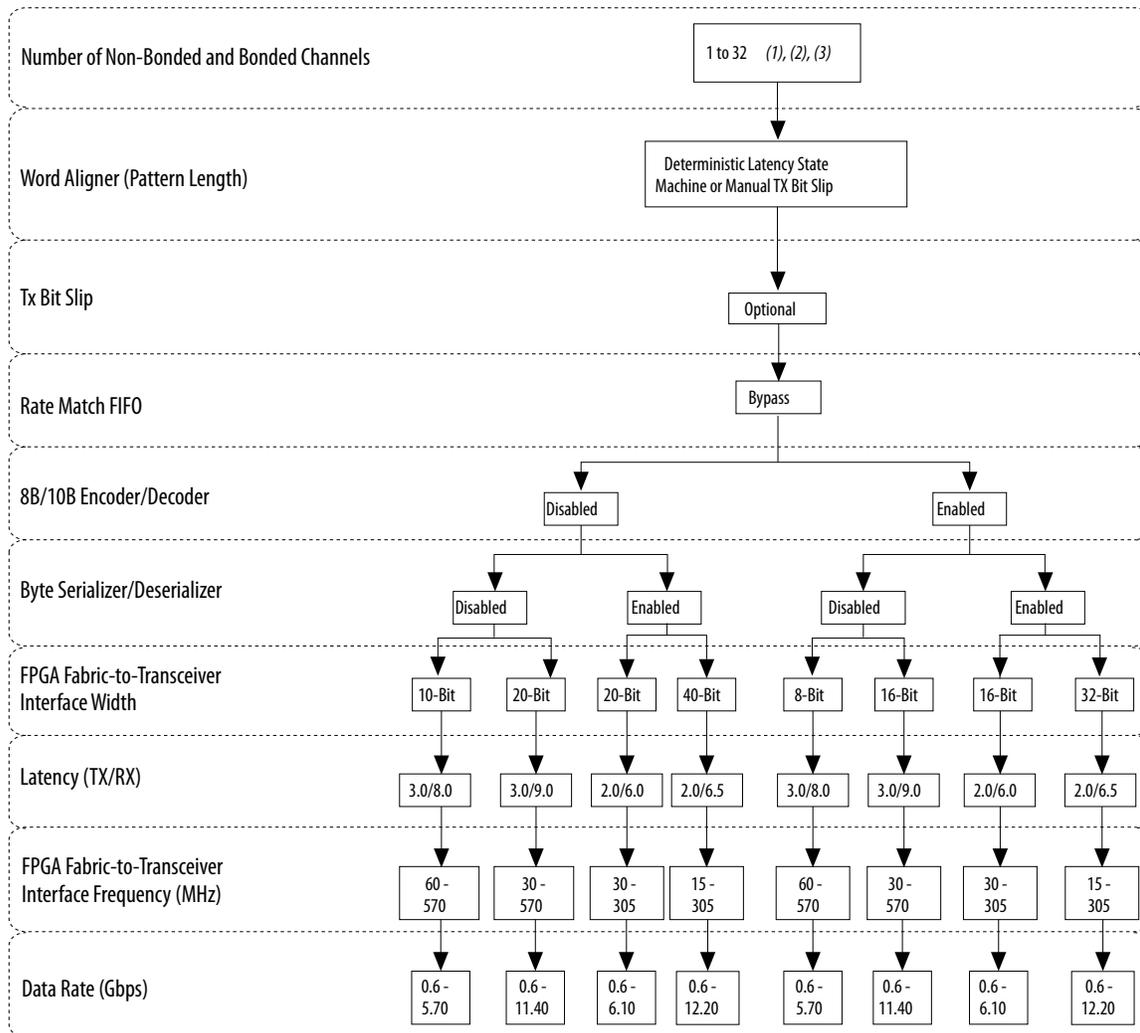
CPRI and OBSAI—Deterministic Latency Protocols

Stratix V 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

Stratix V devices have a number of options available for the deterministic latency datapath configuration.

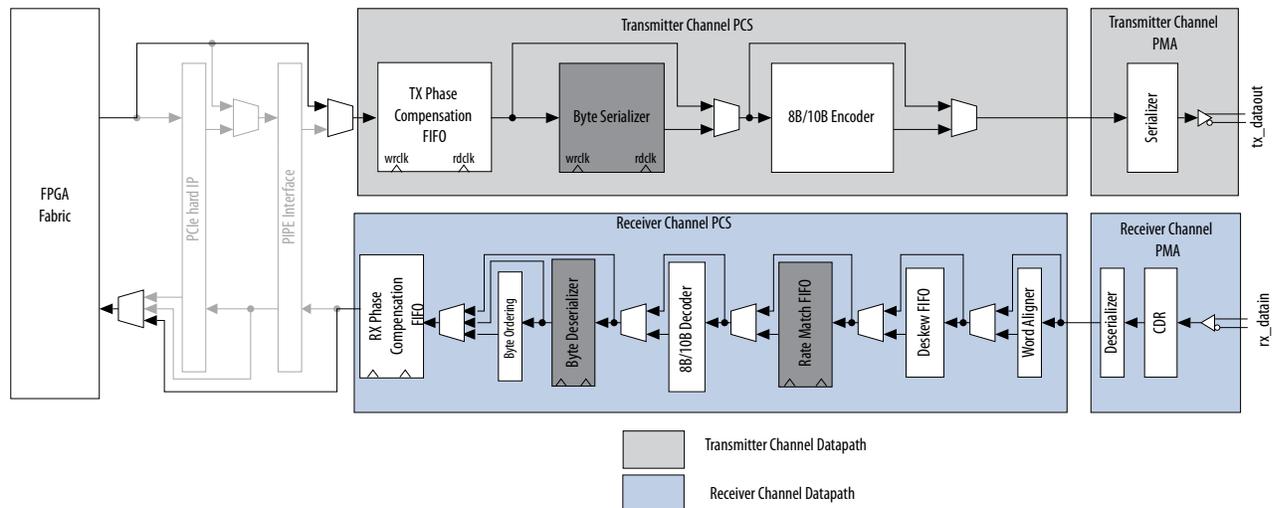
Figure 4-48: Deterministic Latency Datapath Configuration



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 4-49: 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.

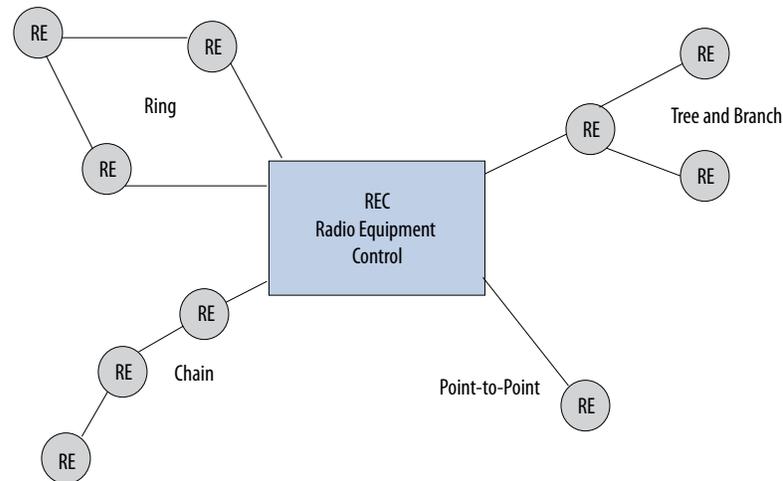
CPRI and OBSAI

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 4-50: 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.

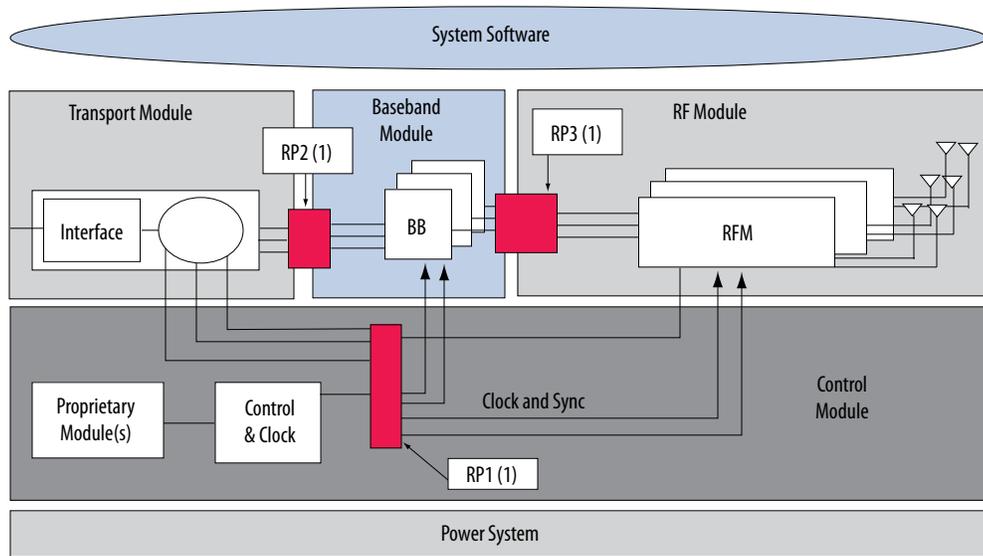
OBSAI was established by several OEMs to develop a set of specifications that can be used for configuring and connecting common modules into base transceiver stations (BTS).

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 4-51: Example of the OBSAI BTS Architecture



(1) RP = Reference Point

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	—	—
1228.8	Yes	Yes	Yes	Yes
2457.6	—	Yes	Yes	Yes
3072	—	Yes	Yes	Yes
4915.2	—	—	—	Yes
6144	—	—	—	Yes
9800 ⁽¹³⁾	—	—	—	Yes

Related Information

For more information, refer to the [Deterministic Latency PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide](#).

⁽¹³⁾ Applicable to C1, C2, C2L, C3, I2, I2L, and I3L with -1 and -2 transceiver speed grades only.

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 4-52: Deterministic Latency State Machine in the Word Aligner

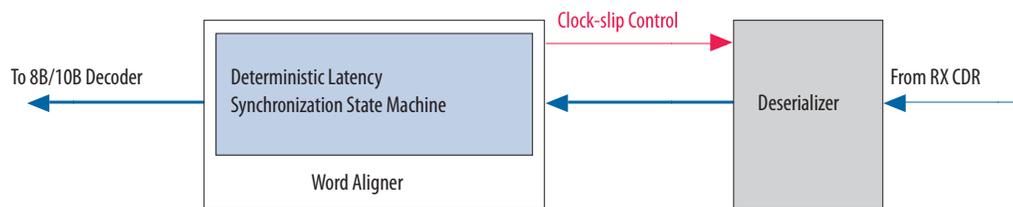


Table 4-11: Methods to Achieve Deterministic Latency Mode in Stratix V Devices

Existing Feature		Enhanced Feature	
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](#)

Transceiver Configurations

Stratix V 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.

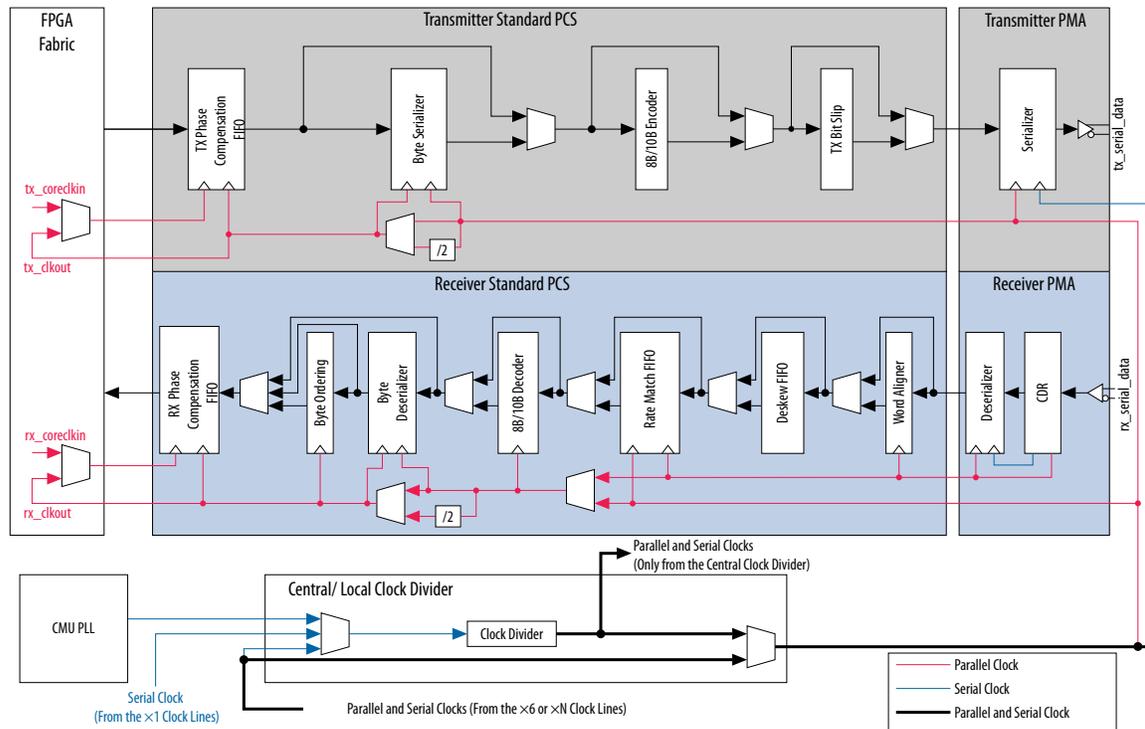
Standard PCS Configurations—Custom Datapath

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 4-53: 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.

