2. Transceiver Clocking in Stratix IV Devices

SIV52002-3.4

This chapter provides detailed information about the Stratix® IV transceiver clocking architecture. For this chapter, the term "Stratix IV devices" includes both Stratix IV GX and GT devices. Similarly, the term "Stratix IV transceivers" includes both Stratix IV GX and GT transceivers.

The clocking architecture chapter is divided into three main sections:

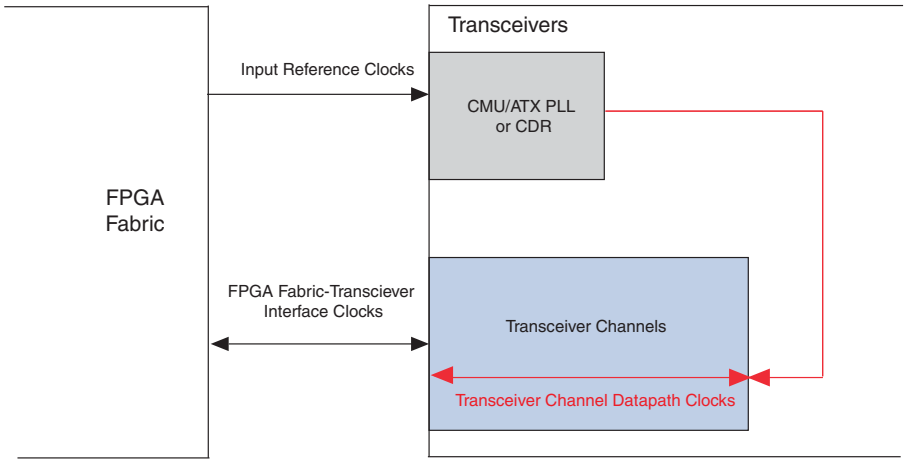
- "Input Reference Clocking" on page 2–2—describes how the reference clock is provided to the clock multiplier unit (CMU)/auxiliary transmit phase-locked loop (ATX PLL) to generate the clocks required for transceiver operation.
- "Transceiver Channel Datapath Clocking" on page 2–20—describes the clocking architecture internal to the transceiver block.
- "FPGA Fabric-Transceiver Interface Clocking" on page 2–51—describes the clocking options available when interfacing the transceiver with the FPGA fabric.

Other sections in this chapter include:

- "FPGA Fabric PLLs-Transceiver PLLs Cascading" on page 2–9
- "Using the CMU/ATX PLL for Clocking User Logic in the FPGA Fabric" on page 2–71
- "Configuration Examples" on page 2–72

Figure 2–1 shows an overview of the clocking architecture.

Figure 2–1. Clocking Architecture Overview



Glossary of Terms

Table 2–1 lists the terms used in the chapter.

Table 2–1. Glossary of Terms Used in this chapter

Convention	Description
ATX PLL	Auxiliary transmit PLL block. For more information, refer to the "Auxiliary Transmit (ATX) PLL Block" section in the *Transceiver Architecture in Stratix IV Devices* chapter.
CDR	Clock data recovery block. For more information, refer to the "Clock and Data Recovery Unit" section in the *Transceiver Architecture in Stratix IV Devices* chapter.
CMU	Clock multiplier unit. For more information, refer to "CMU Channel Architecture" section in the *Transceiver Architecture in Stratix IV Devices* chapter.
ITB lines	The Inter-Transceiver block (ITB) clock lines provide an input reference clock path from the `refclk` pins of one transceiver block CMU PLLs and receiver CDRs of other transceiver blocks. They also provide input reference clock to ATX PLLs. For more information, refer to "Inter-Transceiver Block (ITB) Clock Lines" on page 2–8.

Input Reference Clocking

Each transceiver block has:

- Two clock multiplier unit channels—the `CMU0_Channel` and `CMU1_Channel`

  You can configure each as either a CMU to generate transceiver clocks or as a PMA-Only channel.

  For more information, refer to the "CMU Channel Architecture" section in the *Transceiver Architecture in Stratix IV Devices* chapter.

- Four regular channels

  When the CMU channel is configured as a CMU, the CMU PLL synthesizes the input reference clock to generate the high-speed serial transceiver clock. When the CMU channel is configured as a **Receiver Only** or **Receiver and Transmitter** channel, the CMU PLL acts as a CDR and uses the input reference clock as a training clock when it is in lock-to-reference (LTR) mode. Each of the four regular channels also has a receiver CDR that uses the input reference clock as a training clock when it is in LTR mode.

  Each Stratix IV device also has ATX PLLs that you can use in addition to the CMU PLLs to generate the high-speed serial transceiver clock. The ATX PLLs also need an input reference clock for operation. 6G ATX PLLs are available in both Stratix IV GX and Stratix IV GT devices. 10G ATX PLLs are available only in Stratix IV GT devices.

  For more information, refer to the "Auxiliary Transmit (ATX) PLL Block" and the "Transmitter Channel Datapath" sections in the *Transceiver Architecture in Stratix IV Devices* chapter.

Input Reference Clock Source

Receiver clock data recoveries (CDRs), CMU PLLs (when the CMU channel is configured as a CMU) and ATX PLLs can derive the input reference clock from one of the sources listed in Table 2–2.

Table 2–2. Input Reference Clock Source

Index	Clock Source	CMU PLL	6G ATX PLL	10G ATX PLL	CDR	Jitter Performance (4)
1	refclk0 and refclk1 pins of the same transceiver block	Yes	No (1)	No (1)	Yes	1
2	refclk0 and refclk1 pins of other transceiver blocks on the same side of the device using the ITB clock lines (2)	Yes	Yes	Yes	Yes	2 (3)
3	Clock output from the left and right PLLs in the FPGA fabric with voltage controlled oscillator (VCO) bypass mode (5), (6)	Yes	Yes	No	Yes	3
4	Clock output from the left and right PLLs in the FPGA fabric	Yes	Yes	No	Yes	4
5	Dedicated CLK input pins on the FPGA global clock network	Yes	Yes	No	Yes	4

Notes to Table 2–2:

(1) ATX PLLs do not have dedicated refclk pins.

(2) For more information, refer to "Inter-Transceiver Block (ITB) Clock Lines" on page 2–8.

(3) For better jitter performance, Altera strongly recommends using the refclk0 and refclk1 pins of the transceiver block located immediately below the ATX PLL.

(4) Lowest number indicates best jitter performance.

(5) For more information, refer to "Configuration Examples" on page 2–72.

(6) When in VCO bypass mode, you can only divide the reference clock by the N integer. For more information, refer to "Left and Right, Left, or Right PLL in VCO Bypass Mode" on page 2–17.

When a CMU channel is configured as a channel, its CMU PLL acts as a receiver CDR and can derive the input reference clock sources 2 through 5 listed in the Table 2–2. You can also use the refclk pin of the other CMU channel within the transceiver block as a clock source as long as the other CMU channel is not configured as a **Receiver only** or **Receiver and Transmitter** channel. For example, the CMU0 PLL can derive its input reference clock from the refclk1 pin if the CMU1 channel is not configured as a **Receiver only** or **Receiver and Transmitter** channel.

When a CMU channel is configured as a channel, its refclk pin is used to receive serial input data. As a result, the refclk pin is not available to provide the input reference clock.

Table 2–3 lists the input reference clock frequencies allowed for the 10G ATX PLL.

Table 2–3. Input Reference Clock Frequencies for the 10G ATX PLL Clock

Data Rate (Gbps)	Allowed Divider Values	Reference Clock Frequency (MHz)
9.9 to 11.3	M = 16, N = 1	281.25 to 322
	M = 16, N = 2	562.5 to 706.25

Figure 2–2 shows the input reference clock sources for CMU PLLs and receiver CDRs within a transceiver block. One global clock line is available for each CMU PLL and receiver CDR in a transceiver block. This allows each CMU PLL and receiver CDR to derive its input reference clock from a separate FPGA CLK input pin.

Figure 2–2. Input Reference Clock Sources in a Transceiver Block

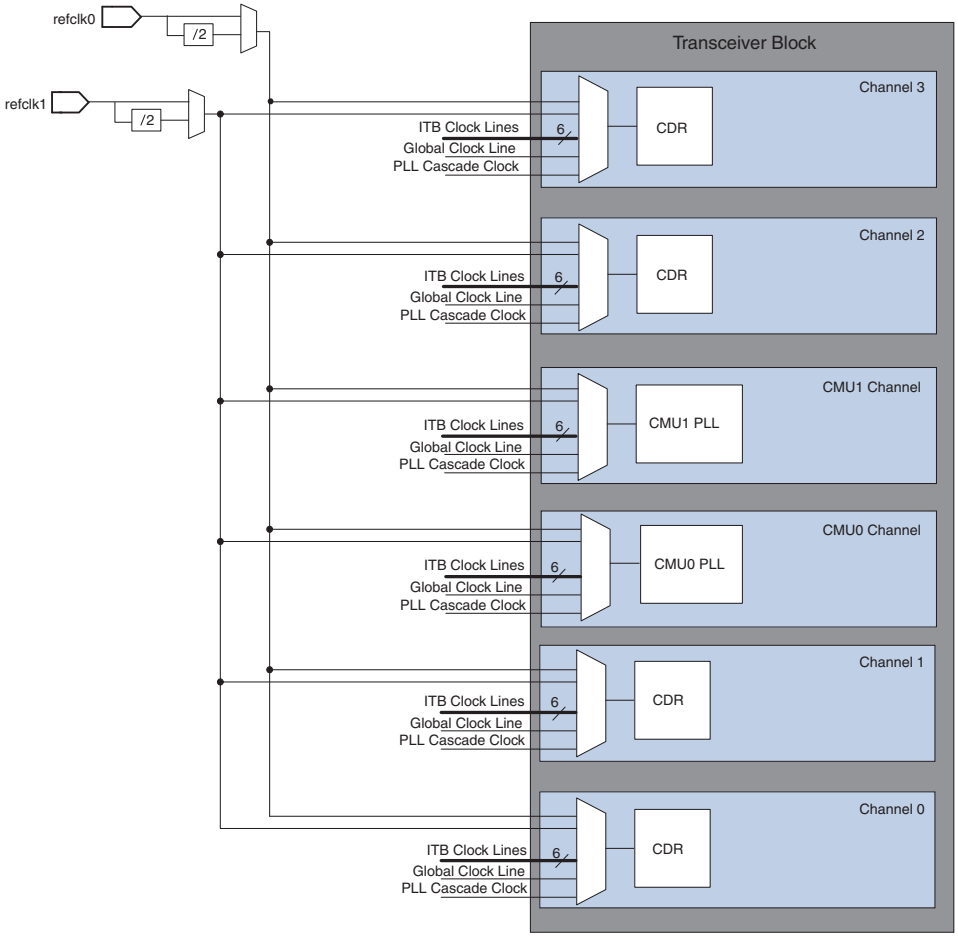


Figure 2–3 shows the input reference clock sources for CMU PLLs, ATX PLLs, and receiver CDRs in four transceiver blocks on the right side of the EP4SGX530F45 device. In this figure, the input reference clock sources for four transceiver blocks are located only on the right side of the device but the EP4SGX530NF45 device has similar input reference clock resources available for the four transceiver blocks located on the left side of the device as well.

Figure 2–3 also shows the ITB clock lines on the right side of the device. The number of ITB clock lines available in any Stratix IV GX device is equal to the number of `refclk` pins available in that device.

Figure 2–3. Input Reference Clock Sources Across Transceiver Blocks

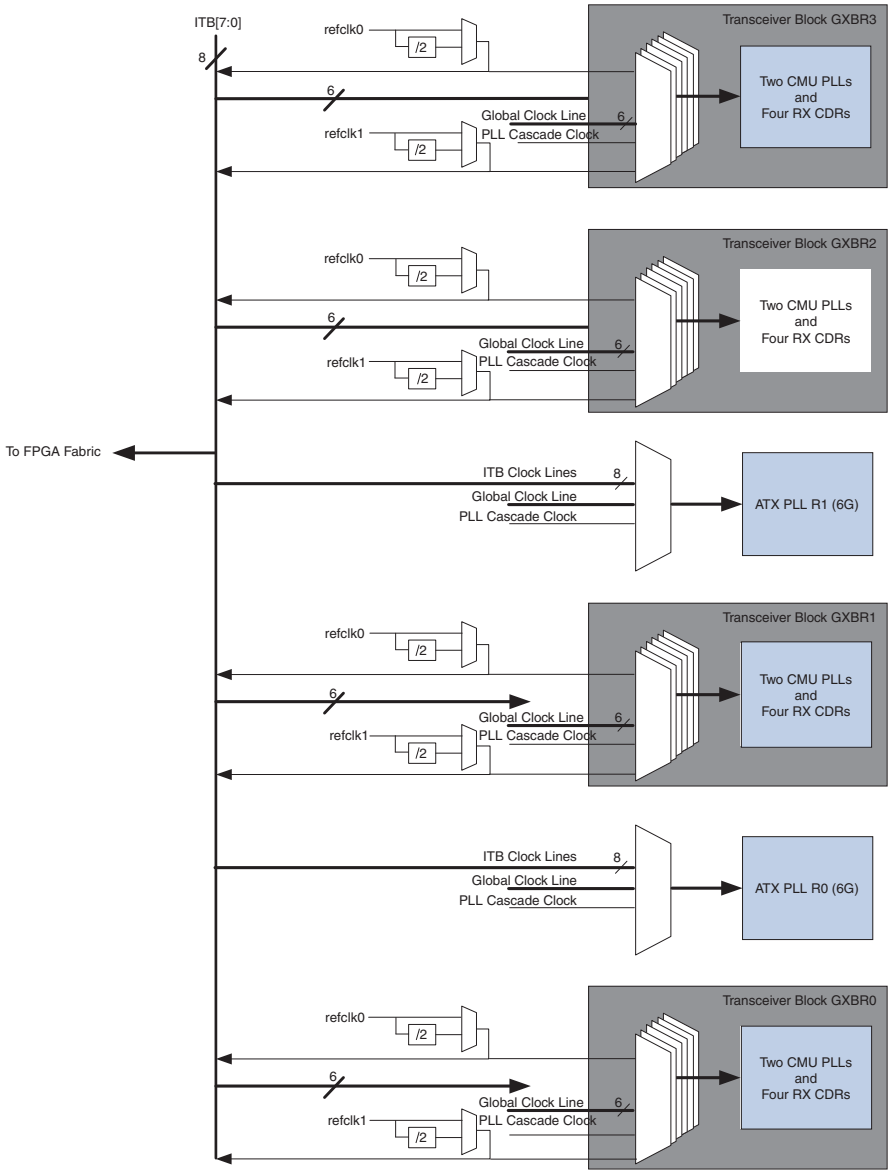
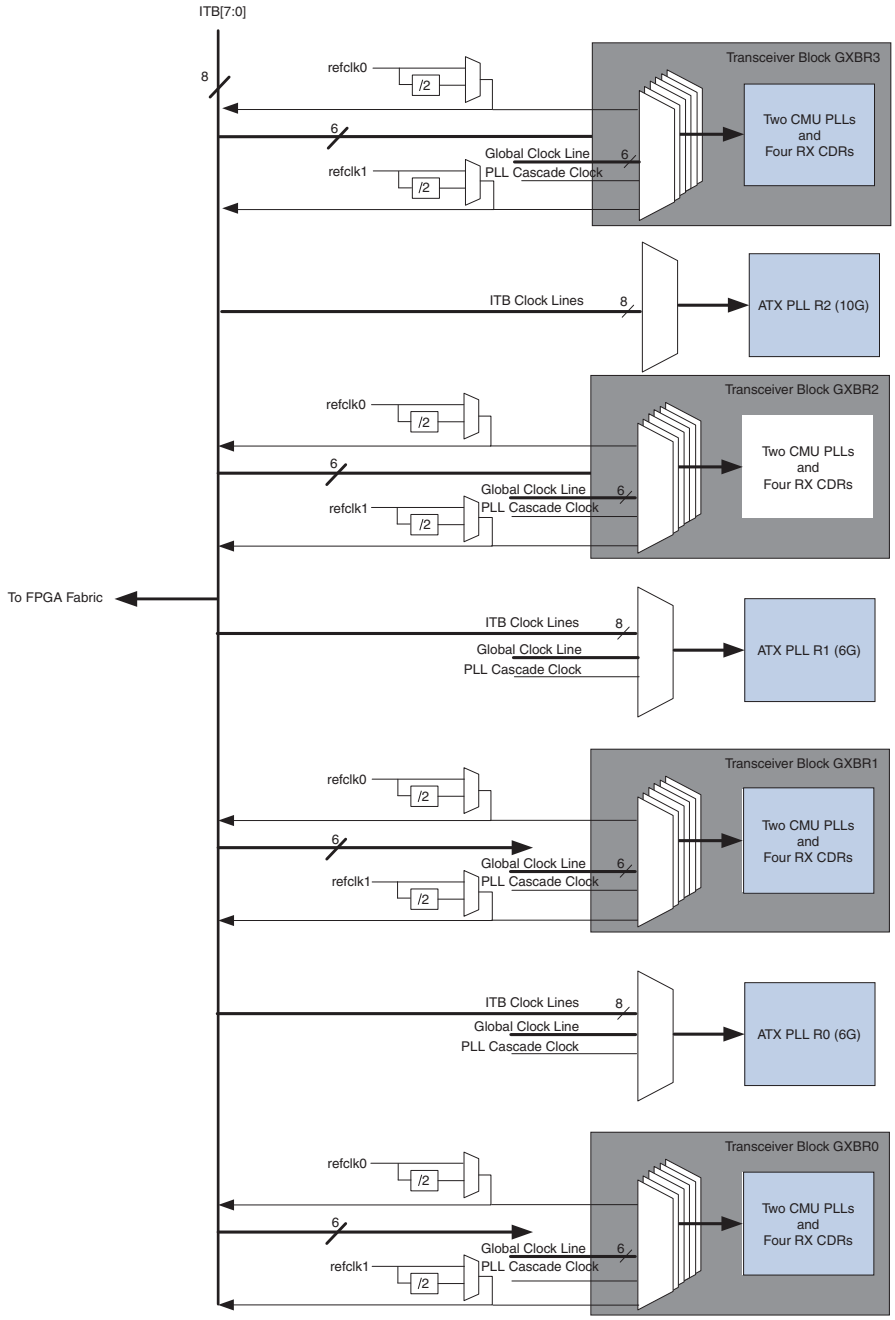


Figure 2–4 shows the input reference clock sources for CMU PLLs, ATX PLLs, and receiver CDRs in four transceiver blocks on the right side of the EP4S100G5F45 device. In this figure, the input reference clock sources for four transceiver blocks are located only on the right side of the EP4S100G5F45 device but the device has similar input reference clock resources available for the four transceiver blocks located on the left side of the device as well.

Figure 2–4 also shows the ITB clock lines on the right side of the EP4S100G5F45 device. The number of ITB clock lines available in any Stratix IV GT device is equal to the number of `refclk` pins available in that device.

Figure 2–4. Input Reference Clock Sources Across Transceiver Blocks for Stratix IV GT Devices



refclk0 and refclk1 Pins

Each transceiver block has two dedicated `refclk` pins that you can use to drive the CMU PLL, receiver CDR, or both, input reference clocks. Each of the two CMU PLLs and four receiver CDRs within a transceiver block can derive its input reference clock from either the `refclk0` or `refclk1` pin.

The `refclk` pins provide the cleanest input reference clock path to the CMU/ATX PLLs when compared with other input reference clock sources. Altera recommends using the `refclk` pins to drive the CMU PLL input reference clock for improved transmitter output jitter performance.

Table 2–4 lists the electrical specifications for the input reference clock signal driven on the `refclk` pins.

For specifications regarding the input frequency supported by the `refclk` pins, refer to the *DC and Switching Characteristics for Stratix IV Devices* chapter.