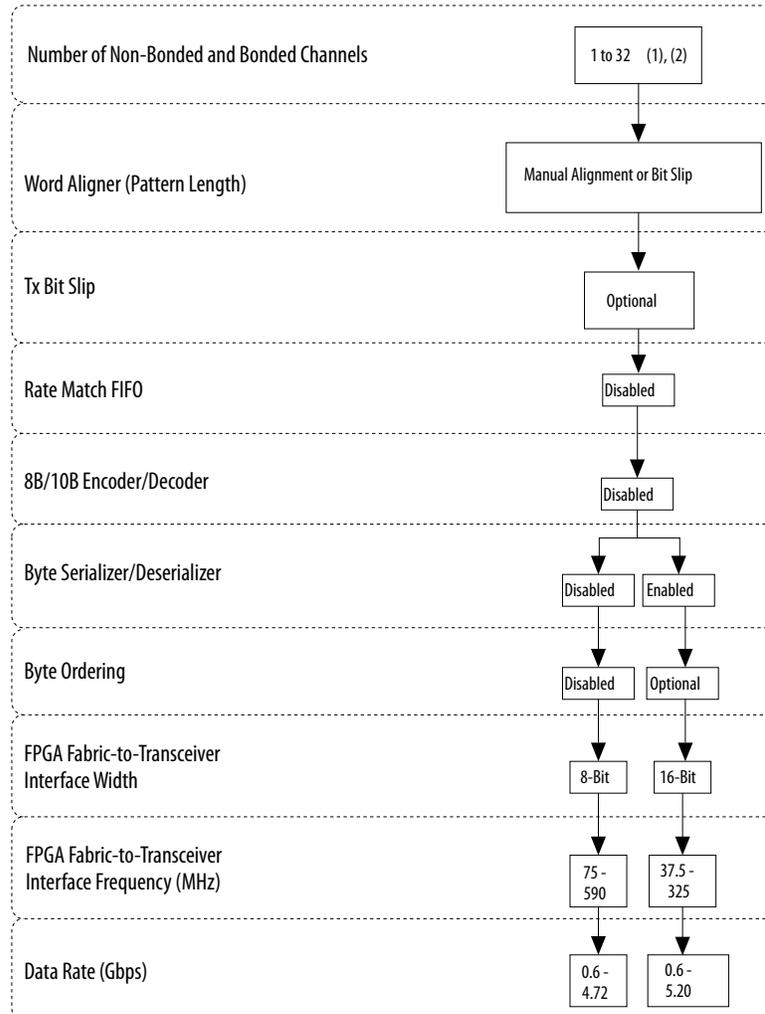
Table 4-12: PCS-PMA Interface Widths and Supported Data Rates

PCS-PMA Interface Width	Supported Data Rate Range PMA
Custom 8-bit width	600 Mbps to 5.20 Gbps
Custom 10-bit width	600 Mbps to 6.50 Gbps
Custom 16-bit width	600 Mbps to 9.76 Gbps

PCS-PMA Interface Width	Supported Data Rate Range PMA
Custom 20-bit width	600 Mbps to 12.20 Gbps

Figure 4-54: Standard PCS Custom 8-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS custom 8-bit PMA-PCS interface width. The maximum frequencies are for the fastest devices.

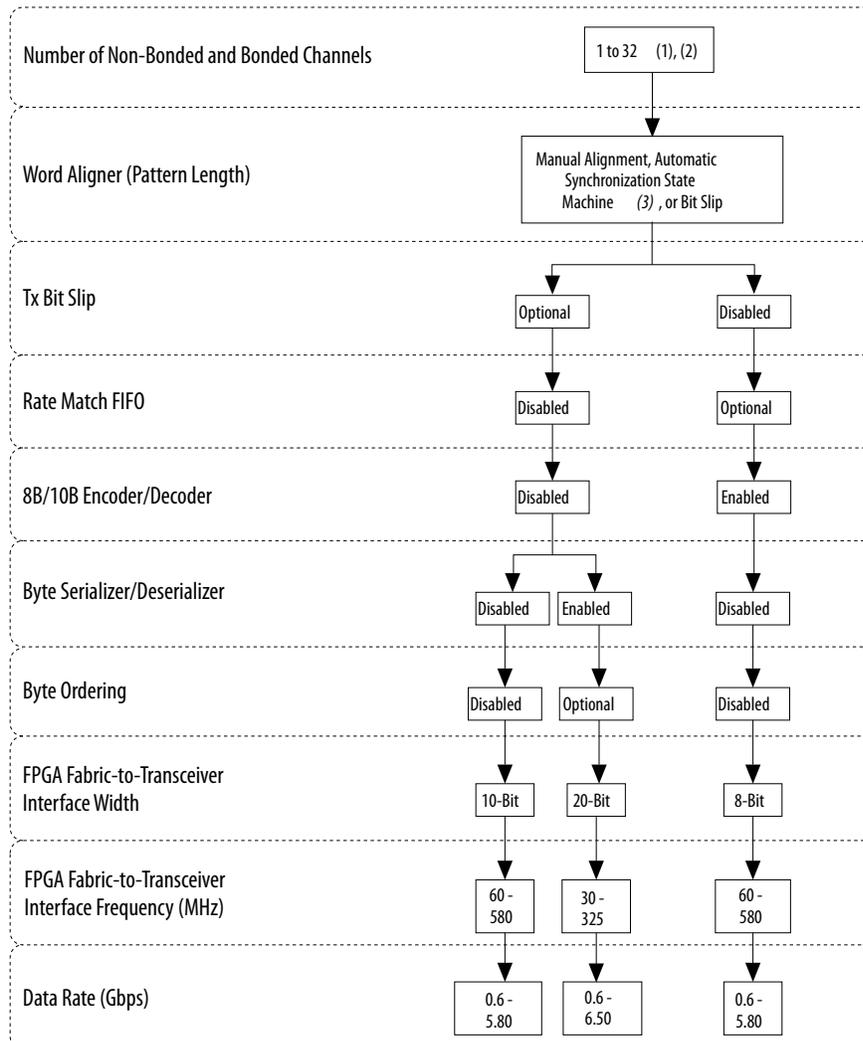


Notes:

- (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.

Figure 4-55: Standard PCS Custom 10-Bit PMA-PCS Interface Width

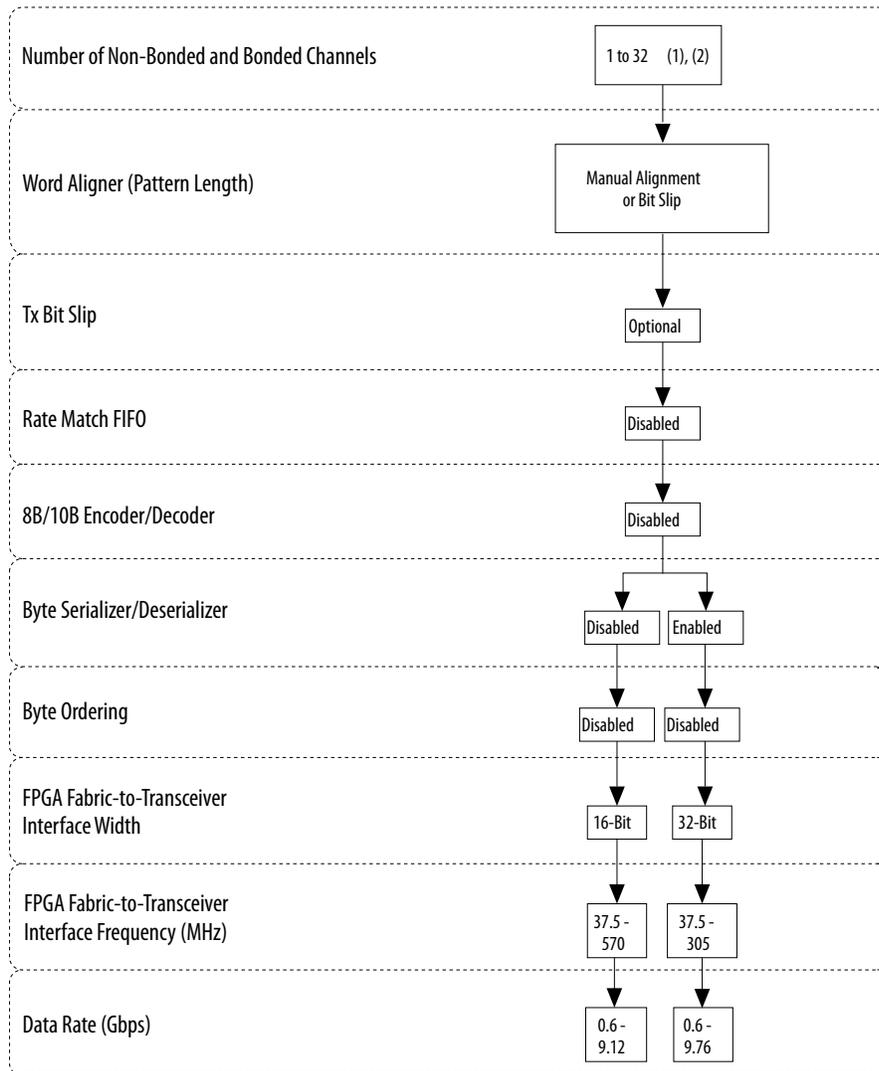
Shows the available options for the standard PCS custom 10-bit PMA-PCS interface width. The maximum frequencies are for the fastest devices.

**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) Automatic Synchronization State Machine requires enabling the 8B/10B Encoder/Decoder.

Figure 4-56: Standard PCS Custom 16-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS custom 16-bit PMA-PCS interface width. The maximum frequencies are for the fastest devices.

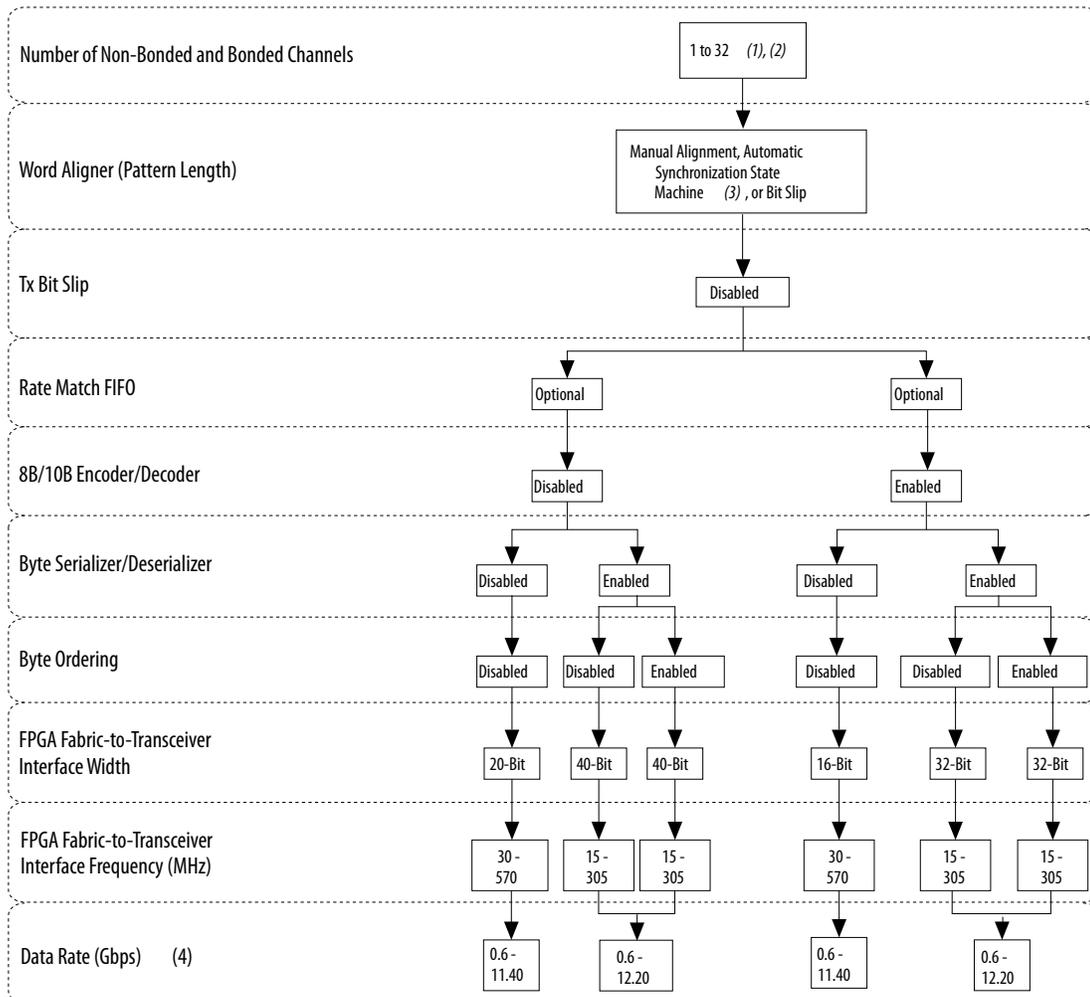


Notes:

- (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.

Figure 4-57: Standard PCS Custom 20-Bit PMA-PCS Interface Width

Shows the available options for the standard PCS custom 20-bit PMA-PCS interface width. 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) 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 “Standard PCS Architecture” section in the Transceiver Architecture in Stratix V Devices](#)
- [For information about the maximum data rate for a certain speed grade, refer to the Stratix 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

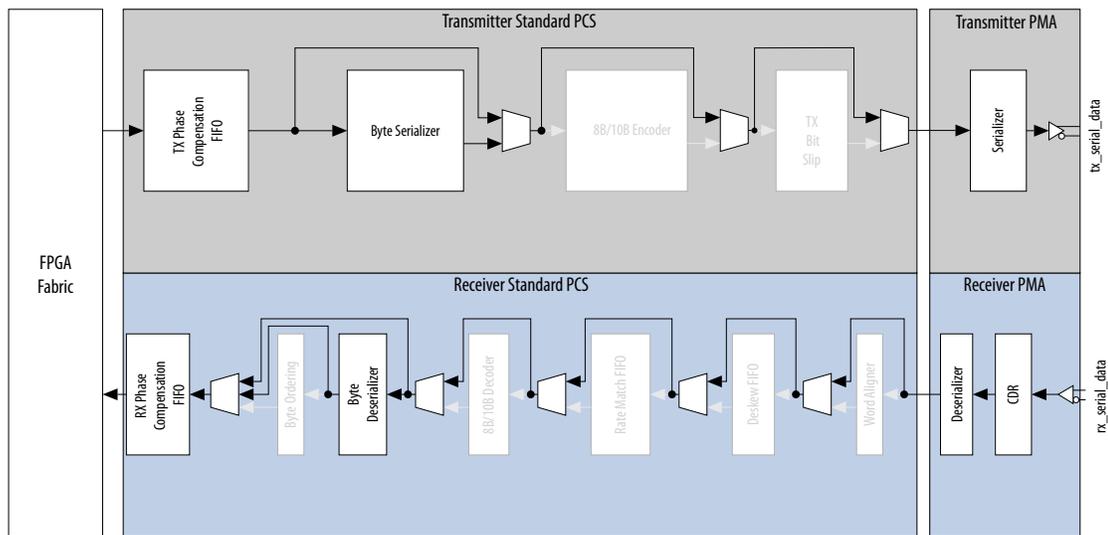
A low latency datapath bypasses much of the standard PCS, allowing more design control in the FPGA fabric. Use the Low Latency PHY IP to enable the standard PCS in a low latency datapath.

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 4-58: 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.

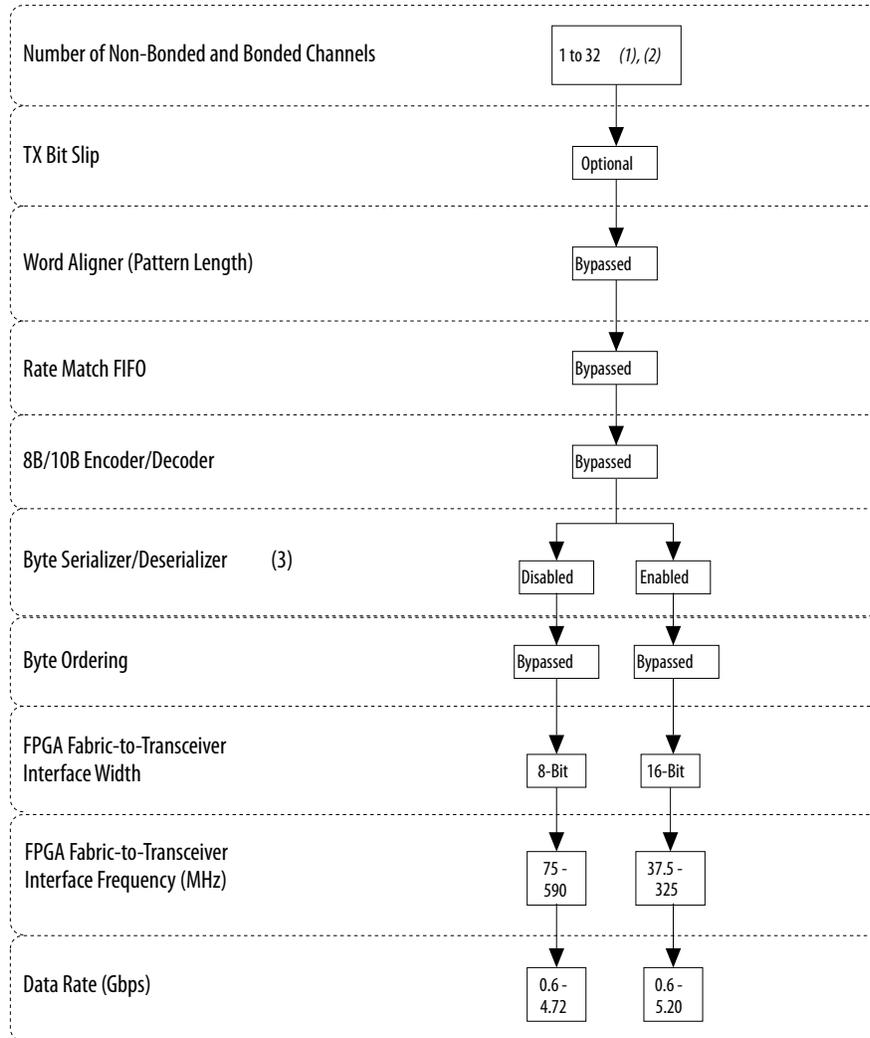
Table 4-13: PCS-PMA Interface Widths and Data Rates

Low Latency PHY IP Core	Supported Data Rate Range PMA
Low Latency 8-bit width	600 Mbps to 5.20 Gbps
Low Latency 10-bit width	600 Mbps to 6.50 Gbps
Low Latency 16-bit width	600 Mbps to 9.76 Gbps
Low Latency 20-bit width	600 Mbps to 12.20 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 4-59: 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.

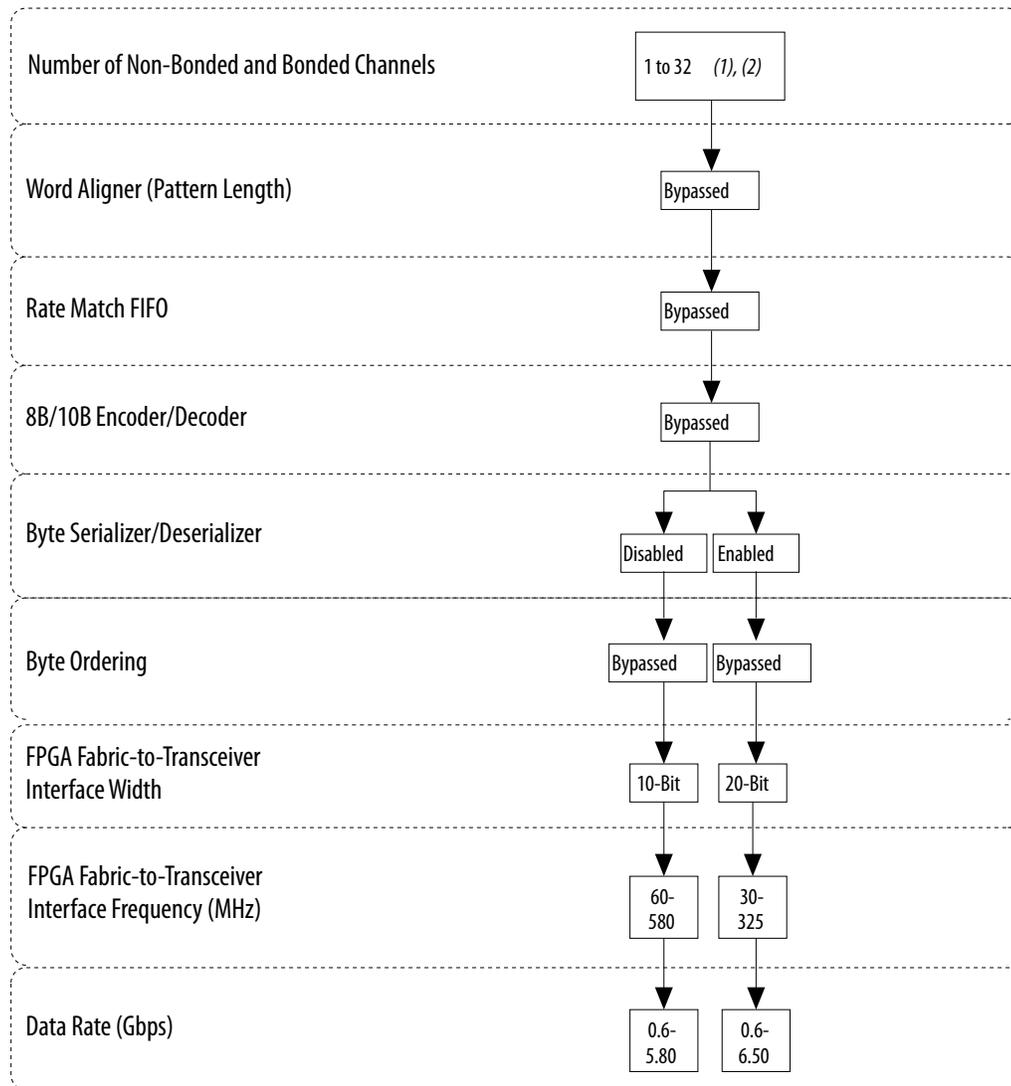


Notes:

- (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 4-60: 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.

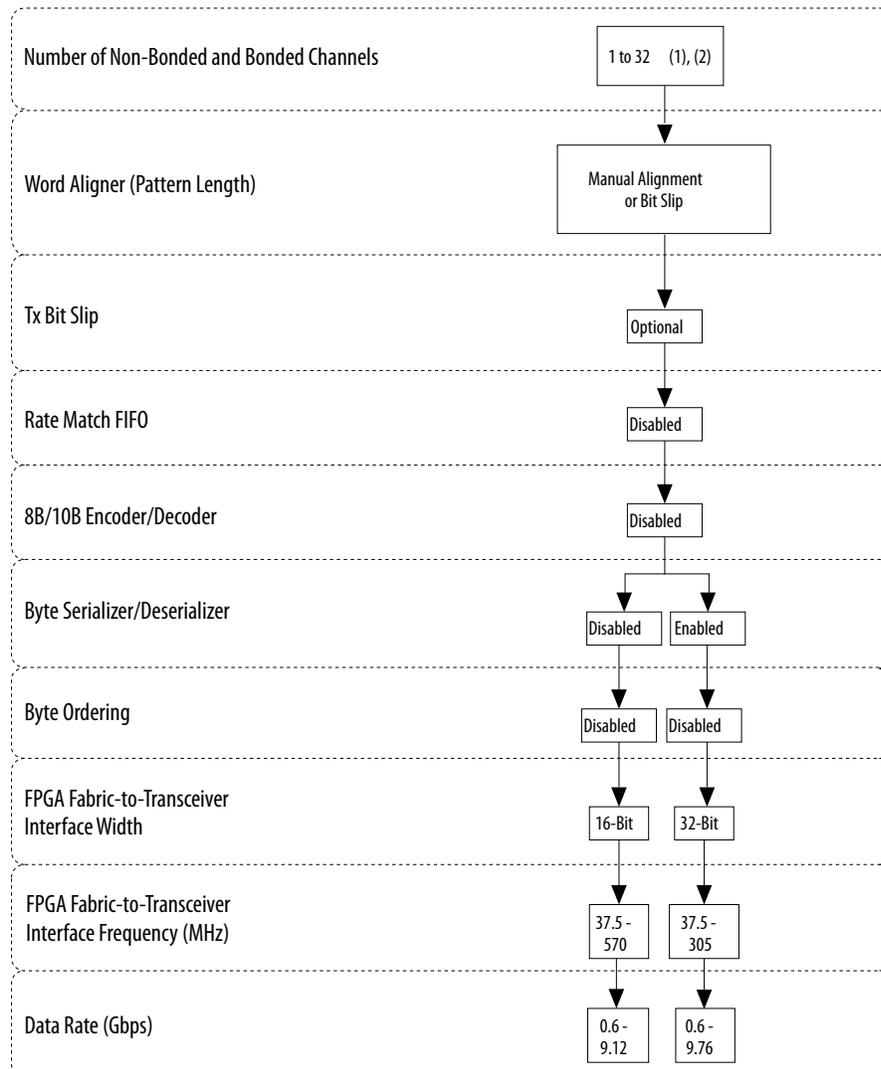


Notes:

- (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 4-61: Standard PCS Low Latency 16-Bit PMA-PCS Interface Width