Table 2–4. Electrical Specifications for the Input Reference Clock

Protocol	I/O Standard	Coupling	Termination
■ GIGE ■ XAUI ■ Serial RapidIO® ■ SONET/SDH ■ SDI ■ (OIF) CEI PHY Interface ■ Basic	■ 1.2-V PCML, 1.4 PCML ■ 1.4-V PCML ■ 1.5-V PCML ■ 2.5-V PCML ■ Differential LVPECL ■ LVDS	AC	On-chip (2)
PCI Express® (PCIe)	■ 1.2-V PCML, 1.4 PCML ■ 1.4-V PCML ■ 1.5-V PCML ■ 2.5-V PCML ■ Differential LVPECL ■ LVDS	AC	On-chip (2)
	■ HCSL *(1)*	DC	Off-chip (3)

Notes to Table 2–4:

(1) 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 configured the transceiver in PCIe functional mode. For more information, refer to Figure 2–5 on page 2–8.

(2) Termination values supported are the same as the Receiver pin differential on-chip termination resistors specified in the *DC and Switching Characteristics for Stratix IV Devices* chapter.

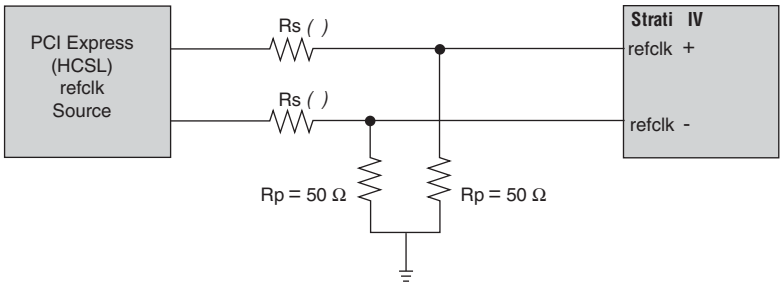
(3) For an example termination scheme, refer to Figure 2–5 on page 2–8.

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

```
set_instance_assignment -name INPUT_TERMINATION OFF -to <refclk_pin_name>
```

Figure 2–5 shows an example termination scheme for a reference clock signal when configured as HCSL.

Figure 2–5. Termination Scheme for a Reference Clock Signal When Configured as HCSL ***(Note 1)***



Notes to Figure 2–5:

(1) No biasing is required if the reference clock signals are generated from a clock source that conforms to the PCIe specification.

(2) Select resistor values as recommended by the PCIe clock source vendor.

Inter-Transceiver Block (ITB) Clock Lines

The `refclk0` and `refclk1` pins of other transceiver blocks using the ITB clock lines provide an input reference clock path from the `refclk` pins of one transceiver block to the CMU PLLs and receiver CDRs of the other transceiver blocks. In designs that have channels located in different transceiver blocks, the ITB clock lines eliminate the need to connect the on-board reference clock crystal oscillator to the `refclk` pin of each transceiver block. The ITB clock lines also drive the clock signal on the `refclk` pins to the clock logic in the FPGA fabric.

The ITB clock lines also provide an input reference clock path from the `refclk` pins of any transceiver block to the ATX PLLs located on the same side of the device.

Each `refclk` pin drives one ITB clock line for a total of up to eight ITB clock lines on each of the right and left sides of the device, as shown in Figure 2–3 on page 2–5.

The ITB clock lines provide input reference clock paths from the `refclk` pins of one transceiver block to the CMU PLLs and receiver CDRs of other transceiver blocks located on the same side of the device.

Dedicated CLK Input Pins on the FPGA Global Clock Network

Stratix IV devices provide up to eight differential clock input pins located in non-transceiver I/O banks that you can use to provide up to eight input reference clocks to the transceiver blocks. The Quartus® II software automatically chooses the global clock network to route the input reference clock signal from the CLK pins to the transceiver blocks.

For more information, refer to the "Dedicated Clock Input Pins" section in the *Clock Networks and PLLs in Stratix IV Devices* chapter.

One global clock resource is available for each CMU PLL, 6G ATX PLL, and receiver CDR. This allows each CMU PLL, 6G ATX PLL, and receiver CDR to derive its input reference clock from a separate FPGA CLK input pin.

Clock Output from Left and Right PLLs in the FPGA Fabric

You can use the synthesized clock output from one of the left or right PLLs to provide the input reference clock to the CMU PLLs, 6G ATX PLLs, and receiver CDRs. Stratix IV devices provide a dedicated clock path from the left PLLs (`PLL_L1`, `PLL_L2`, `PLL_L3`, and `PLL_L4`) in the FPGA fabric to the PLL cascade network located on the left side of the device. Stratix IV devices also provide a dedicated clock path from the right PLLs (`PLL_R1`, `PLL_R2`, `PLL_R3`, and `PLL_R4`) in the FPGA fabric to the PLL cascade network located on the right side of the device. The additional clock multiplication factors available in the left and right PLLs allow more options for on-board crystal oscillator frequencies.

FPGA Fabric PLLs-Transceiver PLLs Cascading

The CMU PLL synthesizes the input reference clock to generate the high-speed serial clock used in the transmitter PMA. The receiver CDR synthesizes the input reference clock in lock-to-reference (LTR) mode to generate the high-speed serial clock.

This high-speed serial clock output from the CMU PLL and the receiver CDR runs at a frequency that is half the configured data rate. The CMU PLLs and receiver CDRs only support multiplication factors (M) of 2, 4, 5, 8, 10, 16, 20, and 25. If you use an on-board crystal oscillator to provide the input reference clock through the dedicated `refclk` pins or ITB lines, the allowed crystal frequencies are limited by the CMU PLL and the receiver CDR multiplication factors. The input reference clock frequencies are also limited by the allowed phase frequency detector (PFD) frequency range.

Example 1: Channel Configuration with a 4 Gbps Data Rate

Consider a channel configured for a 4 Gbps data rate. The high-speed serial clock output from the CMU PLL and the receiver CDR must run at 2 Gbps. Table 2–5 lists the allowed input reference clock frequencies for Example 1.

Table 2–5. Allowed Input Reference Clock Frequency for Example 1

Multiplication Factor (M)	On-Board Crystal Reference Clock Frequency (MHz)		Allowed
	With /N = 1	With /N = 2	
2	1000	2000	No. Violates the PFD frequency limit.
4	500	1000	No. Violates the PFD frequency limit.
5	400	800	Yes but only for /N = 1.
8	250	500	Yes
10	200	400	Yes
16	125	250	Yes
20	100	200	Yes
25	80	160	Yes

For a 4 Gbps data rate, the Quartus II software only allows an input reference clock frequency of 80, 100, 125, 160, 200, 250, 400, and 500 MHz. To overcome this limitation, Stratix IV devices allow the synthesized clock output from the left and right PLLs in the FPGA fabric to drive the CMU PLL and receiver CDR input reference clock. The additional clock multiplication factors available in the left and right PLLs allow more options for on-board crystal oscillator frequencies.

Dedicated Left and Right PLL Cascade Network

Stratix IV devices have a dedicated PLL cascade network on the left and right side of the device that connects to the input reference clock selection multiplexer of the CMU PLLs, 6G ATX PLLs, and receiver CDRs on the left and right side of the device, respectively.

The dedicated PLL cascade networks are segmented by bidirectional tri-state buffers located along the clock line. Segmentation of the dedicated PLL cascade network allows two or more left and right PLLs to drive the cascade clock line simultaneously.

Because the number of left and right PLLs and transceiver blocks vary from device to device, the capability of cascading a left and right PLL to the CMU PLLs, 6G ATX PLLs, and receiver CDRs also varies from device to device.

The following sections describe the Stratix IV GX and GT FPGA fabric-Transceiver PLLs cascading for the various device packages.

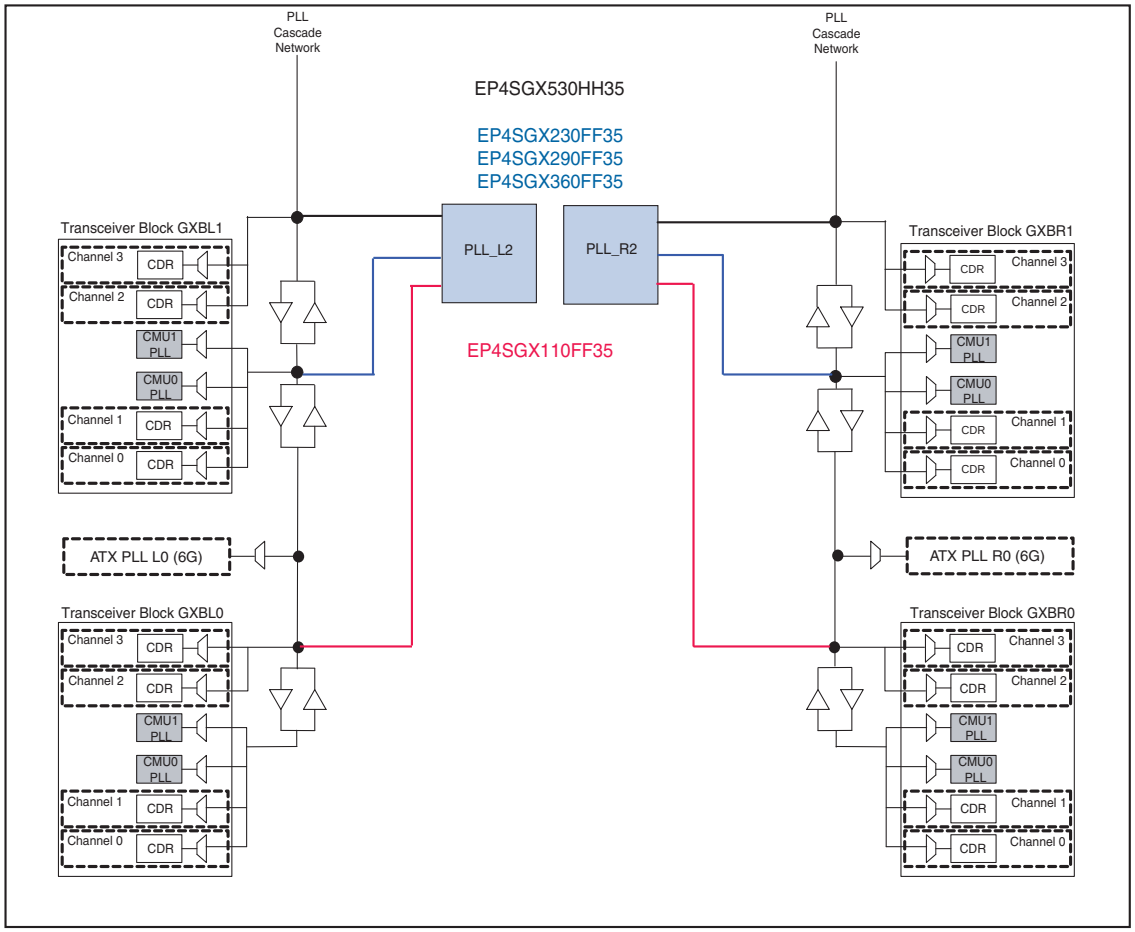
FPGA Fabric PLLs-Transceiver PLLs Cascading in the 780-Pin Package

Stratix IV GX devices in 780-pin packages do not support FPGA fabric PLLs-transceiver PLLs cascading.

FPGA Fabric PLLs-Transceiver PLLs Cascading in the 1152-Pin Package

Figure 2–6 shows the FPGA fabric PLLs-Transceiver PLLs cascading options allowed in the EP4SGX110FF35 device (red), the EP4SGX230FF35, EP4SGX290FF35 and EP4SGX360FF35 devices (blue), and the EP4SGX530HH35 device (black).

Figure 2–6. FPGA Fabric PLLs-Transceiver PLLs Cascading Options Allowed for 1152-Pin Package Devices

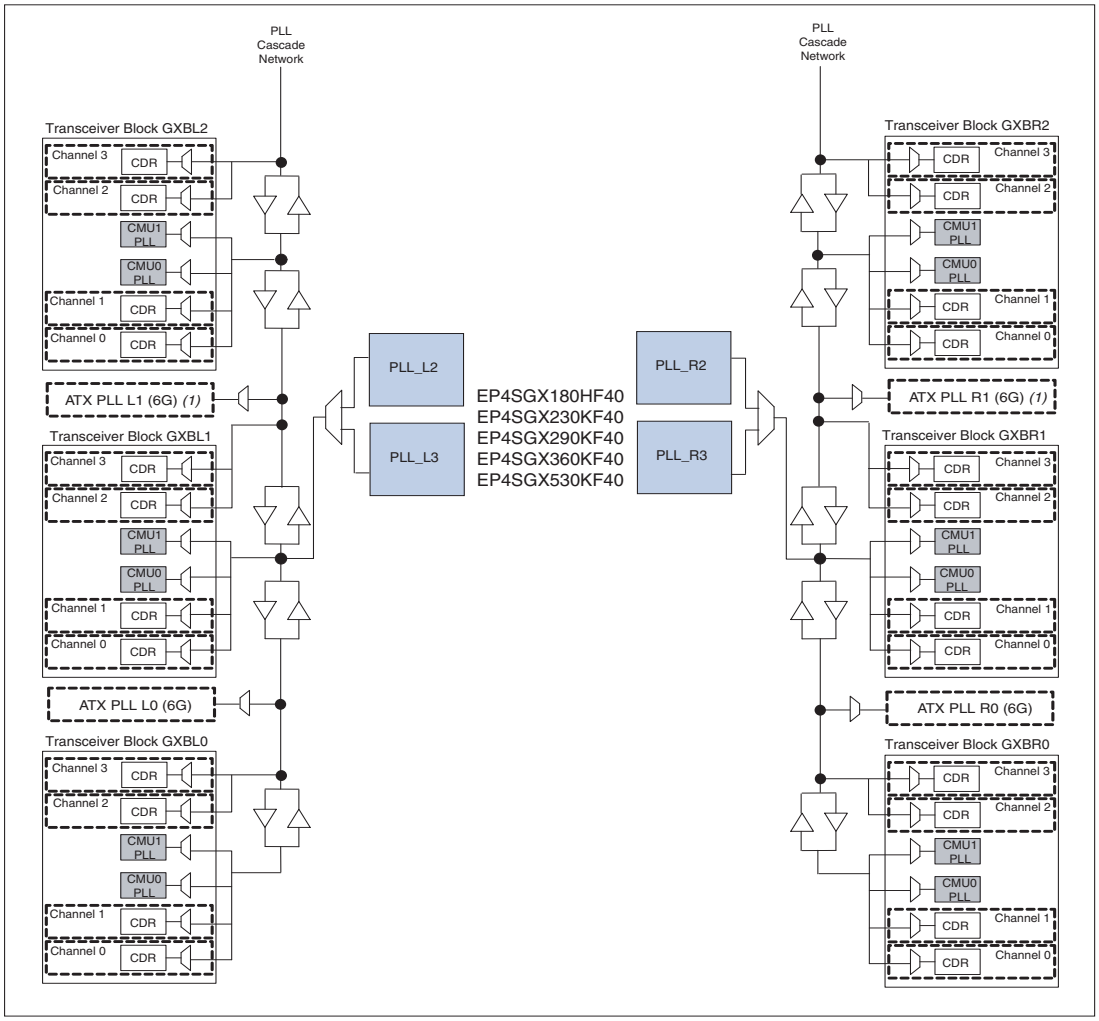


FPGA Fabric PLLs-Transceiver PLLs Cascading in the 1517-Pin Package

Figure 2–7 shows the FPGA fabric PLLs-Transceiver PLLs cascading options allowed in the EP4SGX180HF40, EP4SGX230KF40, EP4SGX290KF40, EP4SGX360KF40, and EP4SGX530KF40 devices.

For Stratix IV GT devices, FPGA fabric PLLs-Transceiver PLLs cascading is not supported for the 10G ATX PLLs. For the EP4S40G2KF40, EP4S40G5KF40, EP4S100G2KF40, and EP4S100G5KF40 devices, FPGA fabric PLLs-Transceiver PLLs cascading for the 6G ATX PLLs and CMU PLLs is the same as the Stratix IV GX devices in the 1517-pin package.

Figure 2–7. FPGA Fabric PLLs-Transceiver PLLs Cascading Options Allowed in the 1517-Pin Package Devices



Note to Figure 2–7:

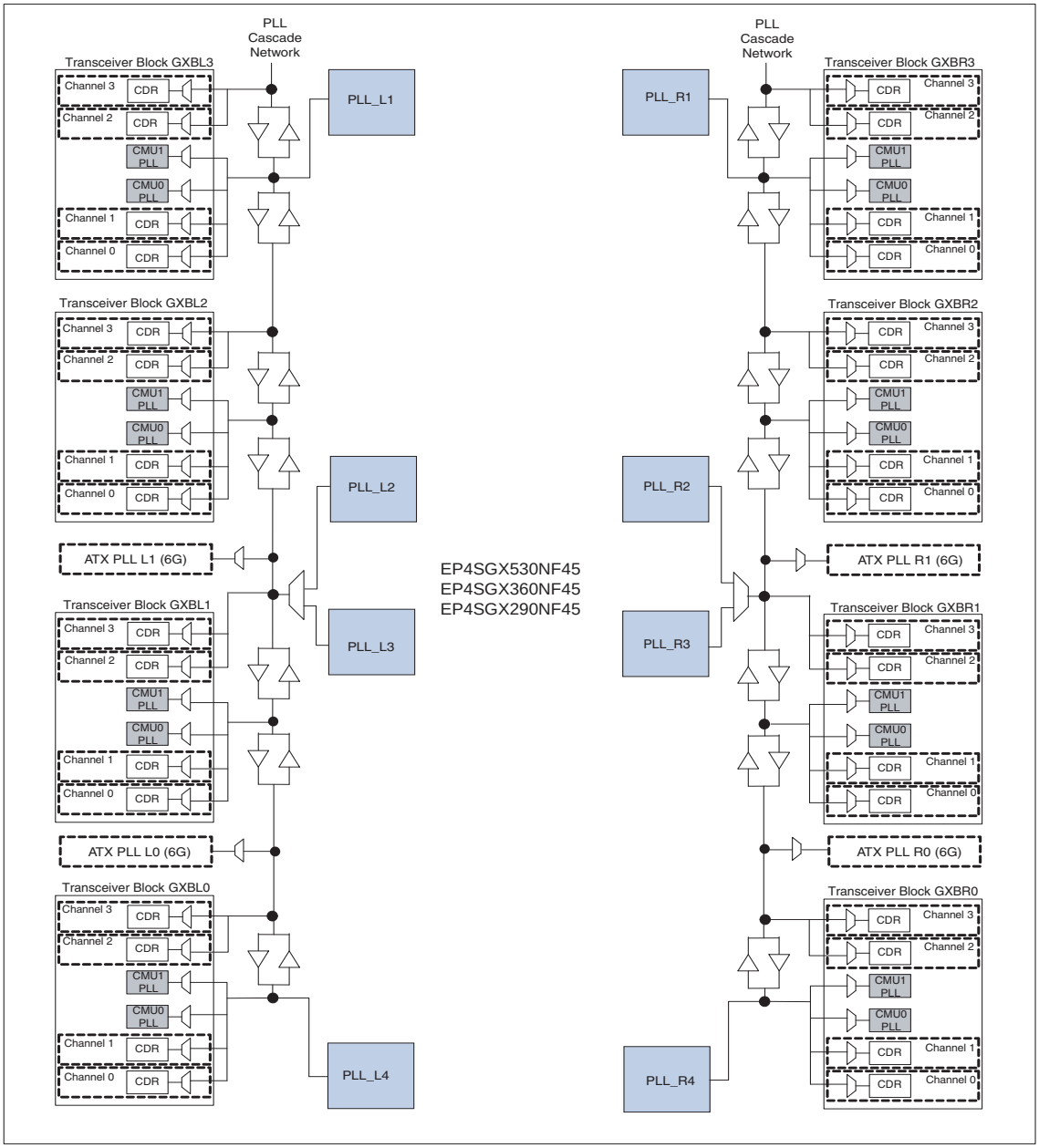
(1) ATX PLL L1 and ATX PLL R1 are not present in the EP4SGX230KF40 device.