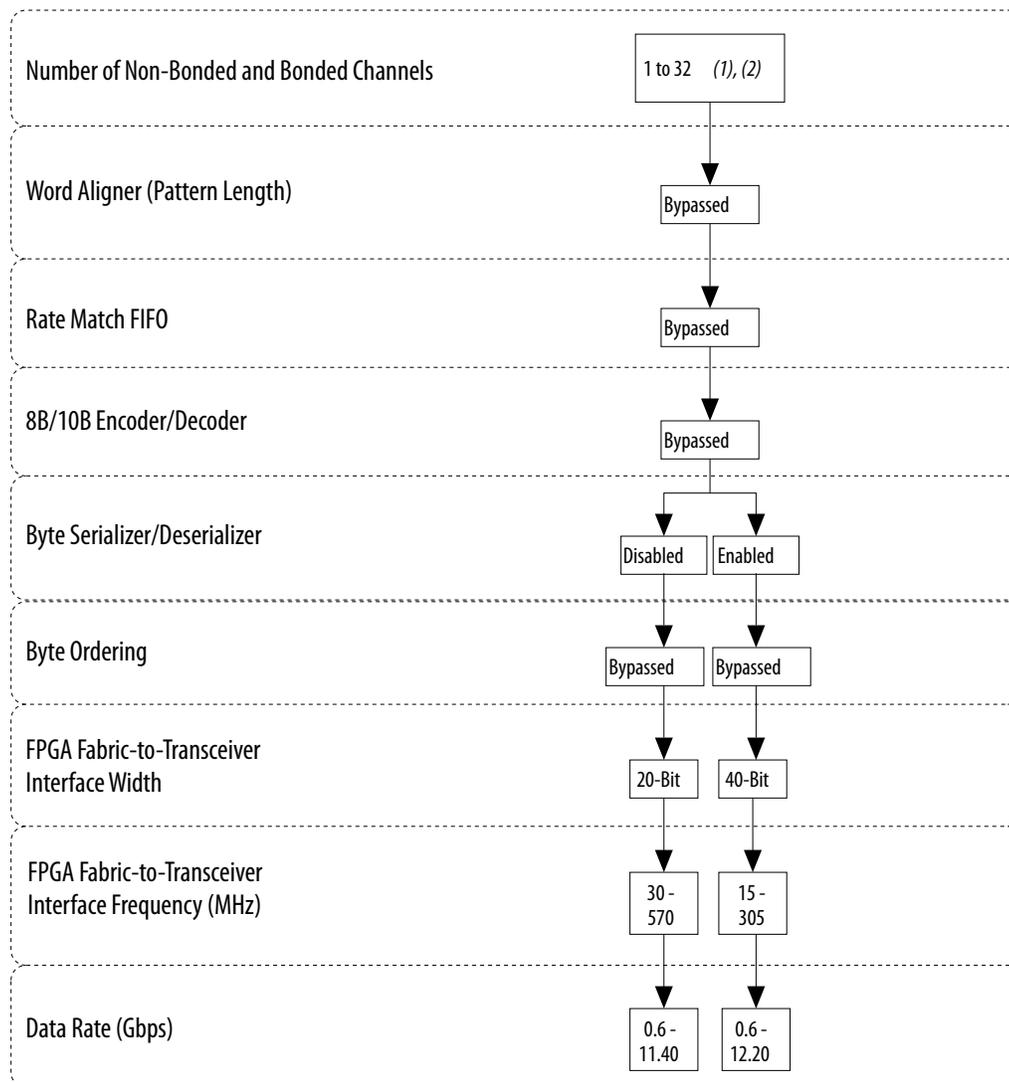
Shows the available options for the standard PCS low latency 16-bit PMA-PCS interface width. The blocks shown as “Disabled” are not used but incur latency. The blocks shown as “Bypassed” are not used and do not incur any latency. The maximum frequencies are for the fastest devices.

**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 4-62: 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.

**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 “Standard PCS Architecture” section in the Transceiver Architecture in Stratix V Devices](#)
- [For information about the maximum data rate for a certain speed grade, refer to the Stratix 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

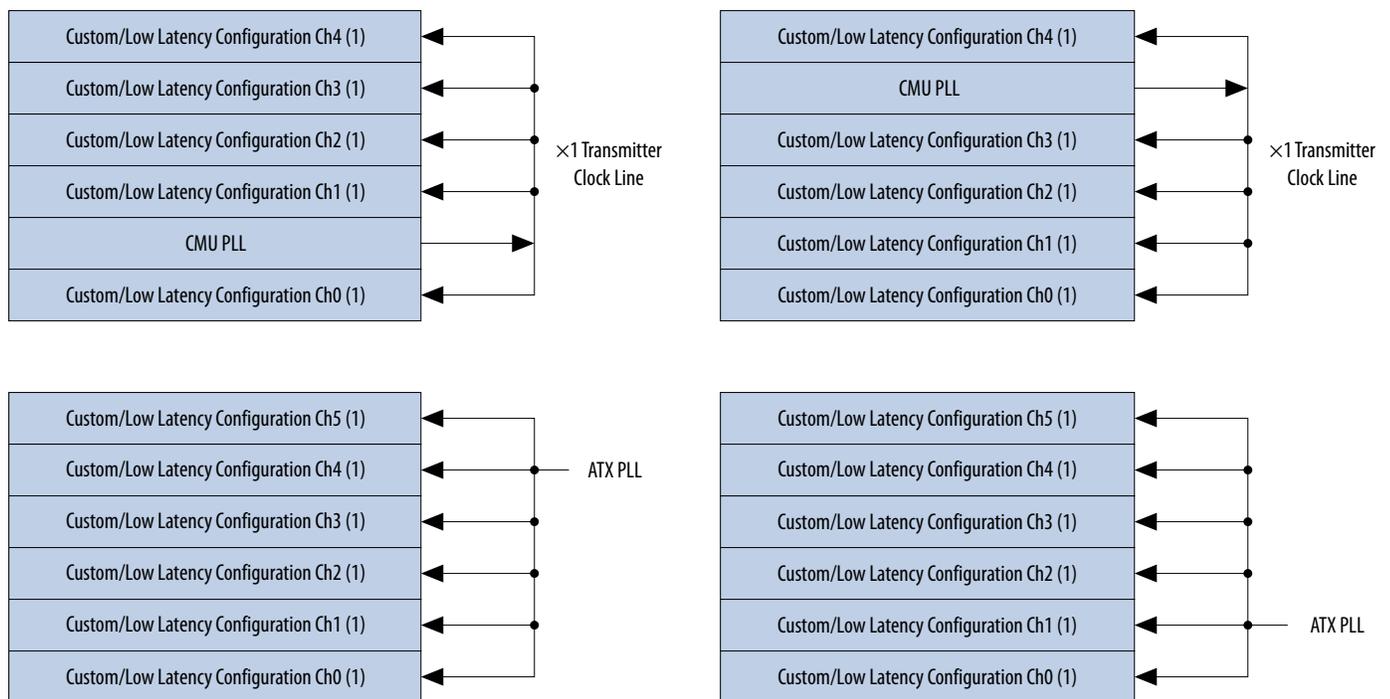
You can use CMU PLLs or ATX PLLs in non-bonded and bonded configurations.

Stratix V 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 4-63: 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.

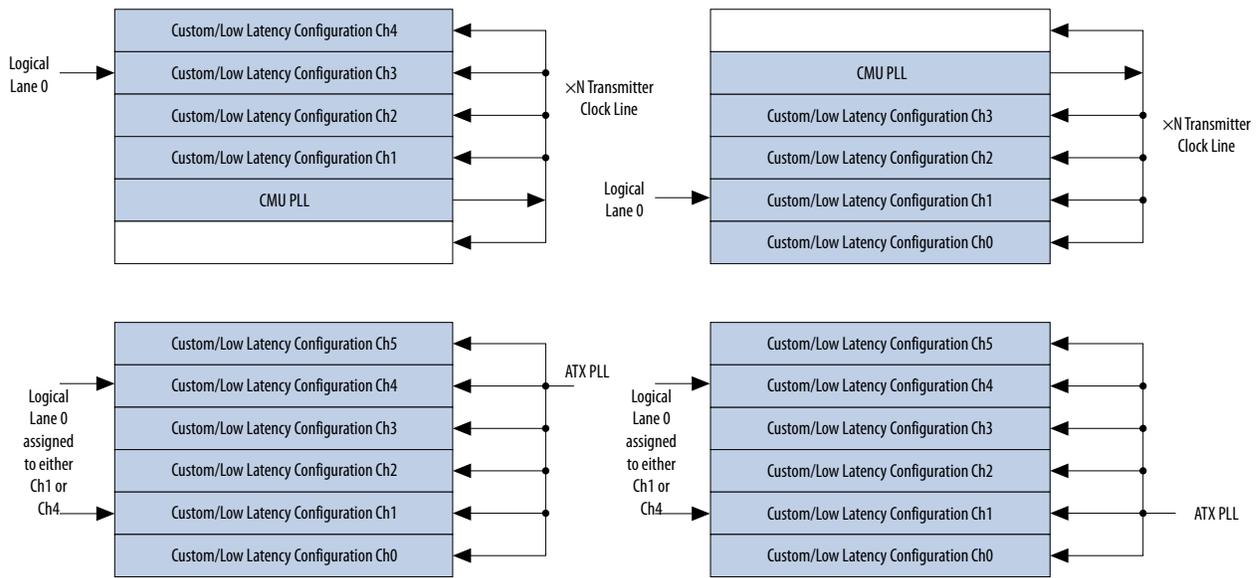


Stratix V devices allow the placement of up to four channels when a CMU PLL is used or up to six channels when an ATX PLL is used in a bonded configuration within the same transceiver bank:

- 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 4-64: Bonded Channel Placement Guidelines with Standard and 10G PCS in Custom and Low Latency Datapath Configurations



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 4-65: 10G PCS Low Latency Configuration Datapath

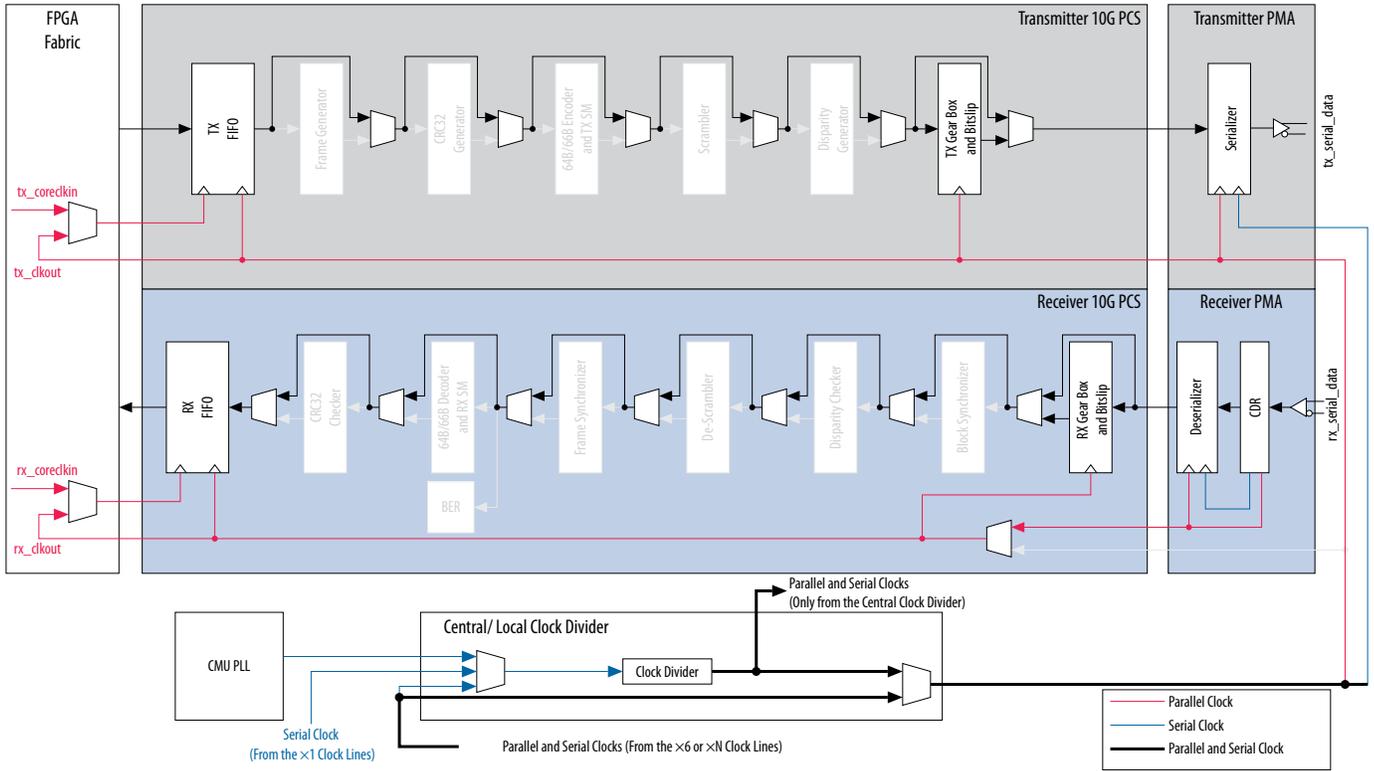
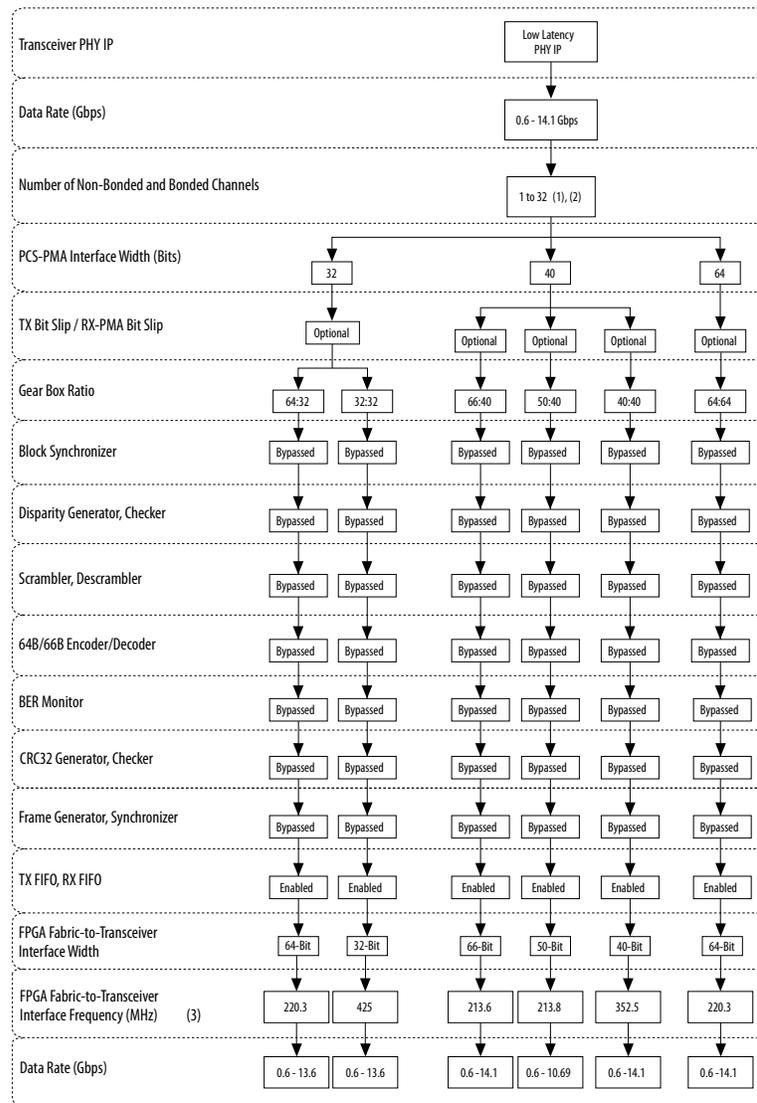


Figure 4-66: Options for 10G PCS Low Latency Configuration

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_coreclk 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 Stratix V Device Datasheet](#)
- [Transceiver Clocking in Stratix V Devices](#)
- [Refer to the Low Latency PHY IP Core chapter in the Altera Transceiver PHY IP Core User Guide](#)

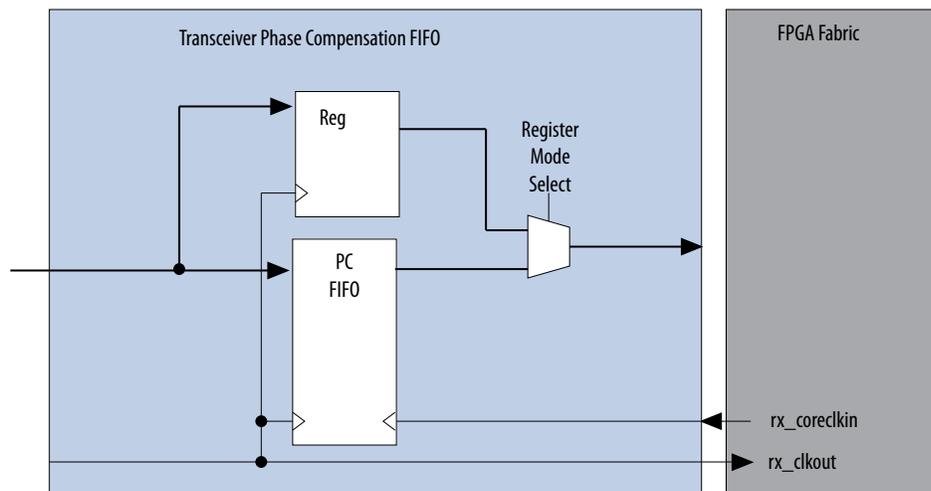
10G PCS Datapath Functionality

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 1-87.

Figure 4-67: 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

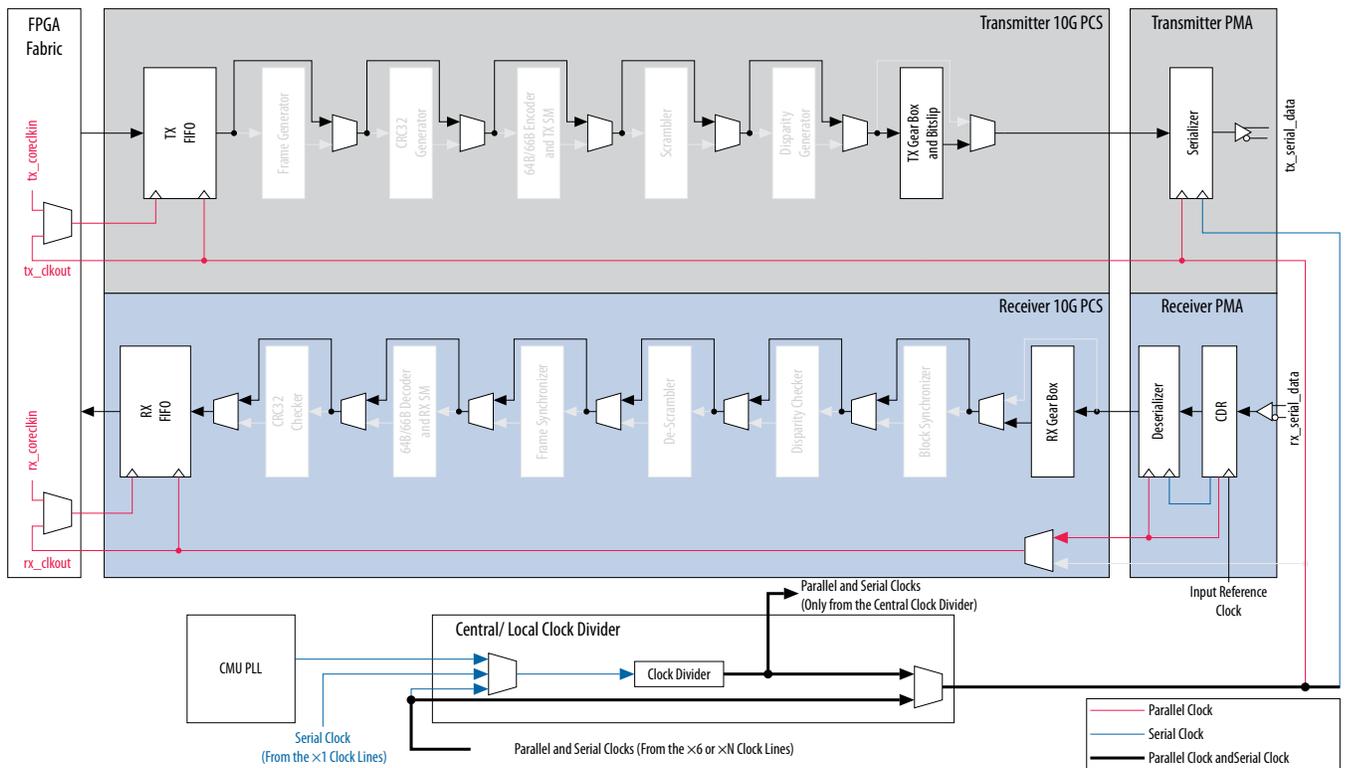
The bit slip feature allows you to slip the transmitter side bits before they are sent to the gear box. The number of bits slipped is equal to the FPGA fabric-to-transceiver interface width minus 1. For example, if the FPGA fabric-to-transceiver interface width is 64 bits, a maximum of 63 bits can be slipped. That is, $\text{bit}[63]$ from the first word and $\text{bit}[62:0]$ are concatenated to form a 64 bit word ($\text{bit}[62:0]$ from the second word, $\text{bit}[63]$ 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.

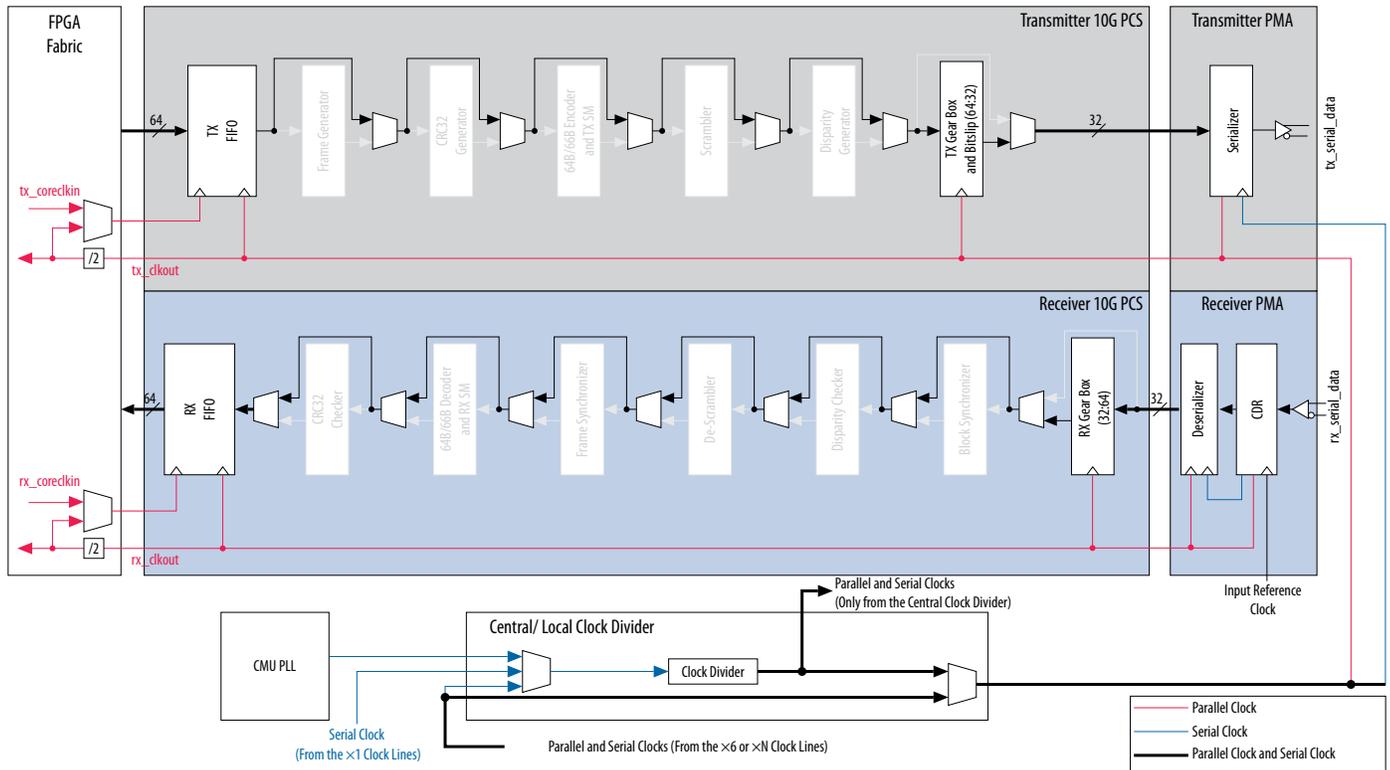
When the gear box ratio is 64:64, 40:40, or 32:32, there is no frequency difference between the read and write side of the TX and RX FIFO clocks because the gear box is the same ratio. The Quartus II software automatically connects the clocks to the read and write side of the TX FIFO and RX FIFO. In 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 4-68: 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_coreclk_in and the rx_coreclk_in ports in the Low Latency PHY IP core and connect the divided clock to these ports.

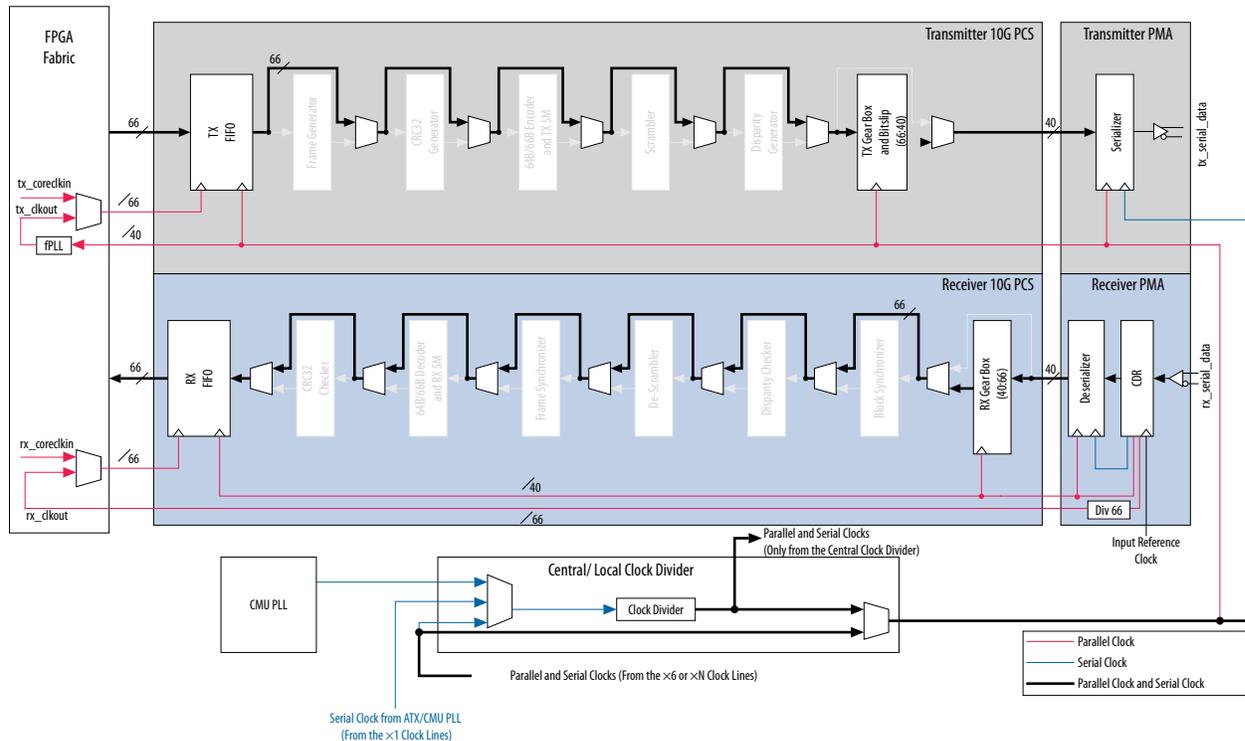
Figure 4-69: 10G PCS Low Latency Datapath with the Gear Box Ratio of 64:32



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.