FPGA Fabric PLLs-Transceiver PLLs Cascading in the 1932-Pin Package

Figure 2–8 shows the FPGA fabric PLLs-Transceiver PLLs cascading options allowed in the EP4SGX530NF45, EP4SGX360NF45, and EP4SGX290NF45 devices.

For Stratix IV GT devices, FPGA fabric PLLs-Transceiver PLLs cascading is not supported for the 10G ATX PLLs. For the EP4S100G3NF45, EP4S100G4N45, and EP4S100G5NF45 devices, FPGA fabric PLLs-Transceiver PLLs cascading for the 6G ATX PLLs and CMU PLLs is the same as the Stratix IV GX devices in the 1932-pin package.

Figure 2–8. FPGA Fabric PLLs-Transceiver PLLs Cascading Options Allowed in the 1932-Pin Package Device



FPGA Fabric PLLs-Transceiver PLLs Cascading Rules

You can only cascade the left PLLs (PLL_L1, PLL_L2, PLL_L3, and PLL_L4) to the transceiver blocks located on the left side of the device. Similarly, you can only cascade the right PLLs (PLL_R1, PLL_R2, PLL_R3, and PLL_R4) to the transceiver blocks located on the right side of the device.

The PLL cascade networks are single clock lines segmented by bidirectional tri-state buffers located along the clock line. Segmentation of the PLL cascade network allows two left and right PLLs to drive the cascade clock line simultaneously and provides the input reference clock to the CMU PLLs and receiver CDRs in different transceiver blocks. When cascading two or more FPGA fabric PLLs to the CMU PLLs and receiver CDRs, there must be no crossover in the cascaded clock paths on the PLL cascade network (Figure 2–9).

For better noise rejection, ensure the bandwidth setting of the FPGA fabric PLL (the upstream PLL) is lower than the transceiver PLL (the downstream PLL).

Example 2: Design Target—EP4SGX530NF45 Device

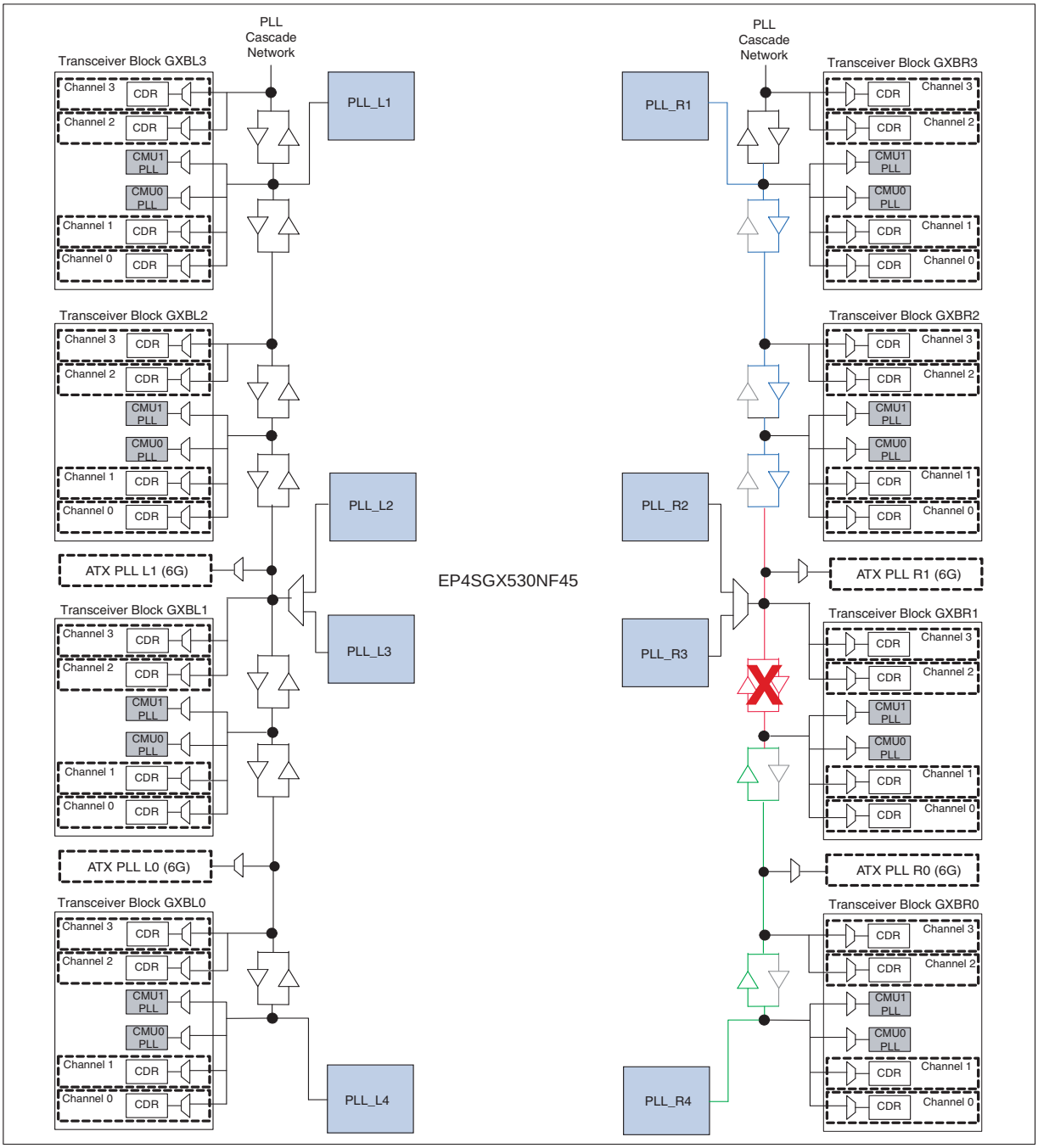
If your design is targeted for a EP4SGX530NF45 device, it requires providing input reference clocks to the following CMU PLLs and receiver CDRs from two right PLLs in the FPGA fabric:

- CMU0 PLL in Transceiver Block GXBR1
- Receiver CDRs in channel 2 and channel 3 in Transceiver Block GXBR1

Case 1: use PLL_R4 to provide the input reference clock to the receiver CDRs in channel 2 and channel 3 (shown in GREEN) and use PLL_R1 to provide the input reference clock to the CMU0 PLL (shown in BLUE) in transceiver block GXBR1.

Figure 2–9 shows that this FPGA fabric-Transceiver PLL cascading configuration is illegal due to crossover (shown in RED) of the cascade clock paths on the PLL cascade network.

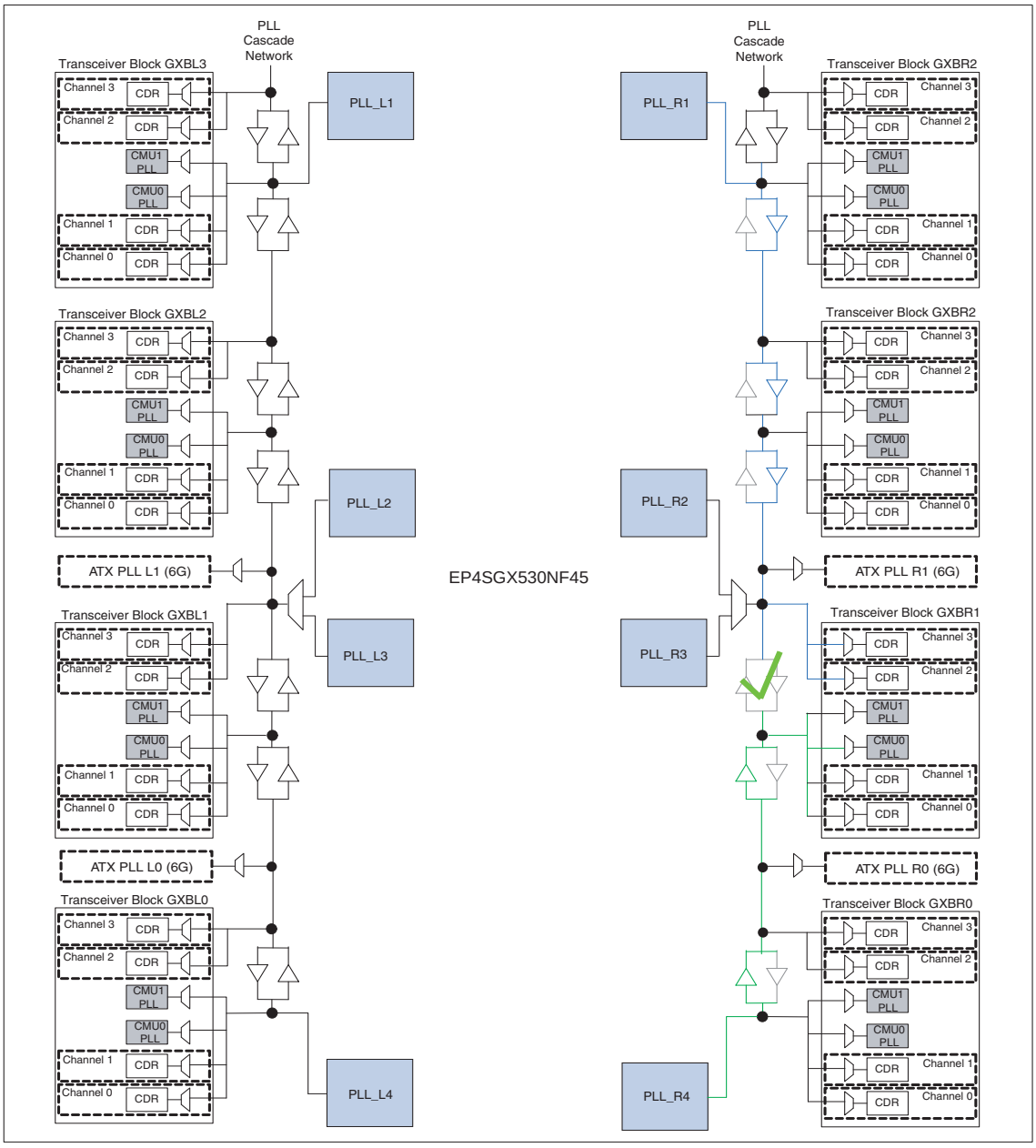
Figure 2–9. Illegal FPGA Fabric-Transceiver PLL Cascading Configuration



Case 2: use PLL_R1 to provide the input reference clock to the receiver CDRs in channel 2 and channel 3 (shown in BLUE) and use PLL_R4 to provide the input reference clock to the CMU0 PLL (shown in GREEN) in transceiver block GXBR1.

Figure 2–10 shows that this FPGA fabric-Transceiver PLL cascading configuration is legal as there is no crossover of the cascade clock paths on the PLL cascade network.

Figure 2–10. Legal FPGA Fabric-Transceiver PLL Cascading Configuration



Left and Right, Left, or Right PLL in VCO Bypass Mode

If all CMU channels on the same side of the device are configured as channels, all `refclk` pins are used as receiver serial input data pins. All CMU PLLs are also used as receiver CDRs. In such designs, you must use the 6G ATX PLLs to generate the high-speed serial and low-speed parallel transceiver clocks provided that the configured data rate is supported by the 6G ATX PLLs. Additionally, Altera recommends providing the input reference clock to the 6G ATX PLL using the left or right PLL cascade clock line because none of the `refclk` pins are available. To avoid jitter amplification because of cascading of the left or right PLL to the 6G ATX PLL, you must place the left or right PLL in VCO bypass mode. When in VCO bypass mode, you can only divide the reference clock by the N integer.

For more information about CMU PLLs, refer to "Configuring CMU Channels as Transceiver Channels" in the *Transceiver Architecture in Stratix IV Devices* chapter.

Figure 2–11 shows that in VCO bypass mode, the input reference clock from the dedicated FPGA CLK pins to the `inclk` port of the left and right, left, or right PLL bypasses the PLL loop and is driven directly on the PLL output clock port.

Figure 2–11. Left and Right, Left, or Right PLL in VCO Bypass Mode

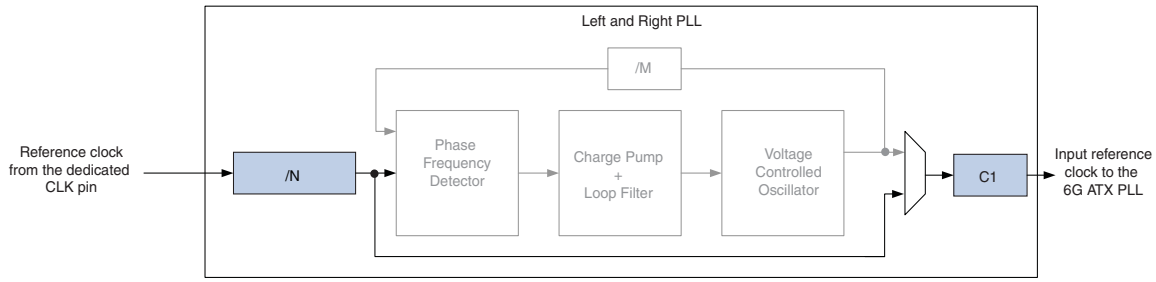
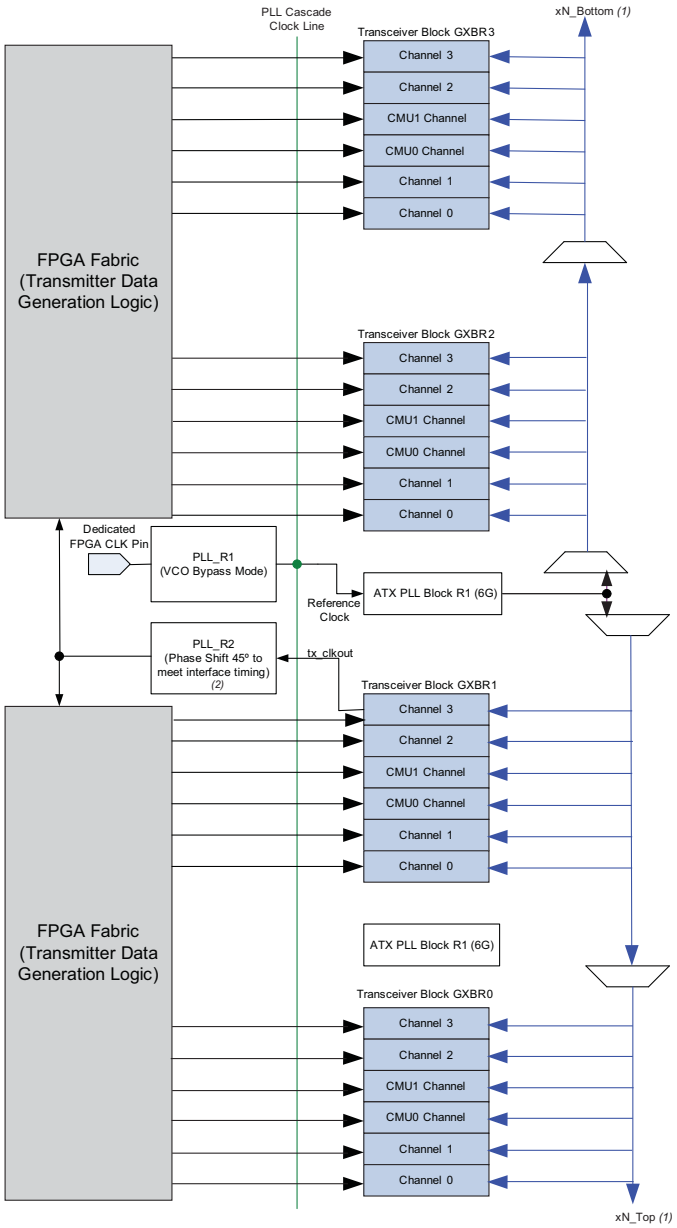


Figure 2–12 shows 24 channels on the right side of the EP4SGX530NF45 device configured in Basic (PMA Direct) ×N mode running at 6.5 Gbps with a 20-bit FPGA fabric-PMA interface width. Because all 24 channels on the right side of the device are configured in Basic (PMA Direct) ×N mode, use the right `PLL_R1` configured in VCO bypass mode to provide the input reference clock to the 6G ATX PLL.

Because the data rate of 6.5 Gbps requires a left and right, left, or right PLL to meet FPGA fabric-Transmitter PMA interface timing, the `tx_clkout` from one of the 24 channels is phase shifted using `PLL_R2`. Use the phase-shifted output clock from `PLL_R2` to clock the FPGA fabric logic that generates the transmitter parallel data and control signals.

Figure 2–12. Input Reference Clocking Using Left and Right, Left, or Right PLL in VCO Bypass Mode (3)



Notes to Figure 2–12:

(1) For more information, refer to "Transceiver Channel Datapath Clocking" on page 2–20.

(2) For more information, refer to *AN 580: Achieving Timing Closure in Basic (PMA Direct) Functional Mode*.

(3) The green line represents the PLL cascade clock line and the blue lines represent 6G ATX PLL R1.

For more information about configuring left or right PLLs in VCO bypass mode, refer to "Configuration Example 4: Configuring Left and Right, Left, or Right PLL in VCO Bypass Mode" on page 2–78.

Transceiver Channel Datapath Clocking

This section describes the transmitter channel and receiver channel datapath clocking in various configurations. Datapath clocking varies with physical coding sublayer (PCS) configurations in different functional modes as well as channel bonding options. This section contains:

- "Transmitter Channel Datapath Clocking" on page 2–20
- "Receiver Channel Datapath Clocking" on page 2–39

Clocking described in this section is internal to the transceiver and clock routing is primarily performed by the Quartus II software.

For more information about manually picking and placing CMU and ATX PLLs, refer to *AN 578: Manual Placement of CMU PLLs and ATX PLLs in Stratix IV GX and GT Devices*.

Transmitter Channel Datapath Clocking

This section describes the transmitter channel PMA and PCS datapath clocking in non-bonded and bonded channel configurations:

- "Non-Bonded Channel Configurations" on page 2–24
- "Bonded Channel Configurations" on page 2–27
- "Non-Bonded Basic (PMA Direct) Mode Channel Configurations" on page 2–34
- "Bonded Basic (PMA Direct) ×N Mode Channel Configurations" on page 2–36

Transmitter Channel-to-Channel Skew Optimization for Modes Other than Basic (PMA Direct) Mode

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. Transmitter datapath clocking is set up to provide low channel-to-channel skew when compared with non-bonded channel configurations.

- In bonded channel configurations—the high-speed serial clock and low-speed parallel clock for all bonded channels are generated by the `CMU0` clock divider or the ATX clock divider block, resulting in lower channel-to-channel clock skew.

  The transmitter phase compensation FIFO in all bonded channels (except in Basic [PMA Direct] ×N mode) share common pointers and control logic generated in the central control unit (CCU), 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 provides lower channel-to-channel skew in bonded channel configurations.

- In non-bonded channel configurations—the high-speed serial clock and low-speed parallel clock in each channel are generated independently by its local clock divider. This results in higher channel-to-channel clock skew.

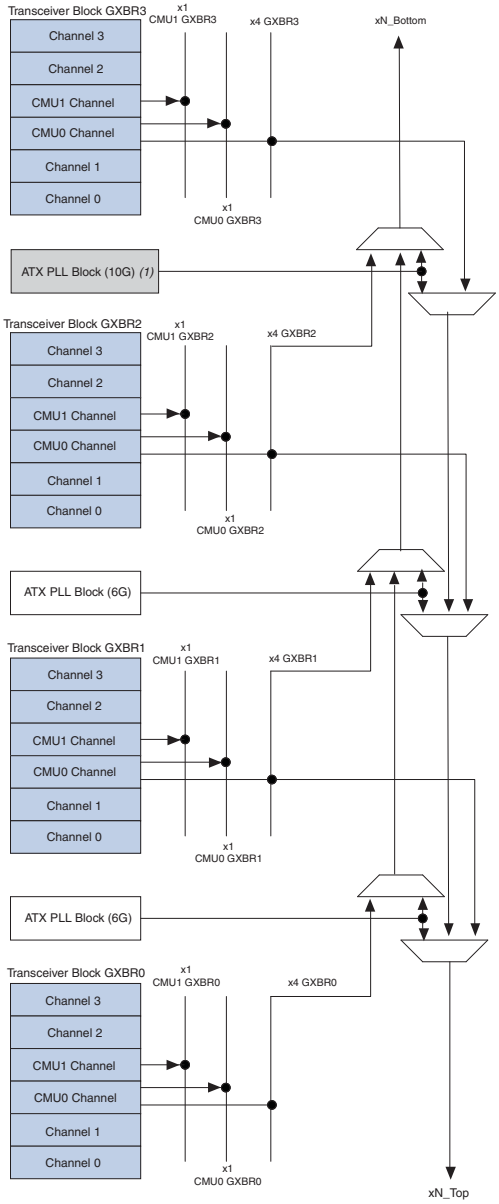
  The transmitter phase compensation FIFO in each non-bonded channel (except in Basic [PMA Direct] mode) has its own pointers and control logic that can result in unequal latency in the transmitter phase compensation FIFO of each channel. The higher transceiver clock skew and unequal latency in the transmitter phase compensation FIFO in each channel can result in higher channel-to-channel skew.

Transmitter Channel Datapath Clocking Resources

The Stratix IV transceivers support various non-bonded and bonded transceiver clocking configurations through the dedicated ×1, ×4, and ×N high-speed serial and low-speed parallel clock lines.

Figure 2–13 shows the transceiver clock distribution in ×1, ×4, ×8, and ×N bonded modes.

Figure 2–13. Transceiver Clock Distribution in the Stratix IV GT EP4S100G5F45 and Stratix IV GX EP4SGX530KF40 Devices



Note to Table 2–14:

(1) The 10G ATX PLL block is not available for the EP4SGX530KF40 device.

Non-bonded and bonded configurations use the following:

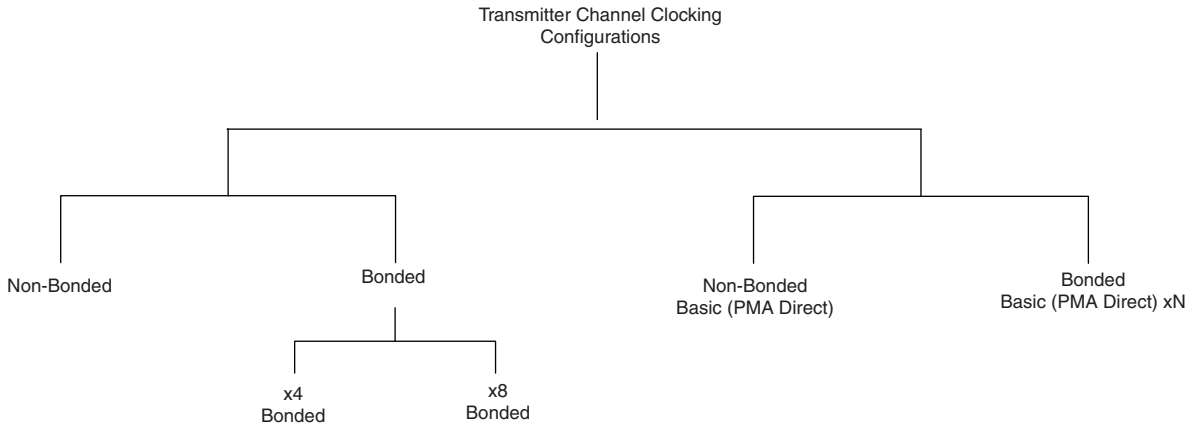
- ×1 non-bonded configurations use the ×1 clock lines to distribute only the high-speed serial transceiver clock synthesized by the CMU0 PLL or CMU1 PLL to the clock transmitter channels located in the same transceiver block. The low-speed parallel transceiver clock is generated in the transceiver channels using the local clock dividers.
- ×4 bonded configurations use the ×4_GXB clock lines to distribute both the high-speed serial and low-speed parallel transceiver clocks generated by the CMU0_Channel to clock the bonded transmitter channels located in the same transceiver block.
- ×8 and ×N bonded configurations use the ×N_Top or ×N_Bottom clock lines to distribute both the high-speed serial and low-speed parallel transceiver clocks generated by the CMU0 channel block to all bonded transmitter channels located across transceiver blocks.

ATX PLLs always use ×N lines to distribute the high-speed serial and low-speed parallel transceiver clocks. Use the xN_Top line if the CMU0 PLL or ATX PLL that generates the transceiver clocks is located at the top of the transmitter channel. Use the xN_Bottom line if the CMU0 PLL or ATX PLL is located at the bottom of the transmitter channel. Because there is only one xN_Top and xN_Bottom line on each side of the device, using an ATX PLL limits the use of the ×N clock lines to distribute the transceiver clocks to other transmitter channels in the design.

Transmitter Channel Clocking Configurations

Figure 2–14 shows various transmitter channel clocking configurations.

Figure 2–14. Transmitter Channel Clocking Configurations



Transmitter channels configured in modes other than Basic (PMA Direct) mode use both the transmitter channel PCS and PMA blocks. As a result, Stratix IV devices allow placing these transmitter channels only in the four regular channels of a transceiver block. Stratix IV devices do not allow configuring the CMU channels in any mode other than Basic (PMA Direct) mode because of the absence of PCS blocks in the CMU channels.

The transmitter channel datapath clocking in modes other than Basic (PMA Direct) mode depends on whether the transmitter channel is configured in non-bonded or bonded mode.

Non-Bonded Channel Configurations

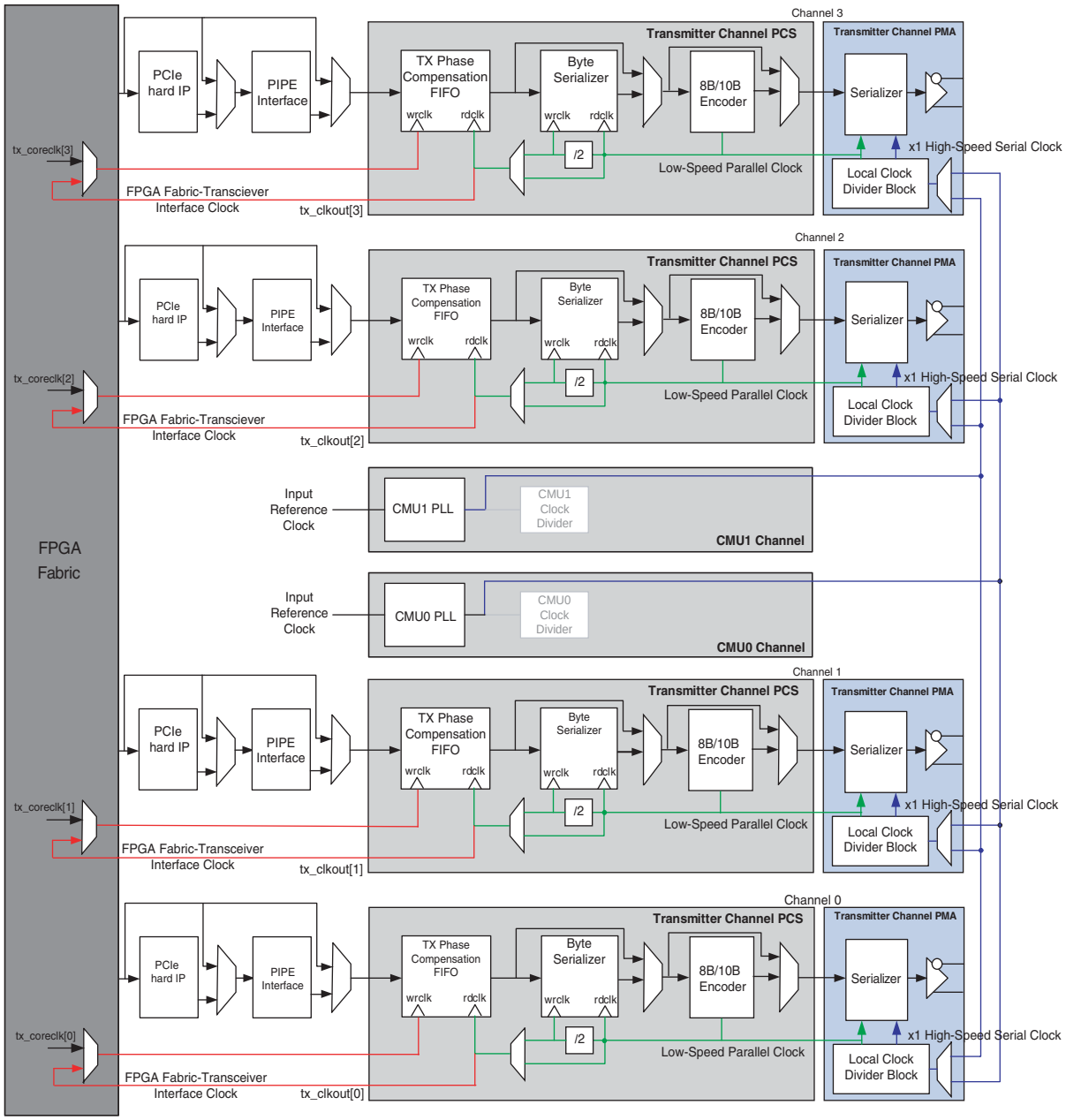
The following modes other than Basic (PMA Direct) functional mode have a non-bonded transmitter channel configuration:

- PCIe ×1—Gen1 and Gen2
- Gigabit Ethernet (GIGE)
- Serial RapidIO
- SONET/SDH
- SDI
- (OIF) CEI PHY Interface
- Basic (except Basic ×4 and Basic ×8 modes)
- Deterministic Latency

Use the CMU channels to generate transceiver clocks for all the non-bonded functional modes listed above. Additionally, you may use the ATX PLLs if the configured data rate falls within the ATX PLL data rate range specified in the *DC and Switching Characteristics for Stratix IV Devices* chapter.

Figure 2–15 shows transmitter channel datapath clocking in non-bonded channel configurations when clocked using the CMU PLLs.

Figure 2–15. Transmitter Datapath Clocking in a Non-Bonded Configuration Clocked by CMU PLLs (1)



Note to Figure 2–15:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, and the blue lines represent the ×1 high-speed serial clock.

In non-bonded channel configurations clocked by the CMU PLL, each channel can derive its clock independently from either CMU0 PLL or CMU1 PLL within the same transceiver block. The CMU PLL synthesizes the input reference clock to generate a clock that is distributed to the local clock divider block in each channel using the ×1 high-speed serial clock line. Depending on the configured functional mode, the local clock divider block in each channel generates the low-speed parallel clock and high-speed serial clock. The serializer in the transmitter channel PMA uses both the low-speed parallel clock and high-speed serial clock for its parallel-in-serial-out operation. The low-speed parallel clock clocks the 8B/10B encoder (if enabled) and the read port of the byte serializer (if enabled) in the transmitter channel PCS.

Depending on whether you use the byte serializer or not, the low-speed parallel clock (when you do not use the byte serializer) or a divide-by-two version of the low-speed parallel clock (when you use the byte serializer) from the local clock divider block clocks the read port of the transmitter phase compensation FIFO in all four bonded channels. This clock is driven directly on the tx_clkout port as the FPGA fabric-Transceiver interface clock. You can use the coreclkout signal to clock the transmitter data and control logic in the FPGA fabric for all four bonded channels.

If you configure the ATX PLL to clock the transmitter channel, the ATX PLL block drives the high-speed serial clock and low-speed parallel clock to the transmitter channel on the ×N_Top or ×N_Bottom lines.

For more information, refer to the *Configuring Multiple Protocols and Data Rates in Stratix IV Devices* chapter.

Table 2–2 lists the transmitter channel datapath clock frequencies in non-bonded functional modes that have a fixed data rate.

Table 2–6. Transmitter Channel Datapath Clock Frequencies in Non-Bonded Functional Modes

Functional Mode	Data Rate	High-Speed Serial Clock Frequency	Low-Speed Parallel Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Serializer (MHz)	With Byte Serializer (MHz)
PCIe ×1 (Gen 1)	2.5 Gbps	1.25 GHz	250	250	125
PCIe ×1 (Gen 2)	5 Gbps	2.5 GHz	500	N/A	250
GIGE	1.25 Gbps	625 MHz	125	125	N/A
Serial RapidIO	1.25 Gbps	625 MHz	125	N/A	62.5
	2.5 Gbps	1.25 GHz	250	N/A	125
	3.125 Gbps	1.5625 GHz	312.5	N/A	156.25
SONET/SDH OC12	622 Mbps	311 MHz	77.75	77.75	N/A
SONET/SDH OC48	2.488 Gbps	1.244 GHz	311	N/A	155.5
HD-SDI	1.485 Gbps	742.5 MHz	148.5	148.5	74.25
	1.4835 Gbps	741.75 MHz	148.35	148.35	74.175
3G-SDI	2.97 Gbps	1.485 GHz	297	N/A	148.5
	2.967 Gbps	1.4835 GHz	296.7	N/A	148.35

Bonded Channel Configurations

In PCS and PMA bonded channel configurations, the PCS and PMA blocks of all bonded channels are clocked by the same low-speed parallel clock and high-speed serial clock from the CMU0 clock divider or the ATX PLL block. The phase compensation FIFOs of all bonded channels also share common read and write pointers and enable signals generated in the CCU.

Stratix IV devices support ×4 PCS and PMA channel bonding that allows bonding of four channels within the same transceiver block. Stratix IV devices also support ×8 channel bonding that allows bonding of eight PCS and PMA channels across two transceiver blocks on the same side of the device.

×4 PCS and PMA Bonded Channel Configuration

The following functional modes support ×4 PCS and PMA bonded transmitter channel configuration:

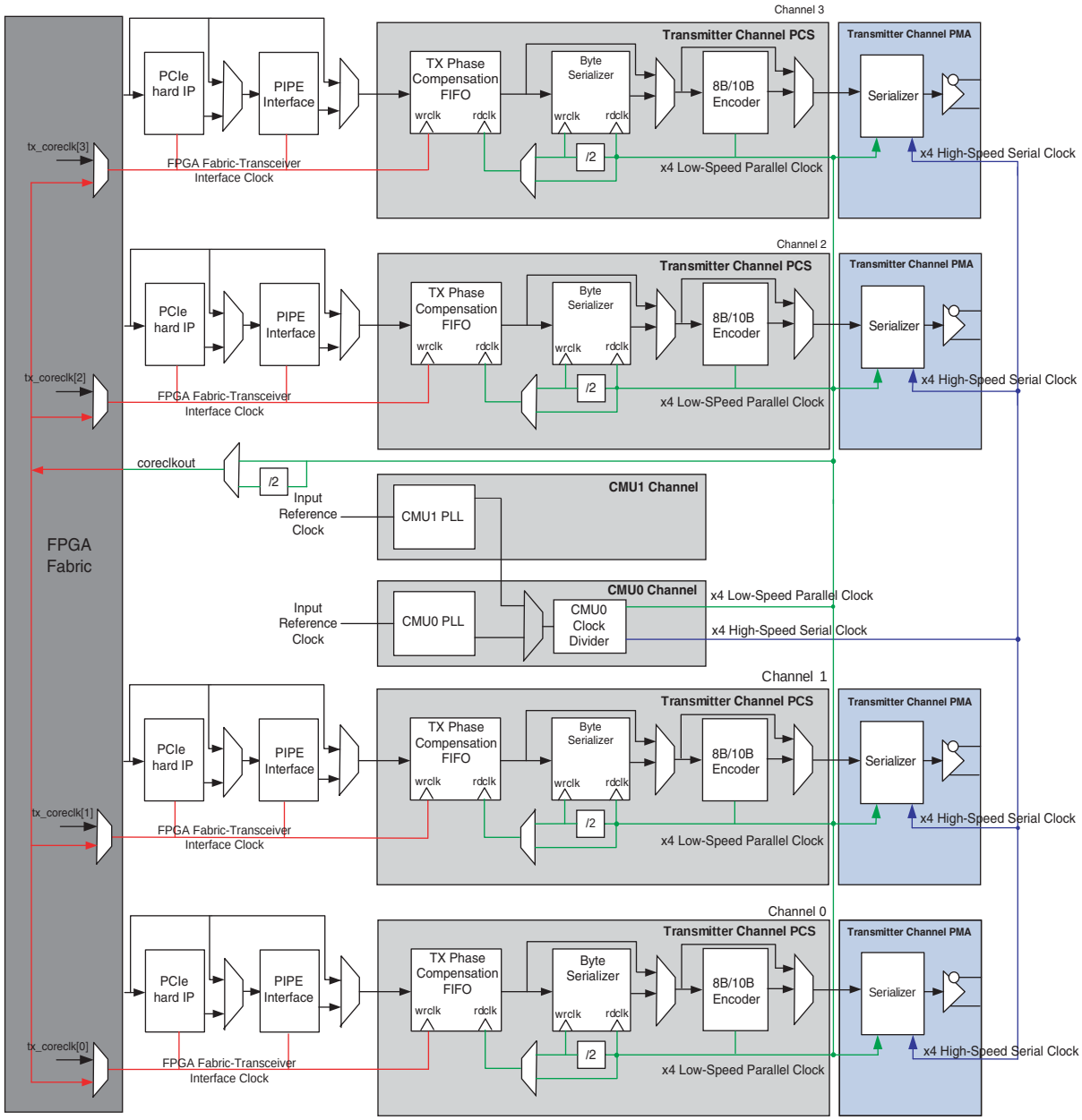
- PCIe ×4—Gen1 and Gen2
- XAUI
- Basic ×4

Use the CMU channels to generate the transceiver clocks for all ×4 bonded functional modes listed above. Additionally, you may use the ATX PLLs to generate the transceiver clocks for PCIe ×4 Gen 2 and Basic ×4 functional mode.

You must assign tx_dataout[0] of the ×4 bonded link (XAUI or PCIe ×4) to physical channel 0 of the transceiver block, tx_dataout[1] to physical channel 1 of the transceiver block, tx_dataout[2] to physical channel 2 of the transceiver block, and tx_dataout[3] to physical channel 3 of the transceiver block. Otherwise, the Quartus II compilation errors out.

Figure 2–16 shows the transmitter channel datapath clocking in ×4 channel bonding configurations when clocked using the CMU0 channel.

Figure 2–16. Transmitter Datapath Clocking in x4 Bonded Configurations (1)



Note to Figure 2–16:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, and the blue lines represent the high-speed serial clock.

The transceiver clocks are distributed to the four bonded channels on the ×4 high-speed serial and ×4 low-speed parallel clock lines. The serializer in the transmitter channel PMA of the four bonded channels uses the same low-speed parallel clock and high-speed serial clock from `CMU0 Channel` for their parallel-in-serial-out operation. The low-speed parallel clock clocks the 8B/10B encoder and the write port of the byte serializer (if enabled) in the transmitter channel PCS.

Depending on whether the you use the byte serializer or not, the low-speed parallel clock (when you do not use the byte serializer) or a divide-by-two version of the low-speed parallel clock (when you use the byte serializer) from the `CMU0` clock divider block clocks the read port of the transmitter phase compensation FIFO in all four bonded channels. This clock is driven directly on the `coreclkout` port as the FPGA fabric-Transceiver interface clock. You can use the `coreclkout` signal to clock the transmitter data and control logic in the FPGA fabric for all four bonded channels.

The ATX PLL block drives the high-speed serial clock and low-speed parallel clock to the transmitter channels on the `×N_Top` or `×N_Bottom` lines.

For more information, refer to the *Configuring Multiple Protocols and Data Rates in Stratix IV Devices* chapter.

In ×4 PCS and PMA bonded channel configurations, the transmitter phase compensation FIFOs in all four bonded channels share common read and write pointers and enable signals generated in the `CMU0` channel of the transceiver block. This ensures equal transmitter phase compensation FIFO latency across all four bonded channels, resulting in low transmitter channel-to-channel skew.

Table 2–3 lists the transmitter datapath clock frequencies in ×4 bonded functional modes that have a fixed data rate.