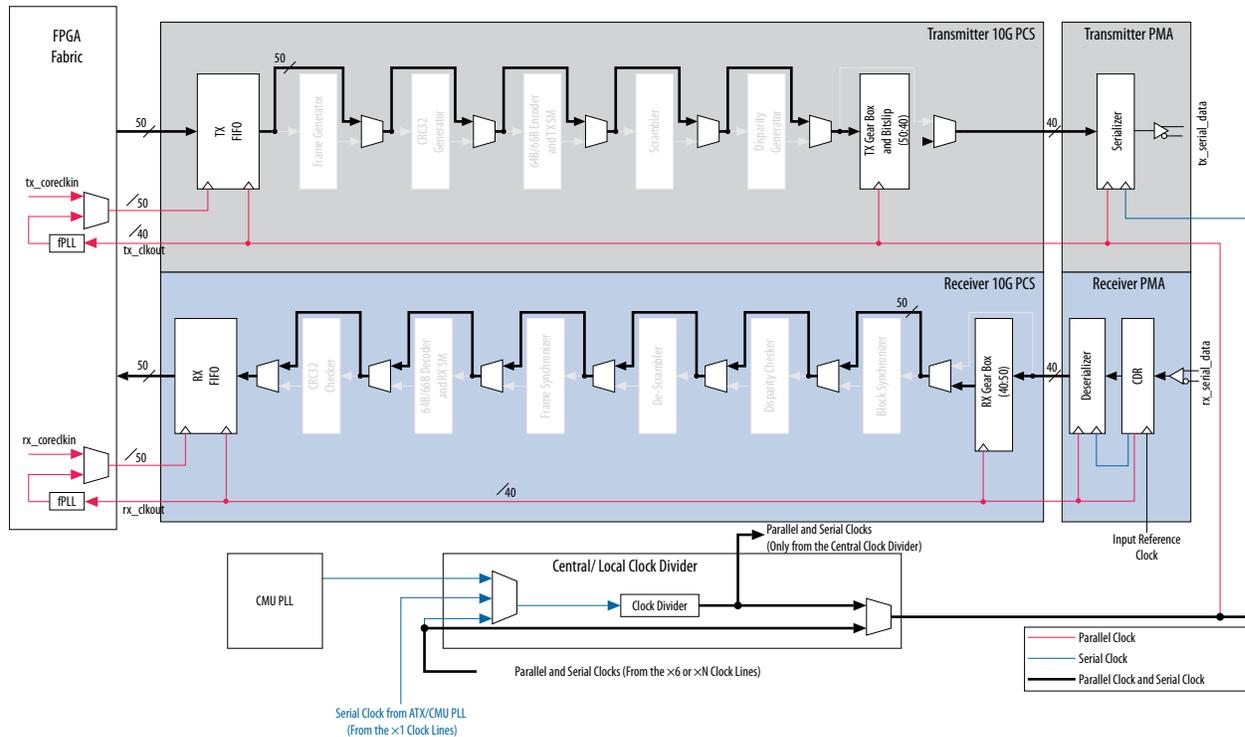
Figure 4-70: 10G PCS Low Latency Datapath with the Gear Box Ratio of 66:40



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_coreclk` port and connect a second fractional PLL output to the `rx_coreclk` 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 4-71: 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 coreclk Ports

The `tx_coreclk` and `rx_coreclk` 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_coreclk` and `rx_coreclk` 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 “User-Selected Transmitter Datapath Interface Clock” and “User-Selected Receiver Datapath Interface Clock” sections in the [Transceiver Clocking in Stratix 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

Stratix V 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 4-82

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.

The Stratix V GT transceiver channel is not supported in the Native PHY IP.

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 **Stratix 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.

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.

Figure 4-72: 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.

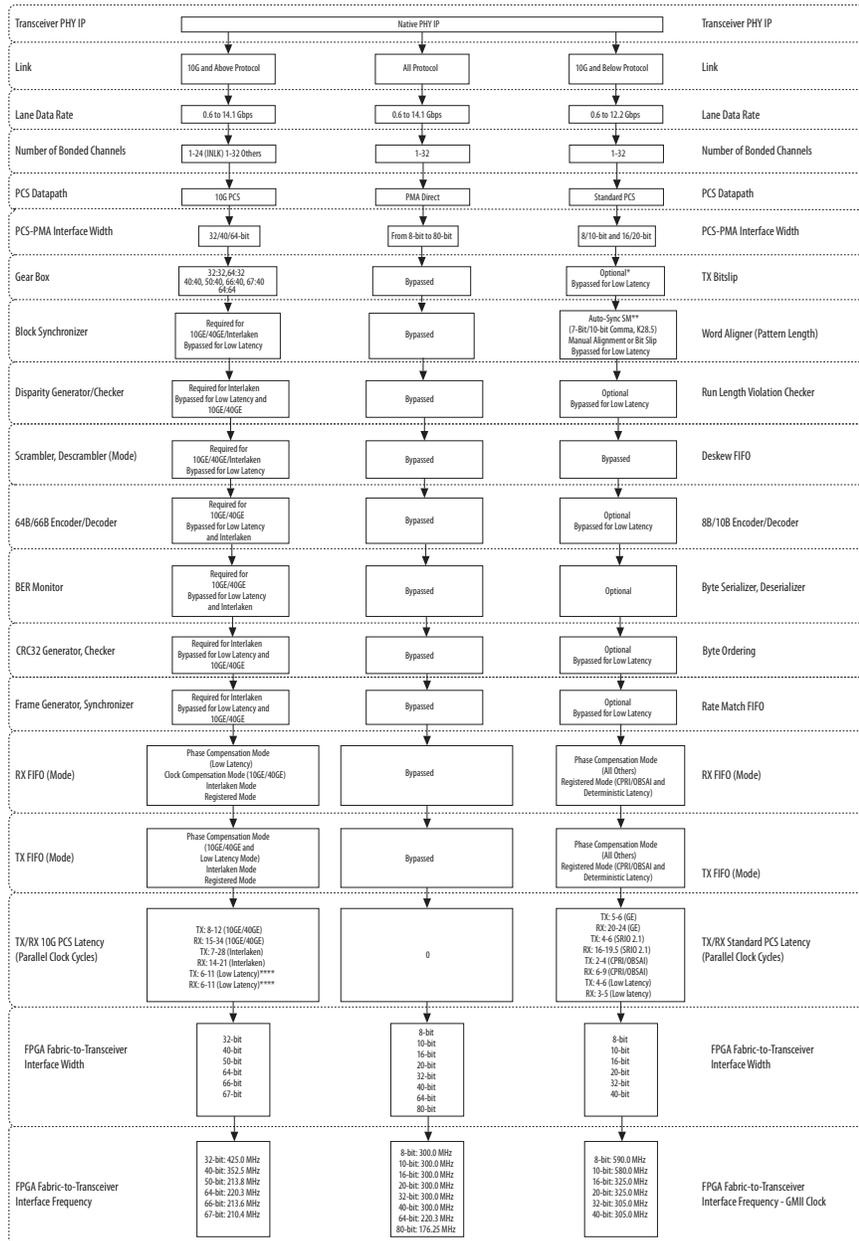
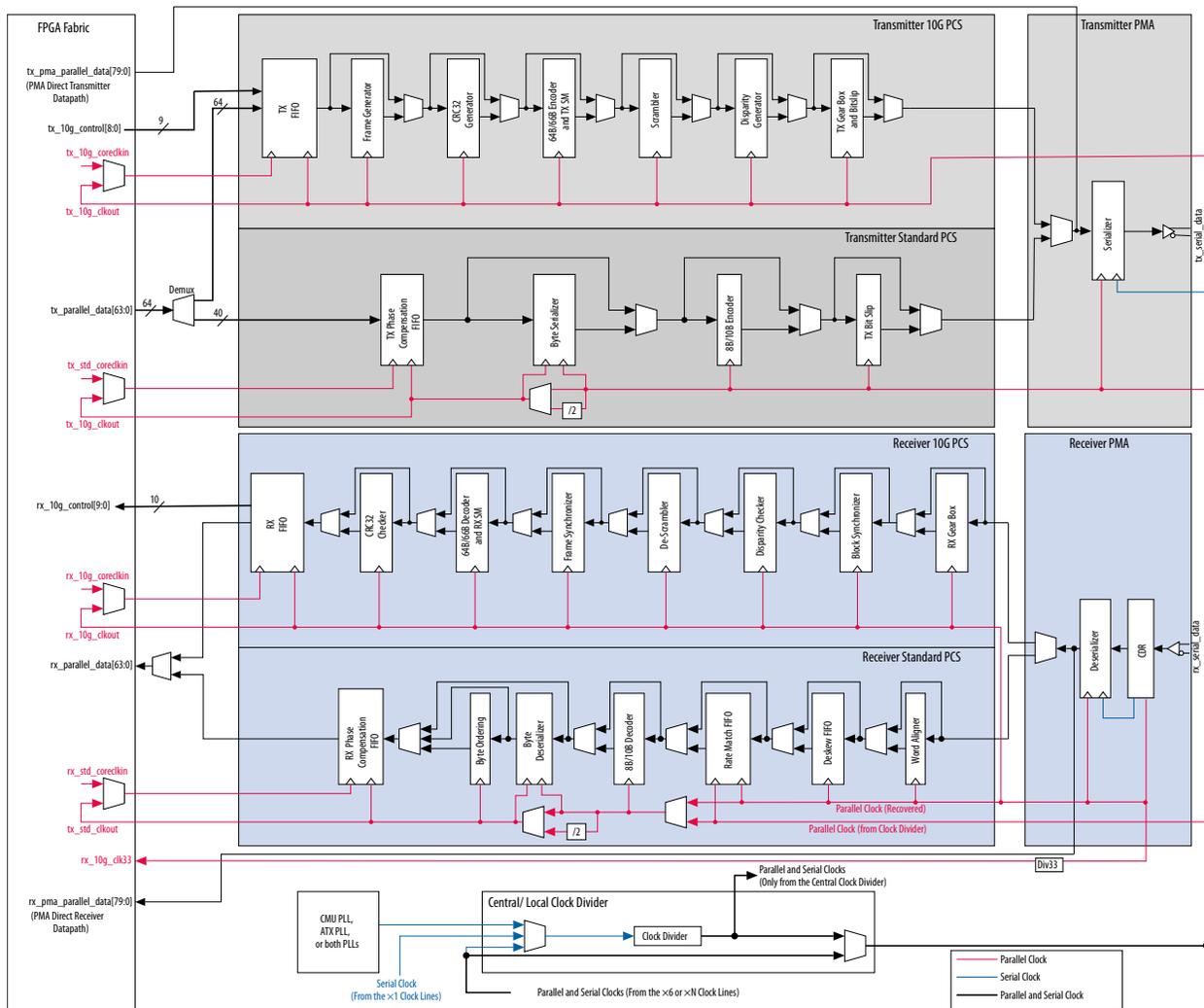


Figure 4-73: Native PHY IP Datapath Configuration

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 12.2 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 **Stratix 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 Bitstripper

Related Information

- [Transceiver Architecture in Stratix 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 10/40/100 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 14.1 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 10/40/100 Gigabit Ethernet and Interlaken. The remaining protocols are supported via 10G PCS Low Latency datapath configuration with the appropriate gearbox ratios.

10/40/100 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

10/40/100 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 Stratix 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 4-14: 10G PCS Datapath Configurations

The table lists the 10G PCS datapath configuration for 10/40/100 Gigabit Ethernet, 10/40/100 Gigabit Ethernet with 1588, Interlaken, 10G SDI, and other 10G protocols.

Transceiver PHY IP	Native PHY IP					
	Link	10/40/ 100GBASE- R/KR	10/40/ 100GBASE-R with 1588	Interlaken	SFI-5.2	10G SDI
Lane Datarate	10.3125Gbps	10.3125Gbps	3.125 - 14.1 Gbps	0.6 - 14.1 Gbps ⁽¹⁴⁾	10.692Gbps	0.6 - 14.1 Gbps ⁽¹⁴⁾
PMA Channel Bonding Option ⁽¹⁵⁾ ⁽¹⁶⁾	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 ⁽¹⁷⁾	66:40 ⁽¹⁷⁾	67:40	32:32, 64:32 ⁽¹⁷⁾ , 40:40, 64:64	50:40 ⁽¹⁷⁾	32:32, 64:32 ⁽¹⁷⁾ , 40:40, 66:40 ⁽¹⁷⁾ , 64:64

⁽¹⁴⁾ Gearbox ratios of 64:32 and 32:32 have a maximum supported datarate of 13.6Gbps.

⁽¹⁵⁾ 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.

⁽¹⁶⁾ 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.

⁽¹⁷⁾ May require the use of an internal fractional PLL (fPLL) for selected Gearbox ratio.

Transceiver PHY IP	Native PHY IP					
Link	10/40/ 100GBASE- R/KR	10/40/ 100GBASE-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	10/40/ 100GBASE- R/KR	10/40/ 100GBASE-R with 1588	Interlaken	SFI-5.2	10G SDI
TX/RX 10G PCS Latency (Parallel Clock Cycles) ⁽¹⁸⁾	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 ⁽¹⁹⁾	32-bit (32:32): 340.0 MHz 40-bit (40:40): 312.5 MHz 64-bit (64:32): 170.0 MHz ⁽²⁰⁾ 64-bit (64:64): 195.4 MHz	50-bit: 213.8 MHz ⁽¹⁹⁾	32-bit (32:32): 340.0 MHz 40-bit (40:40): 312.5 MHz 64-bit (64:32): 170.0 MHz ⁽²⁰⁾ 64-bit (64:64): 195.4 MHz 66-bit (66:40): 189.4 MHz ⁽¹⁹⁾

⁽¹⁸⁾ PCS Latency values are with default recommended FIFO partially full and partially empty values. Disabled if Standard PCS 8B/10 Encoder/Decoder is used.

⁽¹⁹⁾ PCS tx_clkout frequency output is lane datarate/40 for 10G-SDI, Interlaken, and Basic Mode.

⁽²⁰⁾ 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 14.1Gbps 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 4-73 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

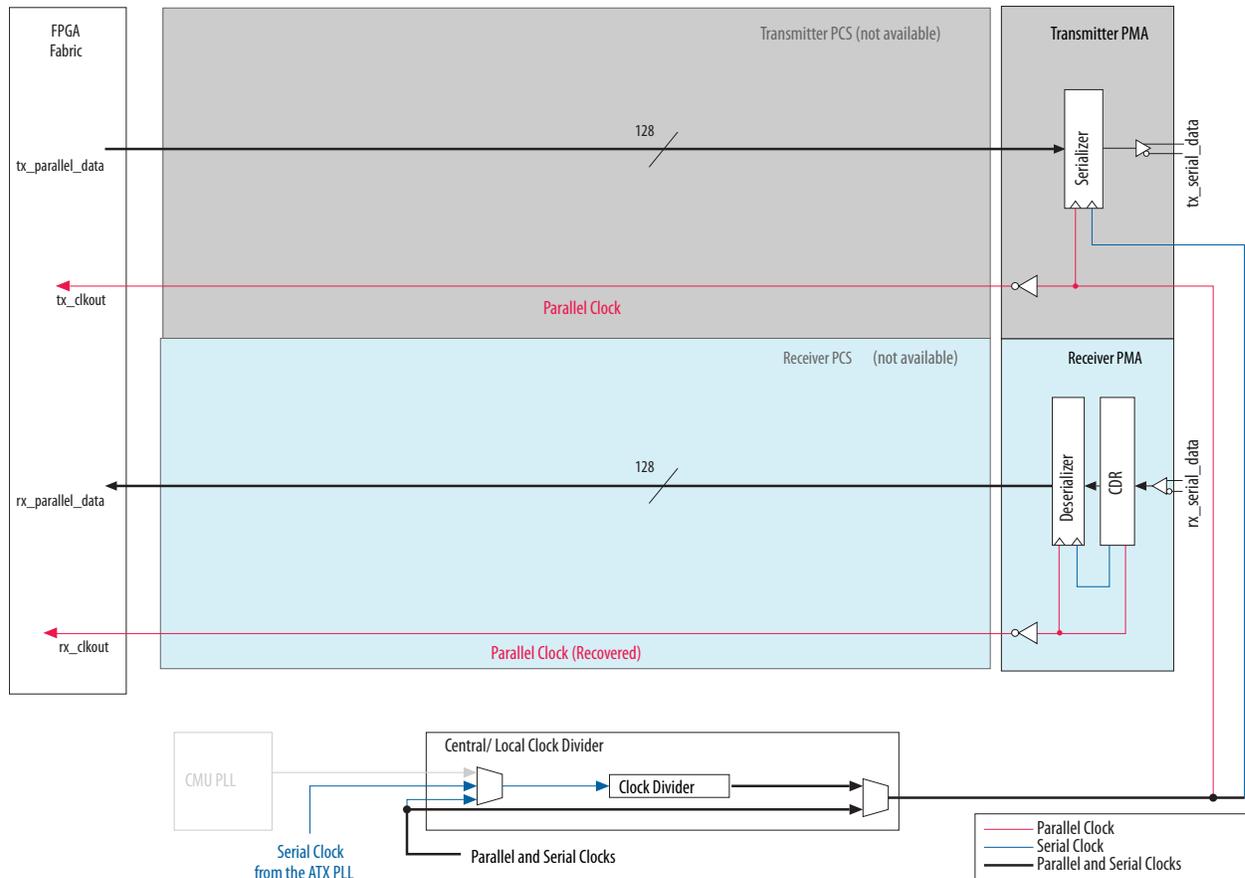
- [Dynamic Reconfiguration in Stratix V Devices](#)
- [Altera Transceiver PHY IP Core User Guide](#)

Stratix V GT Device Configurations

Stratix V GT devices contain both 28.05 Gbps GT transceivers and 12.5 Gbps GX transceivers. The GT transceivers can be configured only in PMA-Direct configuration using the Low Latency PHY IP. The GT channels have serial data rates that range from 19.6 Gbps to 28.05 Gbps.

To implement a Low Latency PHY link with the GT channel, instantiate the **Low Latency PHY IP** core in the IP Catalog, under **Transceiver PHY** in the Interfaces menu. In the **Low Latency GUI** under the **General** tab, select **GT** on the **Datapath type** selection.

Figure 4-74: Transceiver Datapath and Clocking in Stratix V GT Channels



Related Information

- [Transceiver Configurations](#) on page 4-71
Refer to this section for information about configuring GX transceivers.
- [For the clocking details of Stratix V GT channels, refer to Transceiver Clocking in Stratix V Devices](#)

Document Revision History

Table 4-15: Document Revision History

Date	Version	Changes
February 2017	2017.02.15	<ul style="list-style-type: none"> • Added a note to the "Framing Layer Control Word Forwarding" portion of the "Supported Features" section.

Date	Version	Changes
September 2014	2014.09.30	<ul style="list-style-type: none"> Added a note about autonegotiation to the "Rate Match FIFO in 1000BASE-X and 1000BASE-KX Configurations" portion of the "1000BASE-X and 1000BASE-KX Supported Features" section. Added clock labels to the "Native PHY IP Datapath Configuration" figure.
January 2014	2014.01.07	<ul style="list-style-type: none"> Added a note to the "Auto-Speed Negotiation Block" section. Updated "Sample Channel Width Options for Supported Serial Data Rates" table in the "CPRI and OBSAI" section.
October 2013	2013.10.11	<ul style="list-style-type: none"> Updated "Advanced Channel Placement Guidelines for PIPE Configurations" section. Updated "Transceiver Clocking for PCIe Gen3" section.
May 2013	2013.05.06	<ul style="list-style-type: none"> 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.
February 2013	2013.02.21	<ul style="list-style-type: none"> Added "Stratix V GT Device Configurations" section. Updated "Transceiver Clocking and Channel Placement Guidelines" section for PCI Express. Updated "Transceiver Channel Datapath for XAUI Configuration" and "Transceiver Clocking Diagram for XAUI Configuration" figures.
December 2012	2012.12.17	<ul style="list-style-type: none"> Added 10GBASE-KR information. Added Native PHY information.

Date	Version	Changes
June 2012	2.3	<ul style="list-style-type: none"> • Added “CPRI and OBSAI—Deterministic Latency Protocols” section. • Added “Multi-Lane Deskew Alignment”, “Transmit and Receive FIFO Control and Status”, and “Transceiver Multi-Lane Bonding and Transmit Skew” sections to the Interlaken section. • Updated the “Transceiver Channel Placement Guidelines” section in the XAUI section. • Updated Figure 4–6, Figure 4–12, Figure 4–13, Figure 4–19, Figure 4–20, Figure 4–26, Figure 4–36, Figure 4–38, Figure 4–40, Figure 4–41, Figure 4–42, Figure 4–43, Figure 4–45, Figure 4–46, Figure 4–47, Figure 4–48, Figure 4–50, Figure 4–52. • Updated Table 4–1, Table 4–6, Table 4–7, Table 4–8, Table 4–13, Table 4–14, Table 4–15.
February 2012	2.2	<ul style="list-style-type: none"> • Added Figure 4–38 and Figure 4–39. • Updated Figure 4–3, Figure 4–5, Figure 4–6, Figure 4–7, Figure 4–8, Figure 4–9, Figure 4–11, Figure 4–13, Figure 4–14, Figure 4–26, Figure 4–27, Figure 4–38, Figure 4–39, Figure 4–42, and Figure 4–45. • Added Table 4–4. • Updated Table 4–5. • Removed “Transceiver Channel Placement Guidelines” in Interlaken section. • Removed “Transceiver Channel Placement Guidelines” in 10GBASE-R section.

Date	Version	Changes
December 2011	2.1	<ul style="list-style-type: none"> Updated Figure 4–3, Figure 4–5, Figure 4–7, Figure 4–8, Figure 4–9, Figure 4–11, Figure 4–12, Figure 4–14, Figure 4–15, Figure 4–16, Figure 4–17, Figure 4–19, Figure 4–20, Figure 4–21, Figure 4–22, Figure 4–23, Figure 4–30, Figure 4–31, Figure 4–32, Figure 4–33, Figure 4–34, Figure 4–35, Figure 4–36, Figure 4–43, Figure 4–45, and Figure 4–48. Updated Table 4–3 and Table 4–9. Removed “Clock Compensation for Repeater Applications” section. Minor text edits.
November 2011	2.0	<ul style="list-style-type: none"> Merged the 11.0 Transceiver Custom Configurations chapter into this chapter and reorganized this chapter. Added PCI Gen3 information. Added Stratix V GT device information. Removed “GIGE” section.
May 2011	1.2	<ul style="list-style-type: none"> Updated Figure 4–8 and Figure 4–9. Updated “Supported Features” on page 4–10. Updated Table 4–5. Updated Figure 4–16, Figure 4–18, and Figure 4–19. Added “GIGE” section. Updated “XAUI” on page 4–38. Updated “Transceiver Datapath in a XAUI Configuration” on page 4–39. Updated “Transceiver Channel Placement Guidelines” on page 4–44. Updated Figure 4–33. Chapter moved to volume 3 for the 11.0 release.
December 2010	1.1	<ul style="list-style-type: none"> Updated “PCI Express (PIPE) 2.0 Interface”, “Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates”, “Receiver Status”, and “Receiver Detection” sections. Updated Figure 4–32.
April 2010	1.0	Initial release.

Transceiver Loopback Support in Stratix V Devices

5

2014.09.30

SV52007



Subscribe



Send Feedback

The Stratix V loopback options allow you to verify how different functional blocks work in the transceiver.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the Stratix V Device Handbook chapters.

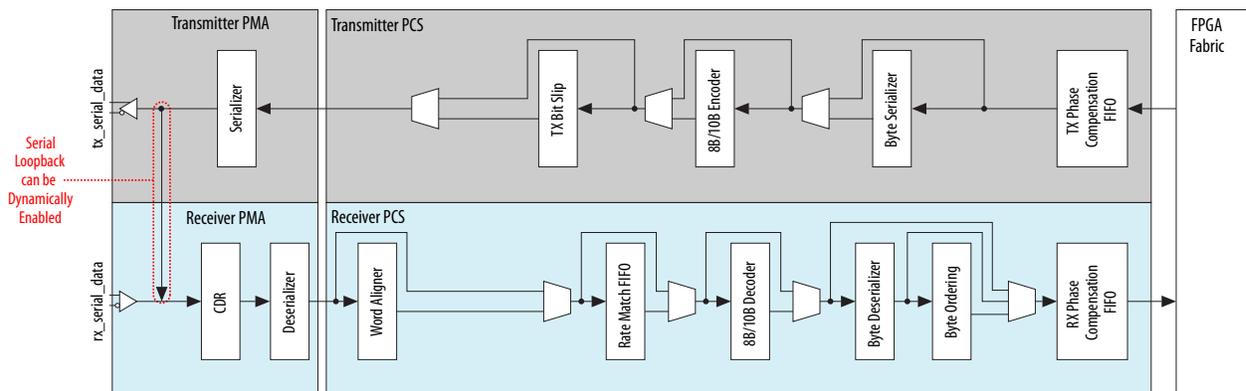
Serial Loopback

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 5-1: 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

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

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.