Table 2–7. Transmitter Datapath Clock Frequencies in ×4 Bonded Functional Modes

Functional Mode	Data Rate (Gbps)	High-Speed Serial Clock Frequency (GHz)	Low-Speed Parallel Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Serializer (MHz)	With Byte Serializer (MHz)
PCIe ×4 (Gen 1)	2.5	1.25	250	250	125
PCIe ×4 (Gen 2)	5	2.5	500	N/A	250
XAUI	3.125	1.5625	312.5	N/A	156.25

×8 PCS and PMA Bonded Channel Configuration

The following functional modes support ×8 PCS and PMA bonded transmitter channel configuration:

- PCIe ×8—Gen1 and Gen2
- Basic ×8

Use either the CMU PLL or the ATX PLL to generate the transceiver clocks in Basic ×8 functional modes. Use the ATX PLL in PCIe ×8 Gen2 mode in order to meet the transmitter jitter compliance.

The eight bonded channels are located in two transceiver blocks, referred to as the master transceiver block and the slave transceiver block, with four channels each. When clocked using a CMU PLL, the CMU0 clock divider in CMU0 channel of the master transceiver block drives the high-speed serial clock and low-speed parallel clock on the xN_Top clock line. The serializer in the transmitter channel PMA of all eight bonded channels uses the same low-speed parallel clock and high-speed serial clock driven by the CMU0 channel of the master transceiver block on the xN_Top clock line. The low-speed parallel clock from CMU0 channel of the master transceiver block clocks the 8B/10B encoder and the write port of the byte serializer (if enabled) in the transmitter channel PCS of all eight channels.

Depending on whether you use the byte serializer or not, the low-speed parallel clock (when you do not use the byte serializer) or a divide-by-two version of the low-speed parallel clock (when you use the byte serializer) from the CMU0 clock divider block clocks the read port of the transmitter phase compensation FIFO in all eight bonded channels. This clock is driven directly on the coreclkout port as the FPGA fabric-Transceiver interface clock. You can use the coreclkout signal to clock the transmitter data and control logic in the FPGA fabric for all eight bonded channels.

If you choose the ATX PLL to generate the transceiver clocks for the ×8 bonded channels, Altera recommends placing the ATX PLL between the master and slave transceiver block to minimize transmitter channel-to-channel skew. In this configuration, the ATX PLL block drives the high-speed serial clock and low-speed parallel clock to the master transceiver block on the ×N_Bottom lines. It drives the high-speed serial clock and low-speed parallel clock to the slave transceiver block on the ×N_Top lines.

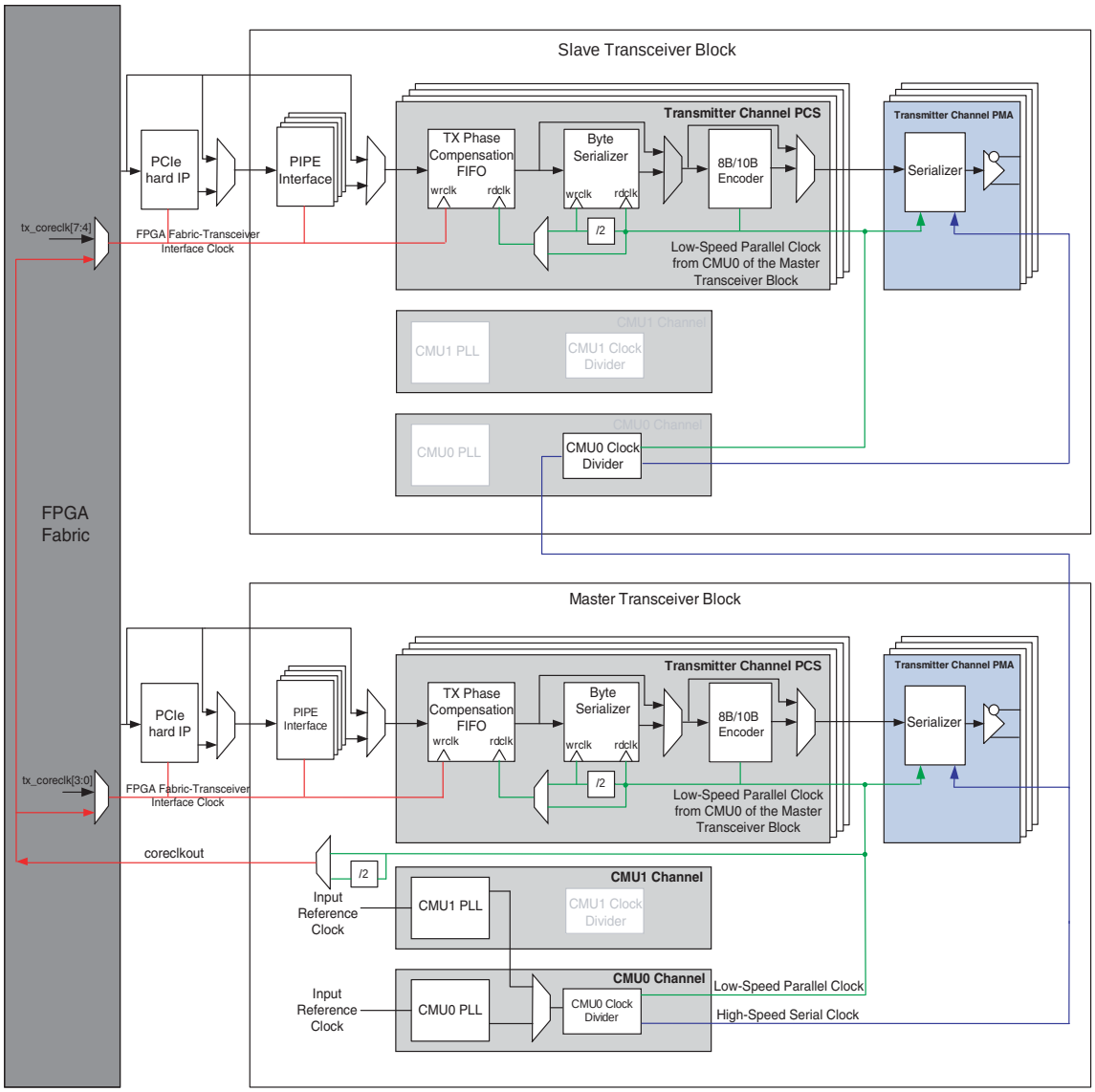
For more information, refer to the *Configuring Multiple Protocols and Data Rates in Stratix IV Devices* chapter.

In PCIe ×8 and Basic ×8 bonded channel configurations, the transmitter phase compensation FIFOs in all eight bonded channels share common read and write pointers and enable signals generated in the CCU of the master transceiver block. This ensures equal transmitter phase compensation FIFO latency across all eight bonded channels, resulting in low transmitter channel-to-channel skew.

The difference in clock routing delays between the ×4 clock lines and the ×N clock lines can result in higher transmitter channel-to-channel skew. To compensate for this difference in clock routing delays between the ×4 and the ×N clock lines, the Stratix IV transceivers introduce a fixed amount of delay in the ×4 clock lines of the transceiver block whose CMU0 channel generates the transceiver clocks in Basic ×8 bonded channel configuration.

Figure 2–17 shows the transmitter datapath clocking in PCIe ×8 channel bonding configurations when clocked using the CMU channel in the master transceiver block.

Figure 2–17. Transmitter Datapath Clocking in x8 Bonded Configuration (1)



Note to Figure 2–17:

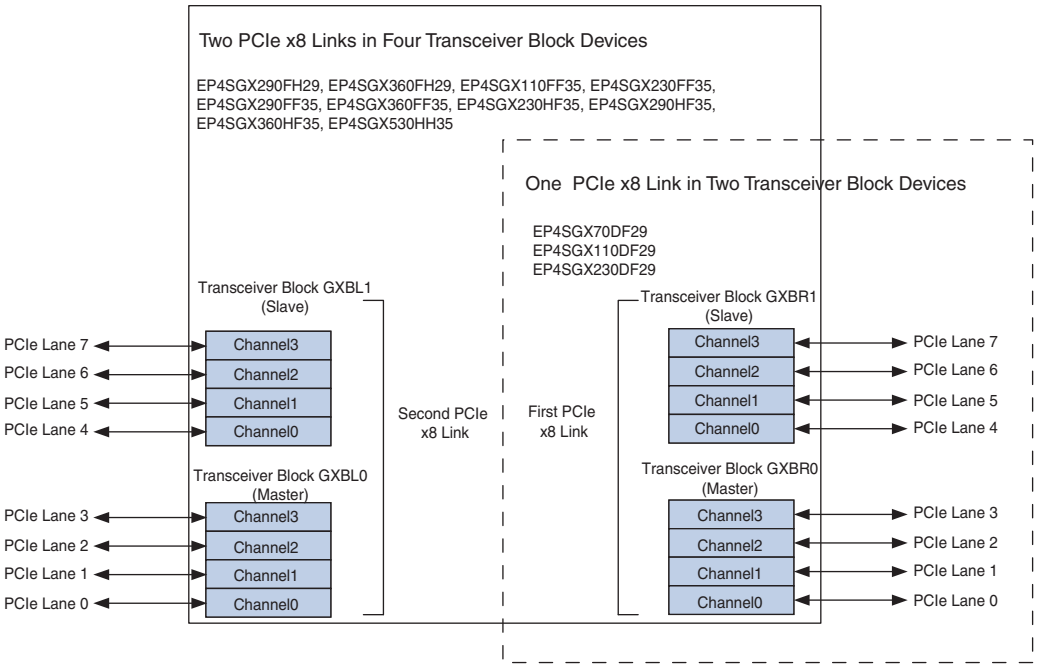
(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, and the blue lines represent the high-speed serial clock.

Figure 2–18 through Figure 2–20 show the allowed master and slave transceiver block locations and PCIe logical lane to physical transceiver channel mapping in all Stratix IV devices.

The Quartus II compilation errors out if you do not map the PCIe logical lanes to the physical transceiver channels, as shown in Figure 2–18 through Figure 2–20.

Figure 2–18 shows one PCIe ×8 link in two transceiver block devices and two PCIe ×8 links in four transceiver block devices.

Figure 2–18. One PCIe x8 Link in Two Transceiver Block Devices and Two PCIe x8 Links in Four Transceiver Block Devices (1)

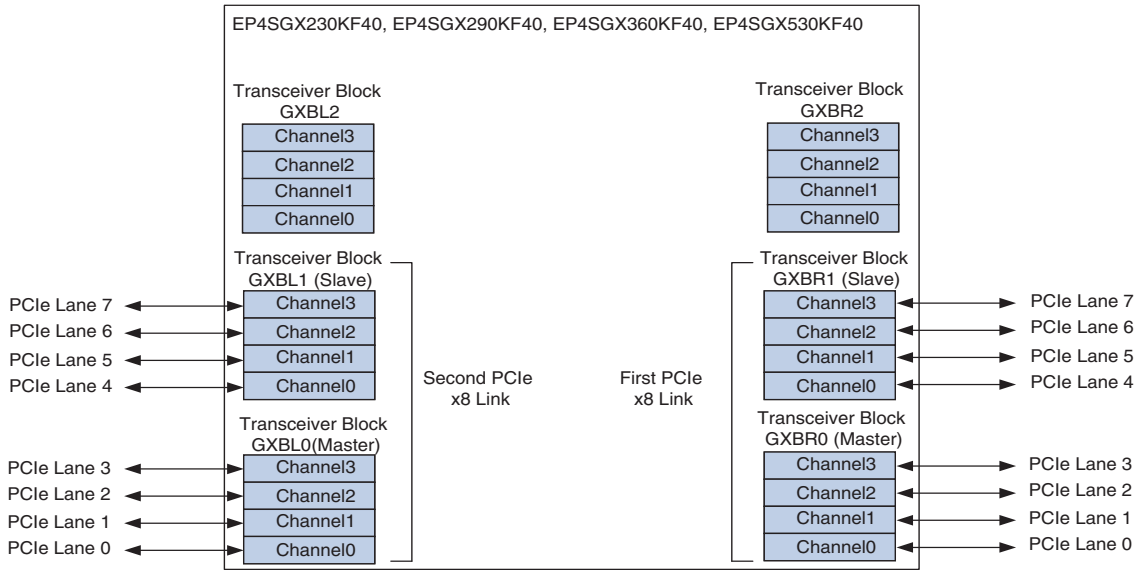


Note to Figure 2–18:

(1) You can use a ×4 PCIe configuration in either a master or slave block.

Figure 2–19 shows two PCIe ×8 links in six transceiver block devices.

Figure 2–19. Two PCIe ×8 Links in Six Transceiver Block Devices ***(Note 1),*** ***(2)***



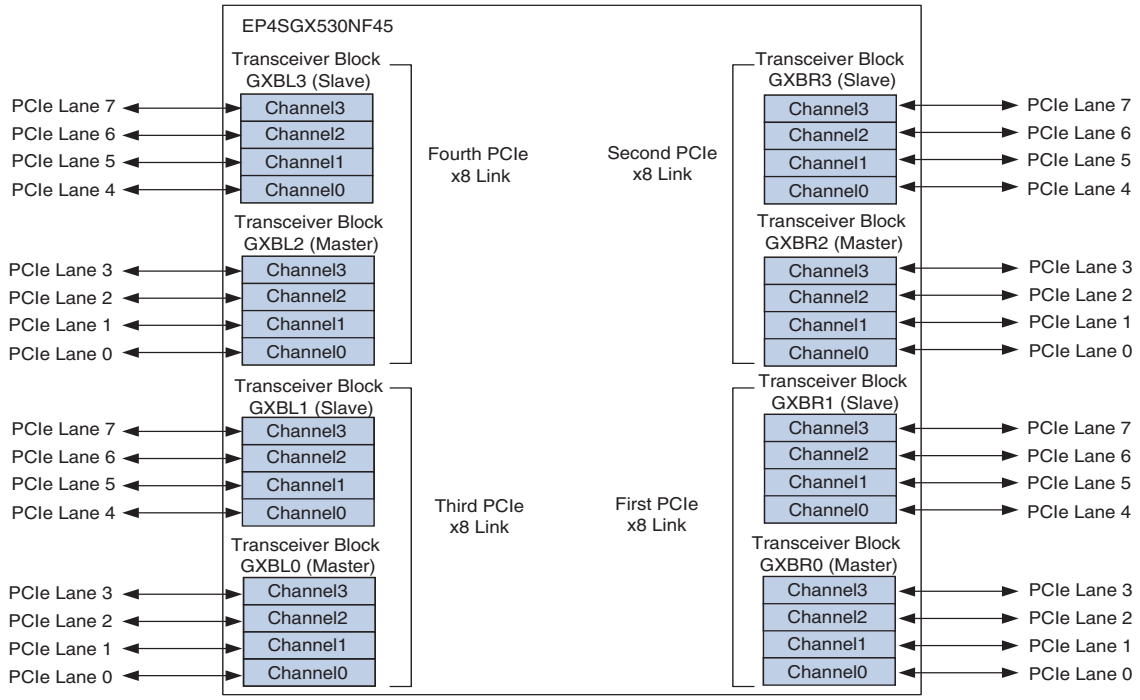
Notes to Figure 2–19:

(1) Stratix IV devices with six transceiver blocks allow a maximum of two PCIe ×8 links occupying four transceiver blocks. You can configure the other two transceiver blocks to implement other functional modes.

(2) You can use a ×4 PCIe configuration in either a master or slave block.

Figure 2–20 shows four PCIe ×8 links in eight transceiver block devices.

Figure 2–20. Four PCIe ×8 Links in Eight Transceiver Block Devices (1)



Note to Figure 2–20:

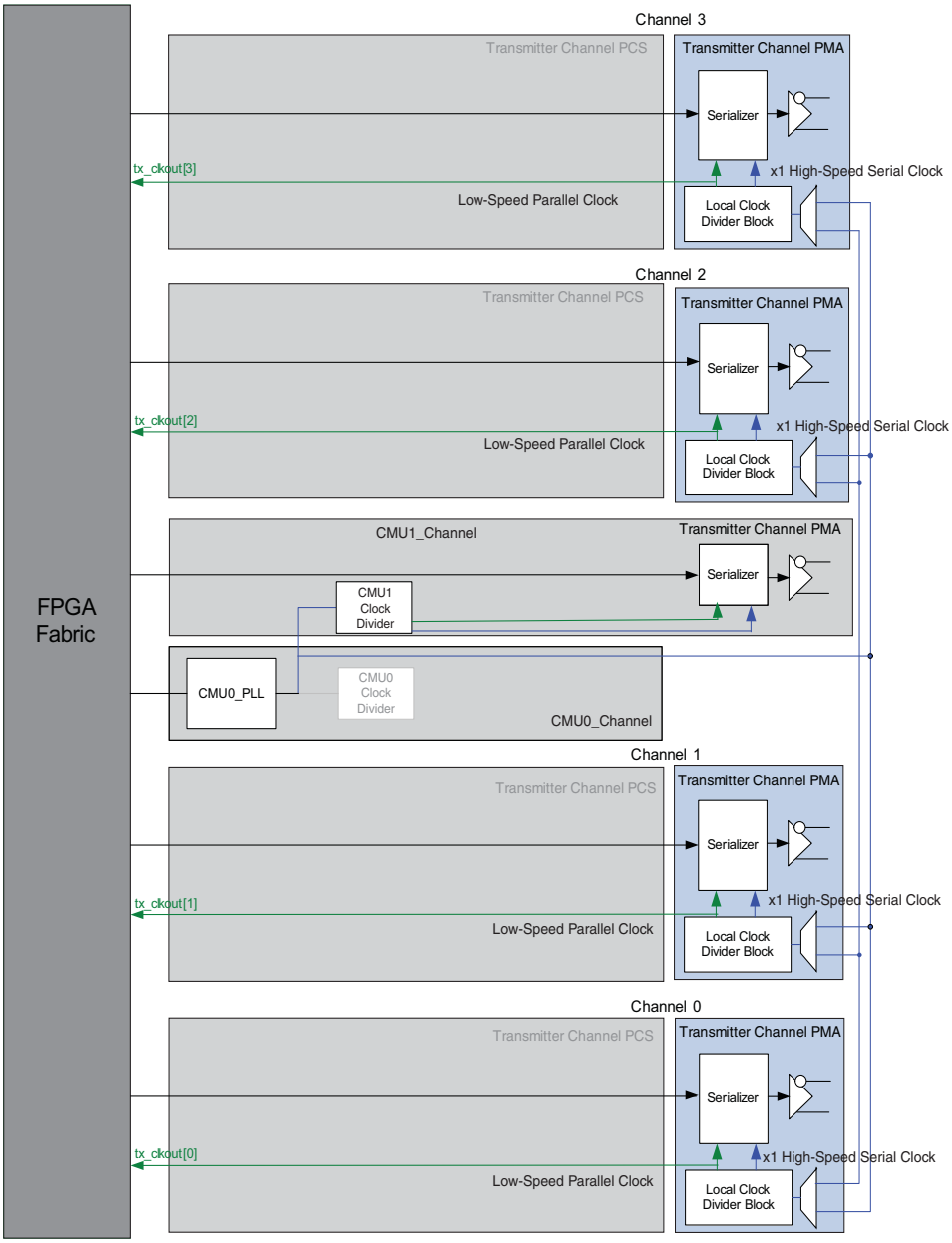
(1) You can use a ×4 PCIe configuration in either a master or slave block.

Non-Bonded Basic (PMA Direct) Mode Channel Configurations

Figure 2–21 shows four regular channels and the CMU1 channel in a transceiver block configured in non-bonded Basic (PMA Direct) mode. Each channel derives its clock independently from either the CMU0 PLL or CMU1 PLL within the same transceiver block if the CMU channel is configured as a CMU PLL.

For more information about Basic (PMA Direct) mode, refer to the *Transceiver Architecture in Stratix IV Devices* chapter.

Figure 2–21. Transmitter Channel PMA Directly Interfacing to the User Logic in the FPGA Fabric (1)



Note to Figure 2–21:

(1) The green lines represent the low-speed parallel clock and the blue lines represent the high-speed serial clock.

Stratix IV devices do not allow the 6G ATX PLL to generate transceiver clocks in non-bonded Basic (PMA Direct) mode. The transmitter clock for channels configured in non-bonded Basic (PMA Direct) mode must be generated by one of the CMU PLLs in the transceiver block containing the channels.

The CMU0 PLL synthesizes the input reference clock to generate a clock that is distributed to the local clock divider block in each of the four regular channels using the ×1 high-speed serial clock line. It is also forwarded to the CMU1 clock divider in the CMU1 channel configured as a non-bonded Basic (PMA-Direct) channel. The local clock divider block in each regular channel and the CMU1 clock divider in the CMU1 channel generate the low-speed parallel clock and high-speed serial clock. The serializer in the transmitter channel PMA of each channel uses both the low-speed parallel clock and high-speed serial clock for its parallel-in-serial-out operation.

The low-speed parallel clock is also driven directly on the tx_clkout port as the FPGA fabric-Transceiver interface clock. You can use the tx_clkout port to clock transmitter data and control logic in the FPGA fabric.

Bonded Basic (PMA Direct) ×N Mode Channel Configurations

Bonded Basic (PMA Direct) ×N mode offers low transmitter channel-to-channel skew in addition to the flexibility of implementing custom PCS logic in the FPGA fabric. Stratix IV devices allow bonding all regular channels and CMU channels on one side of the device in Basic (PMA Direct) ×N mode. For example, devices such as EP4SGX530NF45 or EP4S100G5F45 allow bonding of up to 24 channels placed in four transceiver blocks on each side of the device.

The coreclkout port is not available in Basic (PMA Direct) ×N mode.

In bonded channel configurations, the CMU0 clock divider of all the transceiver blocks is used, as shown in Figure 2–17. Unlike bonded channel configurations, in Basic (PMA Direct) ×N configuration:

- If you use the ATX PLL to generate the transceiver datapath interface clocks, only the clock divider of the ATX PLL is used.
- If you use the CMU PLL to generate the transceiver datapath interface clocks, only the CMU0 clock divider block of the transceiver block containing the CMU PLL is used.

Figure 2–22 shows transmitter channel clocking for 17 channels configured in Basic (PMA Direct) ×N mode.

Figure 2–22. Transmitter Channel Clocking for 17 Channels Configured in Basic (PMA Direct) ×N Mode

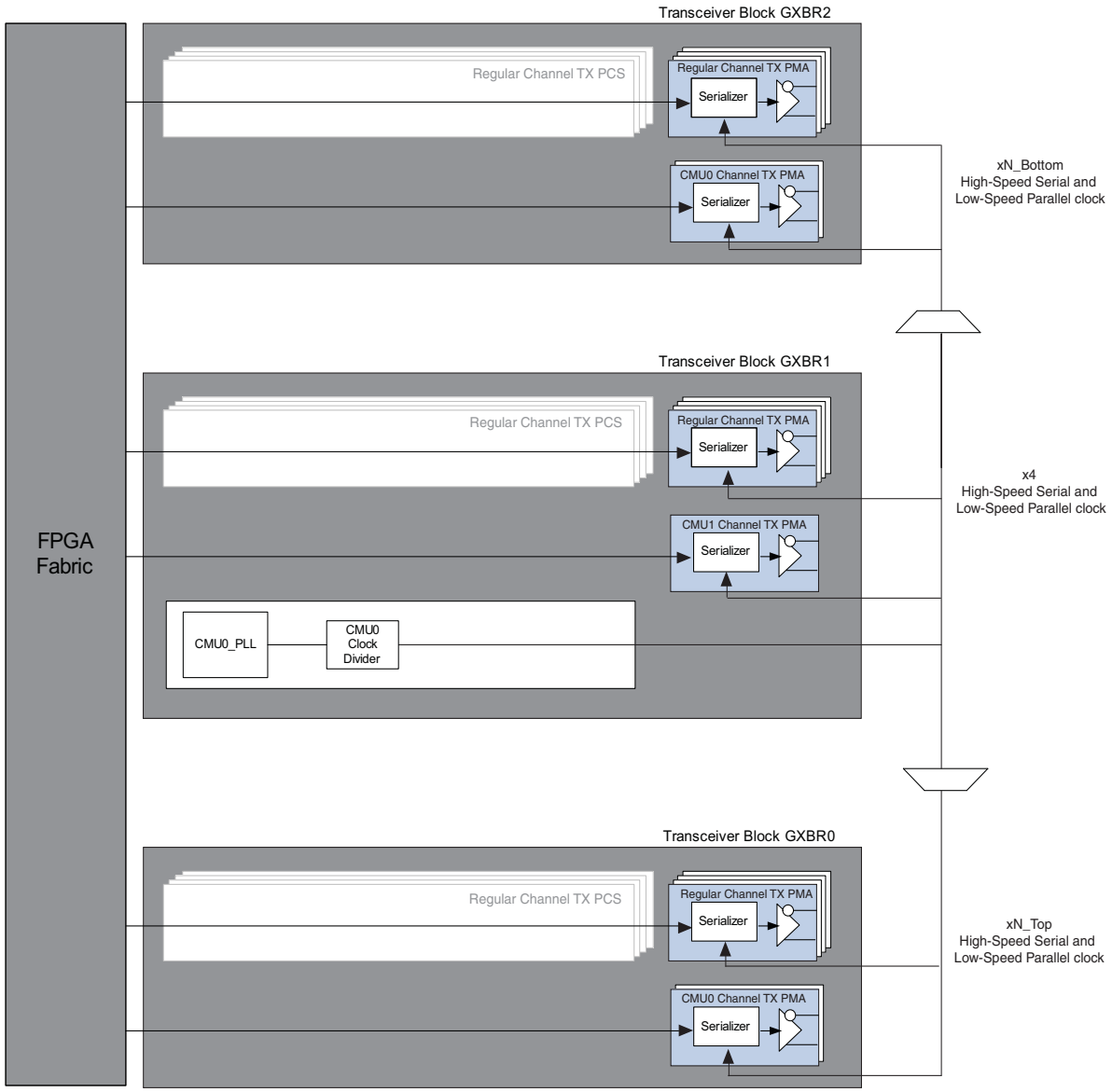


Figure 2–22 shows 17 channels configured in Basic (PMA Direct) ×N mode and located across three transceiver blocks on the right side of the Stratix IV device. Each of the two transceiver blocks, `GXBR0` and `GXBR2`, contain six of the 17 ×N bonded channels located in four regular channels and two CMU channels. The remaining five of the 17 ×N bonded channels are located in four regular channels and the `CMU1` channel of the transceiver block `GXBR1`.

Stratix IV devices allow both CMU channels and 6G ATX PLL blocks to generate the high-speed serial and low-speed parallel transceiver clocks when configured in Basic (PMA Direct) ×N mode.

For more examples regarding this clocking scheme, refer to:

- "Example 1: Channel Configuration with a 4 Gbps Data Rate" on page 2–9
- *AN 571: Implementing the SERDES Framer Interface Level 5 (SFI-5.1) Protocol in Stratix IV Devices*
- *AN 572: Implementing the Scalable SERDES Framer Interface (SFI-S) Protocol in Stratix IV GT Devices*

Transmitter Channel-to-Channel Skew Optimization in Basic (PMA Direct) ×N Mode

In Basic (PMA-Direct) ×N mode, the CMU0 channel distributes the transceiver clocks to the channels placed in the same transceiver block using the ×4 clock lines. The ×4 clock lines drive the ×N_Top and ×N_Bottom clock lines to distribute the transceiver clocks to the transmitter channels located in transceiver blocks on the bottom and top.

The difference in clock routing delays between the ×4 clock lines and the ×N clock lines can result in higher transmitter channel-to-channel skew. To compensate for this difference in clock routing delays between the ×4 and the ×N clock lines, the Stratix IV transceivers introduce a fixed amount of delay in the ×4 clock lines of the transceiver block whose CMU0 channel generates the transceiver clocks.

The delay compensation mechanism engaged in Basic (PMA Direct) mode only compensates for the clock routing delays between the transceiver block whose CMU0 channel generates the transceiver clocks and its adjacent transceiver block located above and below.

To minimize transmitter channel-to-channel skew in ×N bonded channels, use the recommended placement shown in Table 2–8.

Table 2–8. Recommended Placement of Channels and CMU in Bonded Modes

Channel Placement	CMU Placement
2 adjacent transceiver blocks	In either of the two transceiver blocks.
3 adjacent transceiver blocks	In the middle transceiver block.
4 adjacent transceiver blocks	In either of the middle transceiver blocks.

If you use the ATX PLL to generate the transceiver clocks, Altera recommends placing the channels in the transceiver blocks adjacent to the ATX PLL on both sides of the ATX PLL.

For manual placement of the CMU and ATX PLLs, if the Quartus II software does not automatically pick the most optimal location for skew, refer to *AN 578: Manual Placement of CMU PLLs and ATX PLLs in Stratix IV GX and GT Devices*.

Meeting Timing in Basic (PMA Direct) Mode

Timing may not be met for higher data rates when transceiver channels are configured in Basic (PMA Direct) functional mode. To meet FPGA fabric-Transmitter PMA interface timing above certain data rates, you may need to phase shift the interface clock `tx_clkout` used to clock the transmitter user logic. To meet FPGA fabric-Receiver hold time violations, you may have to modify the way data is captured in the FPGA fabric.

For more information, refer to *AN 580: Achieving Timing Closure in Basic (PMA Direct) Functional Mode*.

Receiver Channel Datapath Clocking

This section describes the receiver PMA and PCS datapath clocking in supported configurations. Receiver datapath clocking varies between non-bonded and bonded channel configurations. It also varies with the use of PCS blocks, such as deskew FIFO and rate matcher. This section describes the following:

- "Non-Bonded Channel Configurations"
- "Bonded Channel Configurations" on page 2–43
- "Basic (PMA Direct) Mode Channel Configurations" on page 2–49

Non-Bonded Channel Configurations

In non-bonded channel configurations, receiver PCS blocks of each channel are clocked independently. Each non-bonded channel also has separate `rx_analogreset` and `rx_digitalreset` signals that allow independent reset of the receiver PCS logic in each channel.

For more information about transceiver reset and power down signals, refer to the *Reset Control and Power Down in Stratix IV Devices* chapter.

In non-bonded channel configurations, receiver channel datapath clocking has two scenarios:

- "Non-Bonded Receiver Clocking Without Rate Matcher"
- "Non-Bonded Receiver Clocking with Rate Matcher" on page 2–41

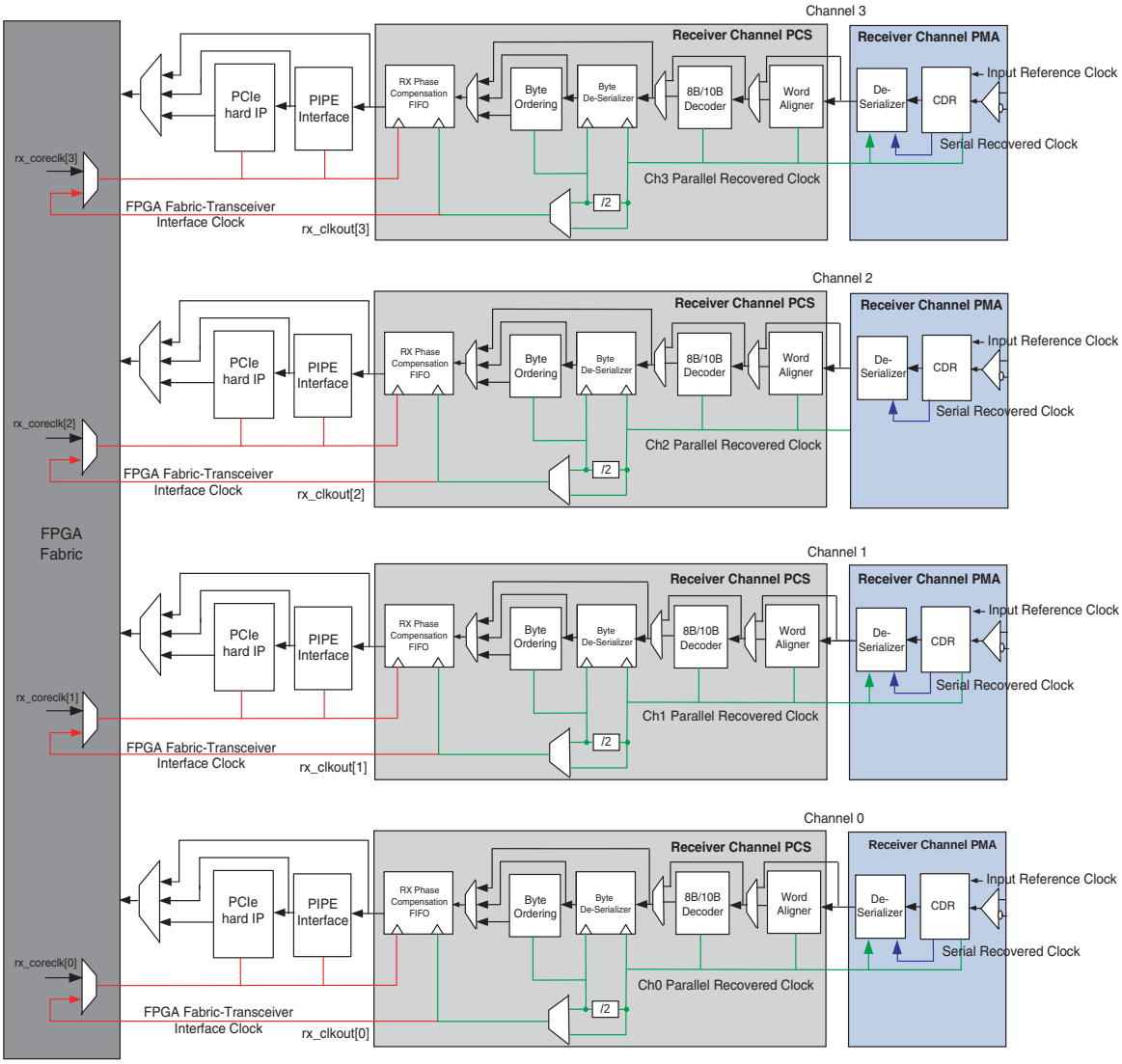
Non-Bonded Receiver Clocking Without Rate Matcher

The following functional modes have non-bonded receiver channel configuration without rate matcher:

- SONET/SDH
- SDI
- (OIF) CEI PHY Interface
- Basic without rate matcher

Figure 2–23 shows receiver datapath clocking in non-bonded channel configurations without rate matcher.

Figure 2–23. Receiver Datapath Clocking in Non-Bonded Configurations Without Rate Matcher (1)



Note to Figure 2–23:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the parallel recovered clock, and the blue lines represent the serial recovered clock.

In non-bonded configurations without rate matcher, the CDR in each receiver channel recovers the serial clock from the received data. The serial recovered clock is divided within the receiver PMA to generate the parallel recovered clock. The deserializer uses the serial recovered clock in the receiver PMA. The parallel recovered clock and deserialized data is forwarded to the receiver PCS. The parallel recovered clock in each channel clocks the word aligner and 8B/10B decoder (if enabled).

Depending on whether you use the byte deserializer or not, the parallel recovered clock (when you do not use the byte deserializer) or a divide-by-two version of the parallel recovered clock (when you use the byte deserializer) clocks the write port of the receiver phase compensation FIFO. This clock is driven directly on the `rx_clkout` port as the FPGA fabric-Transceiver interface clock. You can use the `rx_clkout` signal to capture the receiver data and status signals in the FPGA fabric.

Table 2–9 lists the receiver datapath clock frequencies in non-bonded functional modes without rate matcher.

Table 2–9. Receiver Datapath Clock Frequencies in Non-Bonded Functional Modes Without Rate Matcher

Functional Mode	Data Rate	Serial Recovered Clock Frequency	Parallel Recovered Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Deserializer (MHz)	With Byte Deserializer (MHz)
SONET/SDH OC12	622 Mbps	311 MHz	77.75	77.75	N/A
SONET/SDH OC48	2.488 Gbps	1.244 GHz	311	N/A	155.5
HD-SDI	1.485 Gbps	742.5 MHz	148.5	148.5	74.25
	1.4835 Gbps	741.75 MHz	148.35	148.35	74.175
3G-SDI	2.97 Gbps	1.485 GHz	297	N/A	148.5
	2.967 Gbps	1.4835 GHz	296.7	N/A	148.35

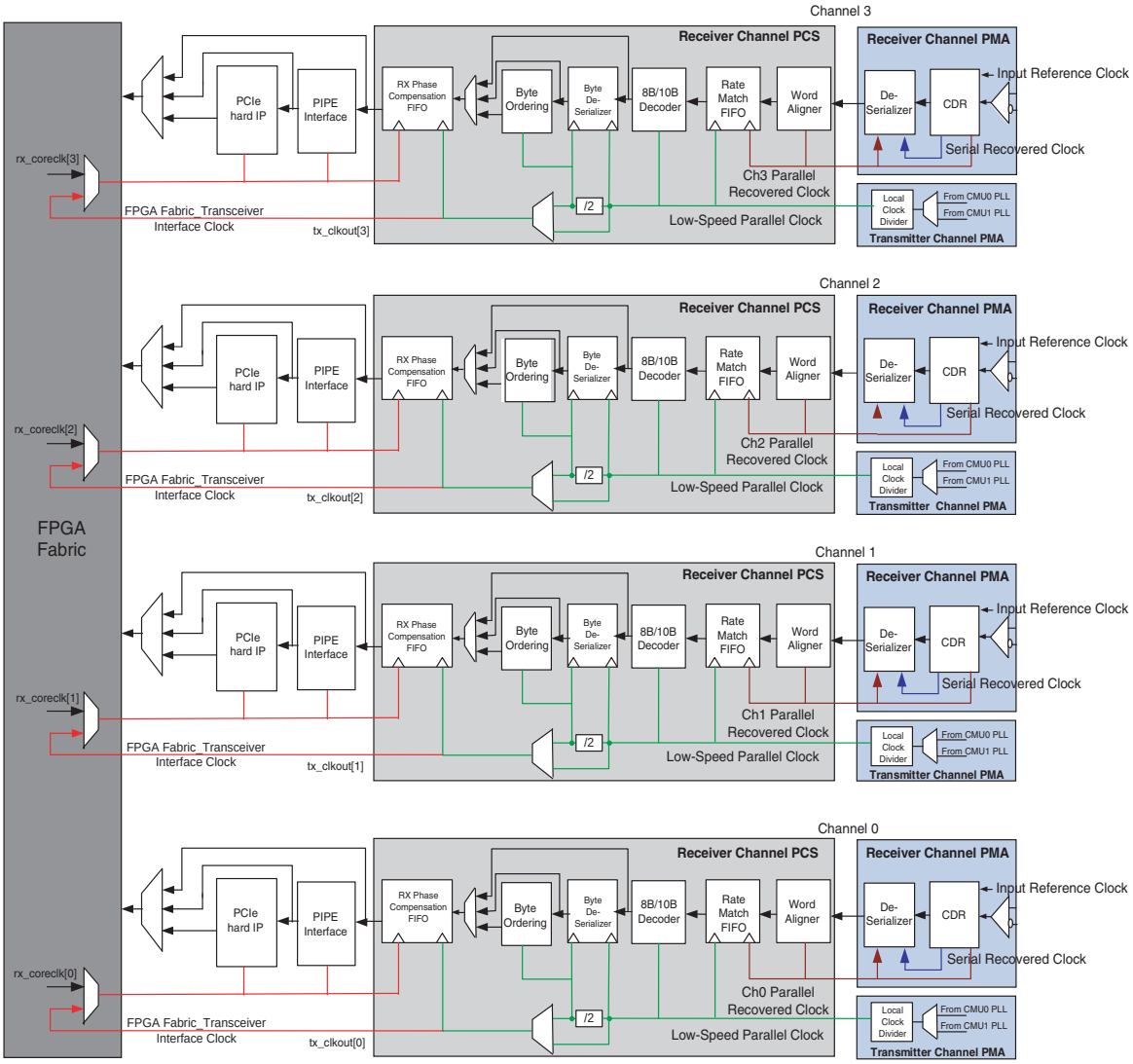
Non-Bonded Receiver Clocking with Rate Matcher

The following functional modes have non-bonded receiver channel configuration with rate-matcher:

- PCIe ×1
- GIGE
- Serial RapidIO
- Basic with rate matcher

Figure 2–24 shows the receiver datapath clocking in non-bonded channel configurations with rate matcher.