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.

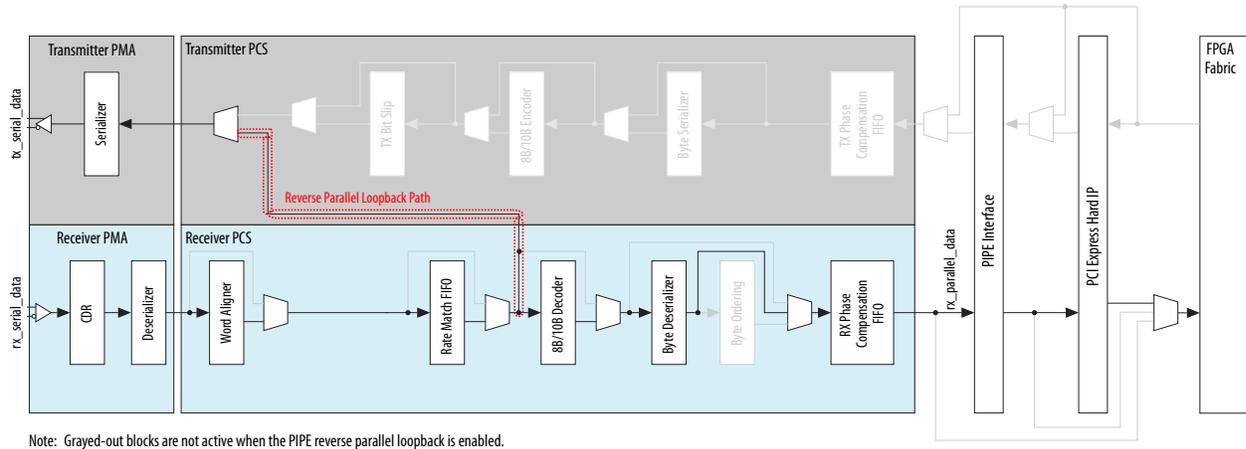
PIPE reverse parallel loopback is only available in the PCIe[®] configuration for Gen1 and Gen2 data rates. The following figure shows the received serial data passing through the receiver CDR, deserializer, word aligner, and rate match FIFO buffer. The parallel data from the rate match FIFO is then 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 `tx_detectrxloopback` signal.

Note: PIPE reverse parallel loopback is the only loopback option supported in the PCIe configuration. PIPE reverse parallel loopback is not supported in the GT channels of Stratix V GT devices.

Note: For more information, refer to the "PCI Express Reverse Parallel Loopback" section in the Transceiver Configurations in Stratix V Devices chapter.

Figure 5-2: PIPE Reverse Parallel Loopback Configuration Datapath



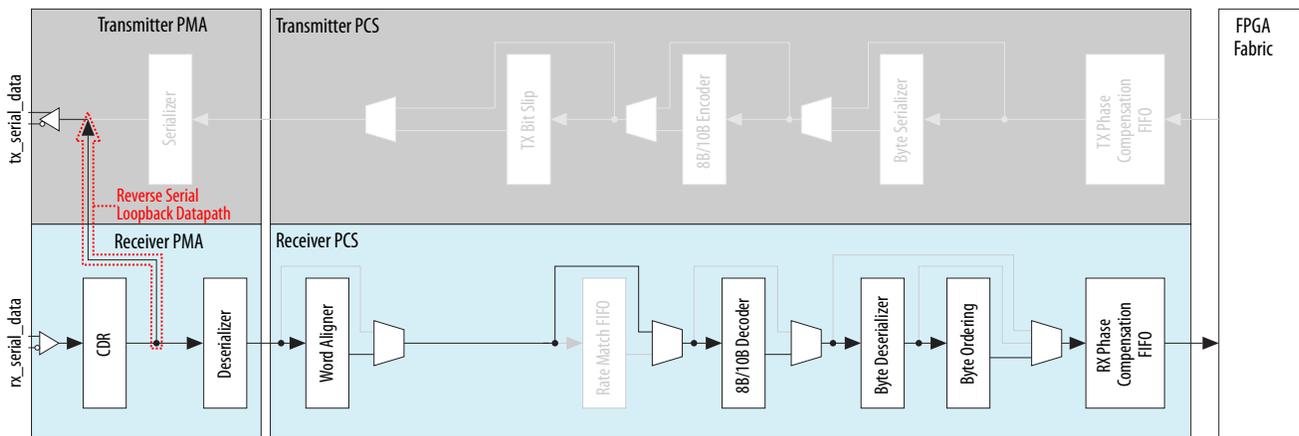
Related Information

[Transceiver Configurations in Stratix V Devices](#)

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.

Figure 5-3: Reverse Serial Loopback Datapath



Note: Grayed-out blocks are not active when the reverse serial loopback is enabled.

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.

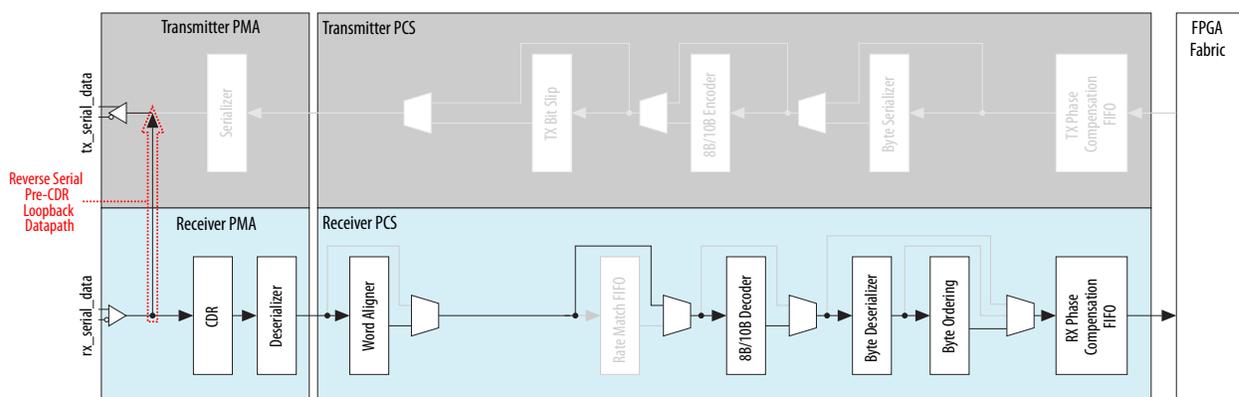
Related Information

[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 5-4: Reverse Serial Pre-CDR Loopback Datapath



Note: Grayed-out blocks are not active when the reverse serial pre-CDR loopback is enabled.

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. In pre-CDR reverse loopback, RX input main data passes through the RX buffer, then loops back to the TX directly. There is no clock in this path. 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.

In post-CDR reverse loopback, the CDR clock can generate pre-emphasis data from the main data and loopback to the TX. However, only 1st post-tap data is generated and sent with the main data to the TX. This is for design and layout cost considerations. The 1st post-tap is the most used tap, and is covered in the test mode.

Related Information

[Altera Transceiver PHY IP Core User Guide](#)

Document Revision History

Table 5-1: Document Revision History

Date	Version	Changes
September 2014	2014.09.30	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base. Updated the Reverse Serial Pre-CDR Loopback topic.
December 2012	2012.12.17	Reorganized content and updated template.
June 2012	2.4	Update for the Quartus II software version 12.0.
February 2012	2.3	Minor edits for clarity.
December 2011	2.2	Updated the document to clarify information.
November 2011	2.1	<ul style="list-style-type: none"> This chapter was formerly chapter 6. There are no content changes for this version of the chapter.
May 2011	2.0	<ul style="list-style-type: none"> Added the “Reverse Serial Loopback” and “Reverse Serial Pre-CDR Loopback” sections. Updated Figure 5–2. Updated the chapter title. Chapter moved to Volume 3. Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.

Date	Version	Changes
July 2010	1.0	Initial release.

2014.09.30

SV52008



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

[Stratix V Device Handbook: Known Issues](#)

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

Dynamic Reconfiguration Features

The following table lists the available dynamic reconfiguration features.

Table 6-1: 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
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

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

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	<ul style="list-style-type: none"> TX PLL
	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.

Offset Cancellation

Offset cancellation adjusts the offsets within the RX PMA and the CDR parameters for process variations.

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`) to the Transceiver Reconfiguration Controller is also used for transceiver calibration and must be within the range of 100-125 MHz. If the clock (`mgmt_clk_clk`) is not free-running, the reconfiguration controller reset (`mgmt_rst_reset`) must be held in reset until the clock is stable.

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

The dynamic reconfiguration controller provides an Avalon[®] Memory-Mapped (Avalon-MM) user interface to dynamically reconfigure individual PMA settings on a per channel basis.

Related Information

- [Altera Transceiver PHY IP Core User Guide](#)
For information about the read and write operations with the reconfiguration controller
- [AN 645: Dynamic Reconfiguration of PMA Controls in Stratix V Devices](#)

On-Chip Signal Quality Monitoring ()

The bit error rate (BER) eye contour can be used to measure the quality of the received data. is a debug and diagnostic tool that analyzes the received data recovery path, including the receiver's gain, noise level, and recovery clock jitter. can also measure vertical eye height, effectively allowing a BER eye contour to be plotted.

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 feature.

Related Information

[Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide.](#)
For information about enabling the feature.

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.

The Transceiver Reconfiguration Controller provides an Avalon-MM user interface to step through the DFE tap settings.

Note: In Stratix V GT devices, the GT channels do not support DFE. This feature is supported in the GX channels.

Related Information

[“DFE” section in the Transceiver Reconfiguration Controller chapter of the Altera Transceiver PHY IP Core User Guide](#)

For more information about DFE.

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.

In Stratix V GT devices, the GT channels do not support AEQ. This feature is supported in the GX channels.

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.

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:

- **Serial loopback path**— The output from the serializer is fed back to the CDR. While in this mode, the serializer also feeds the data to the TX output port. Enabling or disabling serial loopback mode is done through the PHY management interface.
- **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.

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.

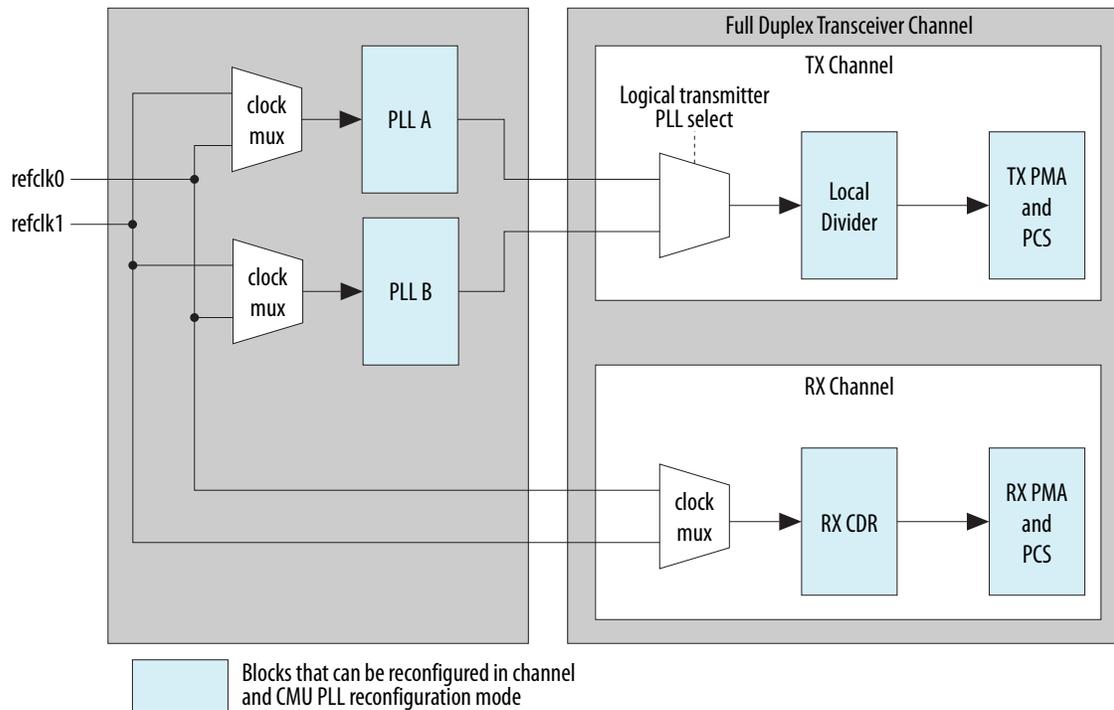
For example, you can reconfigure the custom PHY IP to enable or disable the 8B/10B encoder/decoder. There is no limit to the number of functional modes you can reconfigure the transceiver channel to if the various clocks involved support the transition. When you switch the custom PHY IP from one function 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 6-1: 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.



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.

Document Revision History

Date	Version	Changes
September 2014	2014.09.30	Added FPGA fabric to transceiver channel interface width reconfiguration feature in <i>Table: Dynamic Reconfiguration Features</i> .
May 2013	2013.05.06	Added link to the known document issues in the Knowledge Base.

Date	Version	Changes
December 2012	2012.12.17	<ul style="list-style-type: none">• Rewritten and reorganized content, and updated template
February 2012	2.2	<ul style="list-style-type: none">• Updated the “Decision Feedback Equalization” section.• Minor text edits
December 2011	2.1	Minor text edits.