Figure 2–24. Receiver Datapath Clocking in Non-Bonded Configurations with Rate Matcher (1)



Note to Figure 2–24:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, the dark red lines represent the parallel recovered clock, and the blue lines represent the serial recovered clock.

In non-bonded configurations with rate matcher, the CDR in each receiver channel recovers the serial clock from the received data. The serial recovered clock is divided within the receiver PMA to generate the parallel recovered clock. The deserializer uses the serial recovered clock in the receiver PMA. The parallel recovered clock and deserialized data are forwarded to the receiver PCS.

The parallel recovered clock from the receiver PMA in each channel clocks the word aligner and the write port of the rate match FIFO. The low-speed parallel clock from the transmitter local clock divider block in each channel clocks the read port of the rate match FIFO, the 8B/10B decoder, and the write port of the byte deserializer (if enabled). The parallel transmitter PCS clock or its divide-by-two version (if byte deserializer is enabled) clocks the write port of the receiver phase compensation FIFO. It is also driven on the `tx_clkout` port as the FPGA fabric-Transceiver interface clock. You can use the `tx_clkout` signal to latch the receiver data and status signals in the FPGA fabric.

Table 2–10 lists the receiver datapath clock frequencies in non-bonded functional modes with rate matcher.

Table 2–10. Receiver Datapath Clock Frequencies in Non-Bonded Functional Modes with Rate Matcher

Functional Mode	Data Rate (Gbps)	Serial Recovered Clock Frequency	Parallel Recovered Clock and Parallel Transmitter PCS Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Deserializer (MHz)	With Byte Deserializer (MHz)
PCIe ×1 (Gen 1)	2.5	1.25 GHz	250	250	125
PCIe ×1 (Gen 2)	5	2.5 GHz	500	N/A	250
GIGE	1.25	625 MHz	125	125	N/A
Serial RapidIO	1.25	625 MHz	125	N/A	62.5
	2.5	1.25 GHz	250	N/A	125
	3.125	1.5625 GHz	312.5	N/A	156.25

Bonded Channel Configurations

The Stratix IV device supports ×4 channel bonding that allows bonding of four channels within the same transceiver block. It also supports ×8 channel bonding that allows bonding of eight channels across two transceiver blocks on the same side of the device.

In bonded channel configurations, the low-speed parallel clock for all bonded channels are generated by the same `CMU0` clock divider or the ATX clock divider block, resulting in lower channel-to-channel clock skew. The receiver phase compensation FIFO in all bonded channels (except in Basic [PMA Direct] ×N mode) share common pointers and control logic generated in the CCU, resulting in equal latency in the receiver phase compensation FIFO of all bonded channels.

Bonding is not supported on the receive side for Basic ×4 and Basic ×8 functional modes. If you use rate matcher, the clocking scheme for Basic ×4 and Basic ×8 functional modes, the clocking is similar to PCIe ×4 mode, as shown in Figure 2–26 on page 2–46 and PCIe ×8 mode, as shown in Figure 2–27 on page 2–48.

×4 Bonded Channel Configuration

The following functional modes support ×4 receiver channel bonded configuration:

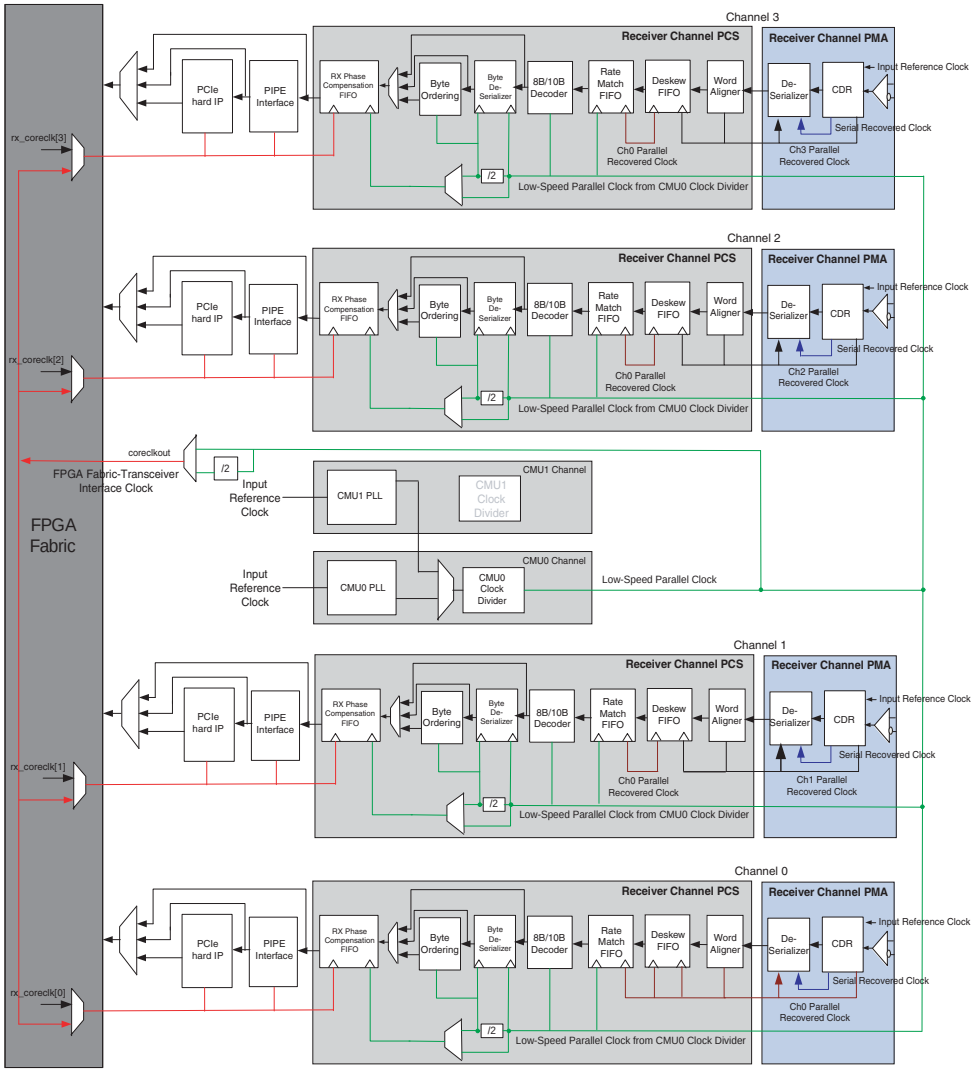
- XAUI ("x4 Bonded Channel Configuration with Deskew FIFO" on page 2–44)
- PCIe ("x4 Bonded Channel Configuration Without Deskew FIFO" on page 2–46)

x4 Bonded Channel Configuration with Deskew FIFO

XAUI functional mode has ×4 bonded channel configuration with deskew FIFO.

Figure 2–25 shows the receiver datapath clocking in ×4 channel bonding configurations with deskew FIFO.

Figure 2–25. Receiver Datapath Clocking in x4 Bonded Channel Configuration with Deskew FIFO (1)



Note to Figure 2–25:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, the dark red lines represent the Ch0 parallel recovered clock, and the blue lines represent the serial recovered clock.

In ×4 bonded channel configurations with deskew FIFO, the CDR in each receiver channel recovers the serial clock from the received data. The serial recovered clock is divided within each channel's receiver PMA to generate the parallel recovered clock. The deserializer uses the serial recovered clock in the receiver PMA. The parallel recovered clock and deserialized data is forwarded to the receiver PCS in each channel.

The parallel recovered clock from the receiver PMA in each channel clocks the word aligner in that channel. The parallel recovered clock from `Channel 0` clocks the deskew FIFO and the write port of the rate match FIFO in all four bonded channels. The low-speed parallel clock from the `CMU0` clock divider block in `CMU0_Channel` clocks the read port of the rate match FIFO, the 8B/10B decoder, and the write port of the byte deserializer (if enabled) in all four bonded channels. The low-speed parallel clock or its divide-by-two version (if byte deserializer is enabled) clocks the write port of the receiver phase compensation FIFO. It is also driven on the `coreclkout` port as the FPGA fabric-Transceiver interface clock. You can use the `coreclkout` signal to latch the receiver data and status signals in the FPGA fabric for all four bonded channels.

Table 2–11 lists the receiver datapath clock frequencies in ×4 bonded functional modes with deskew FIFO.

Table 2–11. Receiver Datapath Clock Frequencies in x4 Bonded Functional Modes with Deskew FIFO

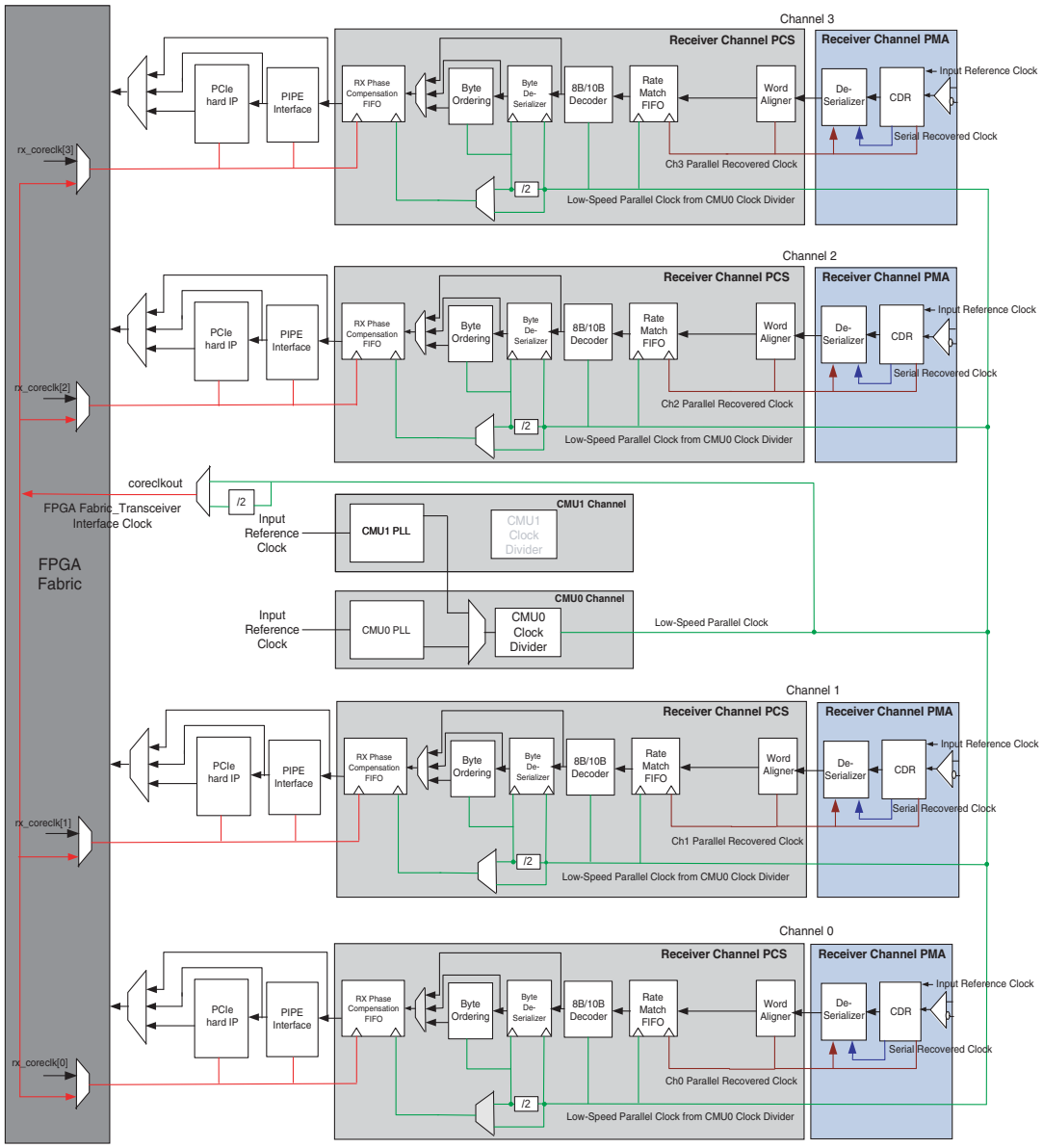
Functional Mode	Data Rate (Gbps)	Serial Recovered Clock Frequency	Parallel Recovered Clock and Parallel Transmitter PCS Clock Frequency (MHz)	FPGA-Fabric Transceiver Interface Clock Frequency	
				Without Byte Deserializer (MHz)	With Byte Deserializer (MHz)
PCIe ×4 (Gen 1)	2.5	1.25 GHz	250	250	125
PCIe ×4 (Gen 2)	5	2.5 GHz	500	N/A	250
XAUI	3.125	1.5625 MHz	312.5	N/A	156.25

x4 Bonded Channel Configuration Without Deskew FIFO

PCIe ×4 functional modes supports the ×4 bonded channel configuration without deskew FIFO.

Figure 2–26 shows the receiver datapath clocking in ×4 channel bonding configurations without deskew FIFO.

Figure 2–26. Receiver Datapath Clocking in x4 Bonded Channel Configuration Without Deskew FIFO (1)



Note to Figure 2–26:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, the dark red lines represent the parallel recovered clock, and the blue lines represent the serial recovered clock.

In ×4 bonded channel configurations without deskew FIFO, the CDR in each receiver channel recovers the serial clock from the received data. The serial recovered clock is divided within each channel's receiver PMA to generate the parallel recovered clock. The deserializer uses the serial recovered clock in the receiver PMA. The parallel recovered clock and deserialized data is forwarded to the receiver PCS in each channel.

The parallel recovered clock from the receiver PMA in each channel clocks the word aligner and the write side of the rate matcher FIFO in that channel. The low-speed parallel clock from the `CMU0` clock divider block in `CMU0_Channel` clocks the read port of the rate match FIFO, the 8B/10B decoder, and the write port of the byte deserializer (if enabled). The low-speed parallel clock or its divide-by-two version (if byte deserializer is enabled) clocks the receiver phase compensation FIFO. It is also driven on the `coreclkout` port as the FPGA fabric-Transceiver interface clock. You can use the `coreclkout` signal to latch the receiver data and status signals in the FPGA fabric for all four bonded channels.

Table 2–12 lists the receiver datapath clock frequencies in ×4 bonded functional modes without deskew FIFO.

Table 2–12. Receiver Datapath Clock Frequencies in x4 Bonded Functional Modes without Deskew FIFO

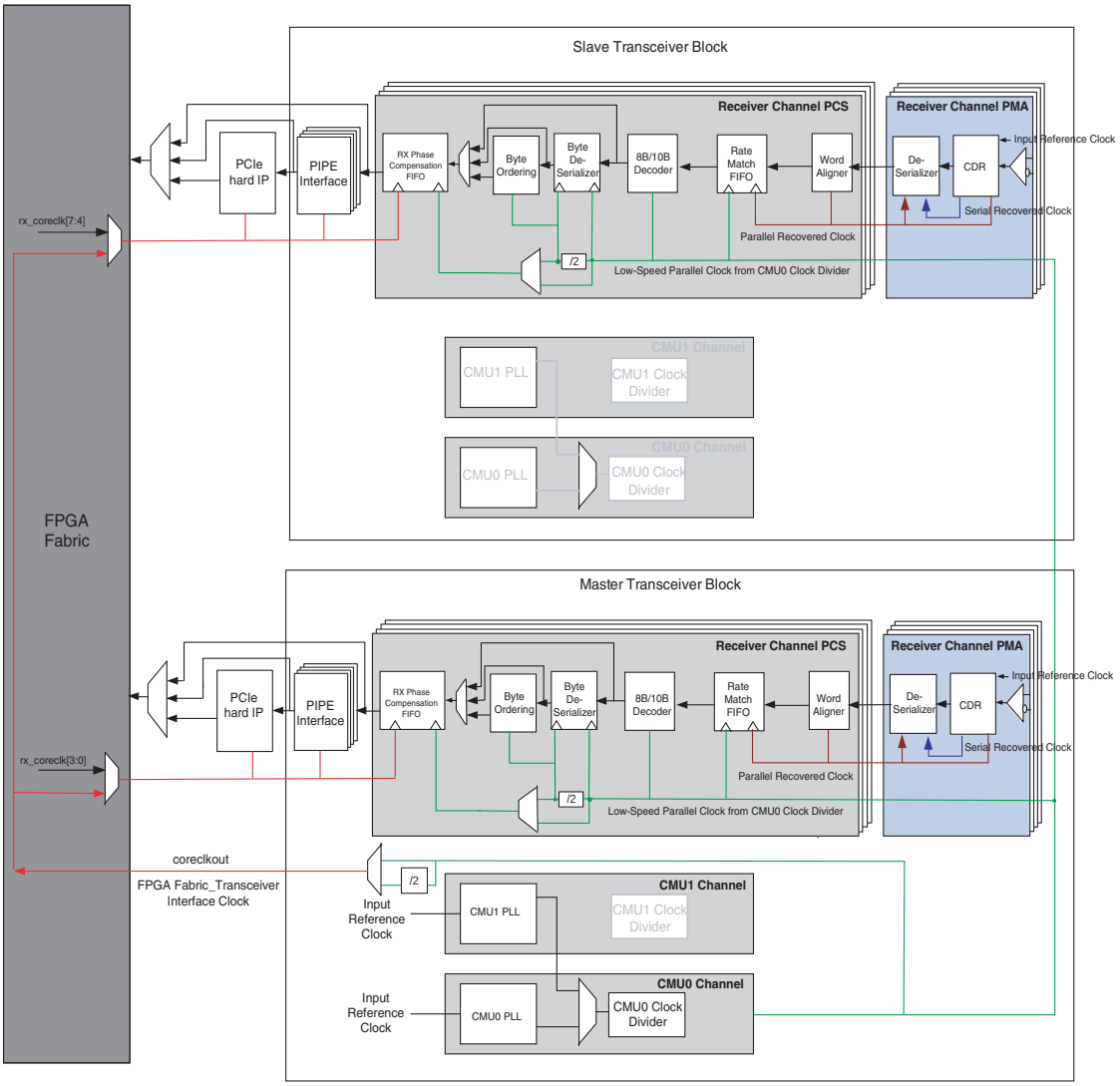
Functional Mode	Data Rate (Gbps)	Serial Recovered Clock Frequency (GHz)	Parallel Recovered Clock and Parallel Transmitter PCS Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Deserializer (MHz)	With Byte Deserializer (MHz)
PCIe ×4 (Gen 1)	2.5	1.25	250	250	125
PCIe ×4 (Gen 2)	5	2.5	500	N/A	250

x8 Bonded Channel Configuration

PCIe ×8 functional mode supports the ×8 receiver channel bonding configuration. The eight bonded channels are located in two transceiver blocks, referred to as the master transceiver block and slave transceiver block, with four channels each.

Figure 2–27 shows the receiver datapath clocking in PCIe ×8 bonded channel configuration.

Figure 2–27. Receiver Datapath Clocking in x8 Bonded Channel Configuration (1)



Note to Figure 2–27:

(1) The red lines represent the FPGA fabric-Transceiver interface clock, the green lines represent the low-speed parallel clock, the dark red lines represent the parallel recovered clock, and the blue lines represent the serial recovered clock.

The CDR in each of the eight receiver channels recovers the serial clock from the received data on that channel. The serial recovered clock is divided within each channel's receiver PMA to generate the parallel recovered clock. The deserializer uses the serial recovered clock in the receiver PMA. The parallel recovered clock and deserialized data from the receiver PMA in each channel is forwarded to the receiver PCS in that channel.

The parallel recovered clock from the receiver PMA in each channel clocks the word aligner and the write side of the rate matcher FIFO in that channel. The low-speed parallel clock from the `CMU0` clock divider of the master transceiver block clocks the read port of the rate match FIFO, the 8B/10B decoder, and the write port of the byte deserializer (if enabled) in all eight channels. The low-speed parallel clock or its divide-by-two version (if byte deserializer is enabled) clocks the write port of the receiver phase compensation FIFO in all eight channels. It is also driven on the `coreclkout` port as the FPGA fabric-Transceiver interface clock. You can use the `coreclkout` signal to latch the receiver data and status signals in the FPGA fabric for all eight bonded channels.

Table 2–13 lists the receiver datapath clock frequencies in PCIe ×8 functional mode.

Table 2–13. Receiver Datapath Clock Frequencies PCIe x8 Functional Mode

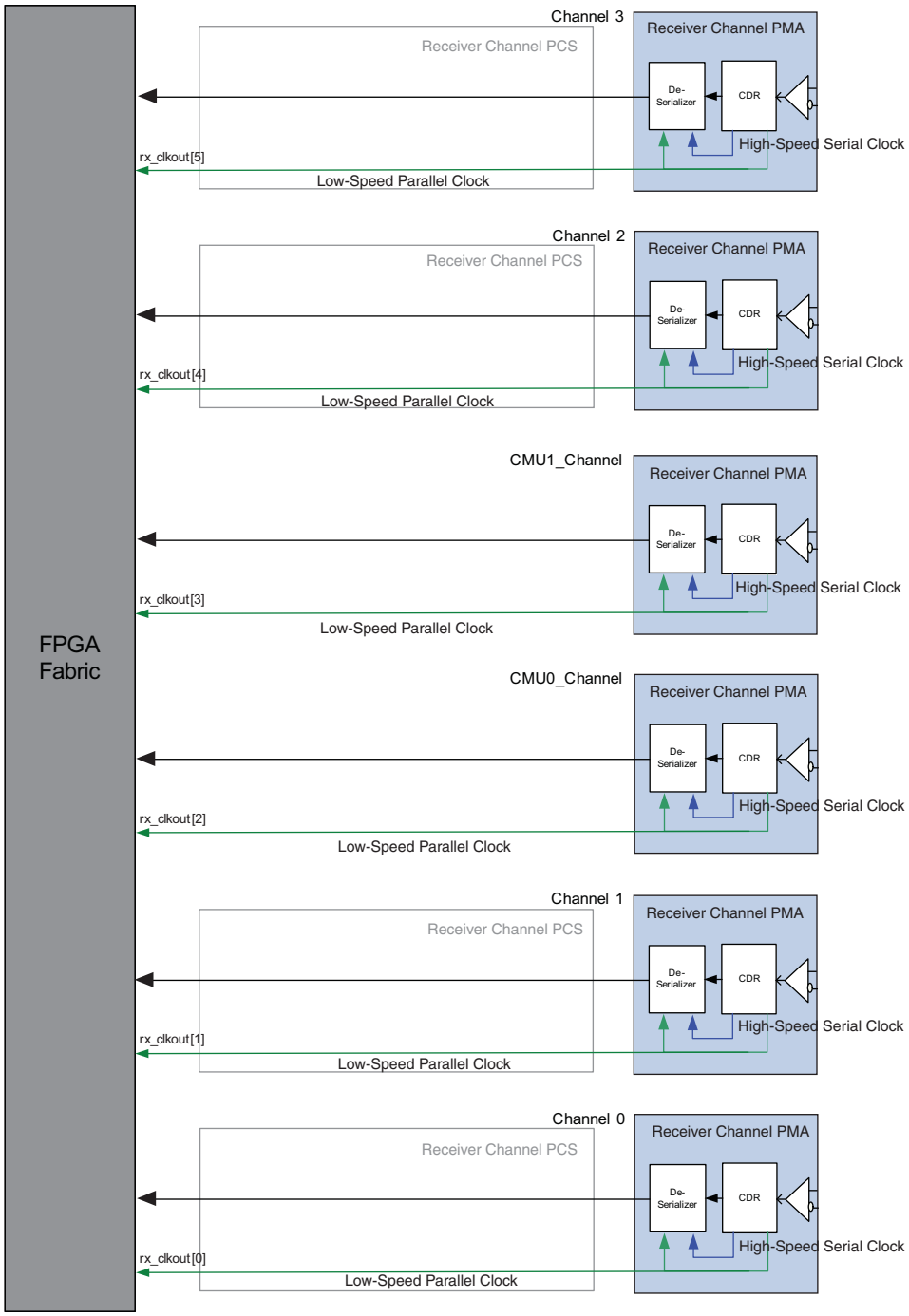
Functional Mode	Data Rate (Gbps)	Serial Recovered Clock Frequency (GHz)	Parallel Recovered Clock and Parallel Transmitter PCS Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Clock Frequency	
				Without Byte Deserializer (MHz)	With Byte Deserializer (MHz)
PCIe ×8 (Gen 1)	2.5	1.25	250	250	125
PCIe ×8 (Gen 2)	5	2.5	500	N/A	250

Basic (PMA Direct) Mode Channel Configurations

Figure 2–28 shows six channels in a transceiver block configured in Basic (PMA Direct) functional mode with two of the channels being CMU channels. The receiver channel PMA directly interfaces to the user logic in the FPGA fabric. The CDR recovers the high-speed serial clock and low-speed parallel clock for the deserializer. The low-speed parallel clock is forwarded to the FPGA fabric as `rx_clkout`.

Bonded mode is not available for receivers configured in Basic (PMA Direct) functional mode. Data registers to capture the receiver data in the FPGA fabric for each channel must be clocked by `rx_clkout` forwarded by that channel's CDR.

Figure 2–28. Receiver Channel PMA Directly Interfacing to the User Logic in the FPGA Fabric (1)



Note to Figure 2–28:

(1) The green lines represent the low-speed parallel clock and the blue lines represent the serial recovered clock.

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 that include the global, regional, and periphery clock networks.

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

- **Input Reference Clocks**—Refer to "Input Reference Clock Source" on page 2–3.
- **Transceiver Datapath Interface Clocks**—are used to transfer data, control, and status signals between the FPGA fabric and the transceiver channels. The transceiver channel forwards the `tx_clkout` signal (in non-bonded modes) or the `coreclkout` signal (in bonded channel modes) 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 rate matcher) or `tx_clkout/coreclkout` (in configurations with rate matcher) to the FPGA fabric to clock the data and status signals from the receiver into the FPGA fabric.
- **Other Transceiver Clocks**—The following transceiver clocks form a part of the FPGA fabric-Transceiver interface clocks:
  - `cal_blk_clk`—calibration block clock
  - `fixed_clk`—125 MHz fixed-rate clock used in the PCIe receiver detect circuitry and for the adaptive equalization (AEQ) block
  - `reconfig_clk`—clock used for transceiver dynamic reconfiguration (for more information, refer to Table 2–5 on page 2–9)

In Basic (PMA Direct) functional mode, only `tx_clkout` and `rx_clkout` are available to clock the logic in the core. In bonded mode, you may use `tx_clkout` of one of the channels to clock all of the channels. For receivers in bonded mode, you must use separate `rx_clkout` for each channel.

Table 2–14 lists the FPGA fabric-Transceiver interface clocks.

Table 2–14. FPGA Fabric-Transceiver Interface Clocks ***(Note 1)*** **(Part 1 of 2)**

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization *(1)*
`pll_inclk`	CMU PLL input reference clock when driven from an FPGA CLK input pin	FPGA fabric-to-transceiver	Global clock
`rx_cruclk`	Receiver CDR input reference clock when driven from an FPGA CLK input pin	FPGA fabric-to-transceiver	Global clock, Regional clock
`tx_clkout`	Phase compensation FIFO clock	Transceiver-to-FPGA fabric	Global clock, Regional clock, Periphery clock
`coreclkout`	Phase compensation FIFO clock	Transceiver-to-FPGA fabric	Global clock, Regional clock, Periphery clock
`rx_clkout`	Phase compensation FIFO clock	Transceiver-to-FPGA fabric	Global clock, Regional clock, Periphery clock
`fixed_clk`	PCIe receiver detect clock	FPGA fabric-to-transceiver	Global clock, Regional clock

Table 2–14. FPGA Fabric-Transceiver Interface Clocks *(Note 1)* **(Part 2 of 2)**

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization *(1)*
reconfig_clk (2)	Transceiver dynamic reconfiguration clock	FPGA fabric-to-transceiver	Global clock
cal_blk_clk	Transceiver calibration block clock	FPGA fabric-to-transceiver	Global clock, Regional clock

Notes to Table 2–11:

(1) For more information about global, regional, and periphery clock resources available in each device, refer to the *Clock Networks and PLLs in Stratix IV Devices* chapter.

(2) Ensure that the `reconfig_clk` is a free-running clock that is not derived from the transceiver blocks.

"FPGA Fabric-Transmitter Interface Clocking" on page 2–52 and "FPGA Fabric-Receiver Interface Clocking" on page 2–61 describe the criteria and methodology to share transmitter and receiver phase compensation FIFO clocks in order to reduce the global, regional, and periphery clock resource usage in your design.

FPGA Fabric-Transmitter Interface Clocking

The transmitter phase compensation FIFO compensates for the phase difference between the FPGA fabric clock (phase compensation FIFO write clock) and the parallel transmitter PCS clock (phase compensation FIFO read clock). The transmitter phase compensation FIFO write clock forms the FPGA fabric-Transmitter interface clock. The phase compensation FIFO write clock and read clocks must have exactly the same frequency (0 parts-per-million [PPM] frequency difference).

Stratix IV transceivers provide the following two options for selecting the transmitter phase compensation FIFO write clock:

- "Quartus II-Selected Transmitter Phase Compensation FIFO Write Clock"
- "User-Selected Transmitter Phase Compensation FIFO Write Clock" on page 2–58

User-selection is provided to share transceiver datapath interface clocks in order to reduce the global, regional, and periphery clock resource usage in your design.

Quartus II-Selected Transmitter Phase Compensation FIFO Write Clock

If you do not select the `tx_coreclk` port in the ALTGX MegaWizard™ Plug-In Manager, the Quartus II software automatically selects the transmitter phase compensation FIFO write clock for each channel in that ALTGX instance. The Quartus II software selects the FIFO write clock depending on the channel configuration.

Non-Bonded Channel Configuration

In a non-bonded channel configuration, the transmitter channels may or may not be identical. Identical transmitter channels are defined as channels that have exactly the same CMU PLL input reference clock source, exactly the same CMU PLL configuration, and exactly the same transmitter PMA and PCS configuration.

Identical transmitter channels may have different transmitter voltage output differential (V_{OD}), transmitter common mode voltage (V_{CM}), or pre-emphasis setting.

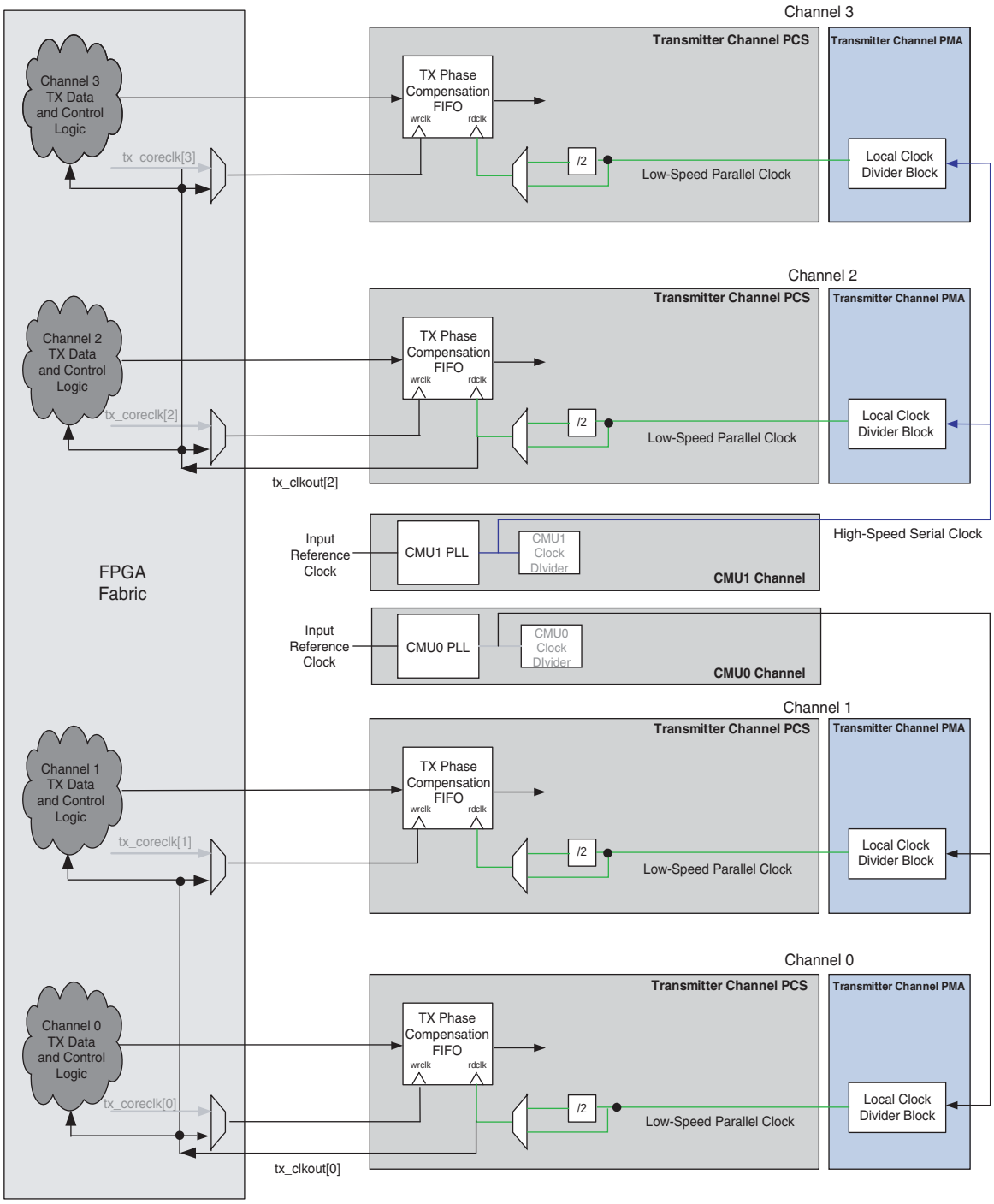
Example 3: Two Groups of Two Identical Channels in a Transceiver Block

Example 3 assumes channels 0 and 1, driven by `CMU0_PLL` in a transceiver block, are identical. Also, channels 2 and 3, driven by `CMU1_PLL` in the same transceiver block, are identical. In this case, the Quartus II software automatically drives the write port of the transmitter phase compensation FIFO in channels 0 and 1 with the `tx_clkout[0]` signal. It also drives the write port of the transmitter phase compensation FIFO in channels 2 and 3 with the `tx_clkout[2]` signal. Use the `tx_clkout[0]` signal to clock the transmitter data and control logic for channels 0 and 1 in the FPGA fabric. Use the `tx_clkout[2]` signal to clock the transmitter data and control logic for channels 2 and 3 in the FPGA fabric.

This configuration uses two FPGA global and/or regional clock resources, one for the `tx_clkout[0]` signal and the other for the `tx_clkout[2]` signal.

Figure 2–29 shows the FPGA fabric-Transmitter interface clocking for Example 3.

Figure 2–29. FPGA Fabric-Transmitter Interface Clocking for Example 3 (1)



Note to Figure 2–29:

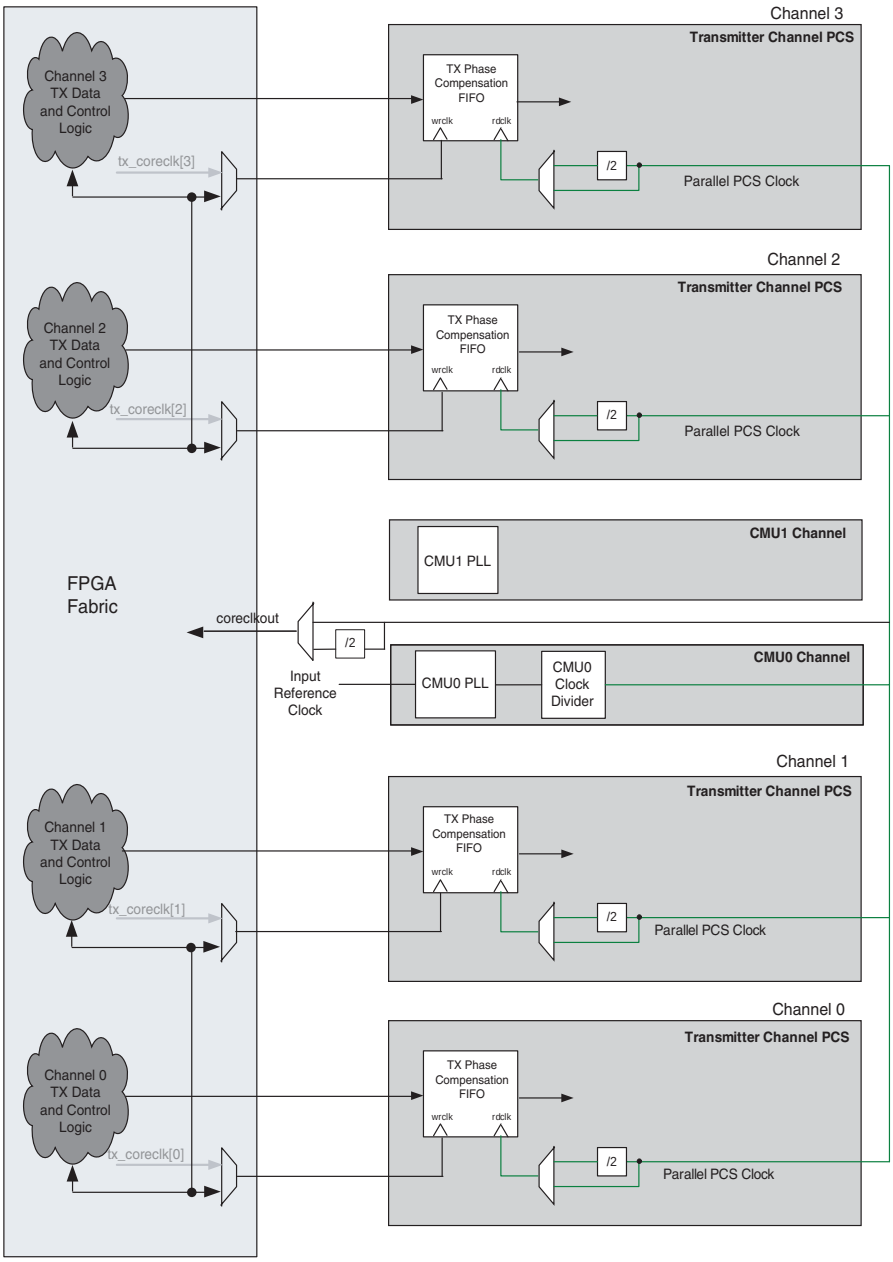
(1) The green lines represent the low-speed parallel clock and the blue lines represent the high-speed serial clock.

Bonded Channel Configuration

In ×4 and ×8 bonded channel configurations, all channels within the transceiver block are identical. The Quartus II software automatically drives the write port of the transmitter phase compensation FIFO in all channels with the `coreclkout` signal. Use the `coreclkout` signal to clock the transmitter data and control logic for all four channels in the FPGA fabric.

Figure 2–30 shows the FPGA fabric-Transmitter interface clocking in a ×4 bonded channel configuration.

Figure 2–30. FPGA Fabric-Transmitter Interface Clocking in a x4 Bonded Channel Configuration (1)



Note to Figure 2–30:

(1) The green lines represent the parallel PCS clock.

Limitations of the Quartus II Software-Selected Transmitter Phase Compensation FIFO Write Clock

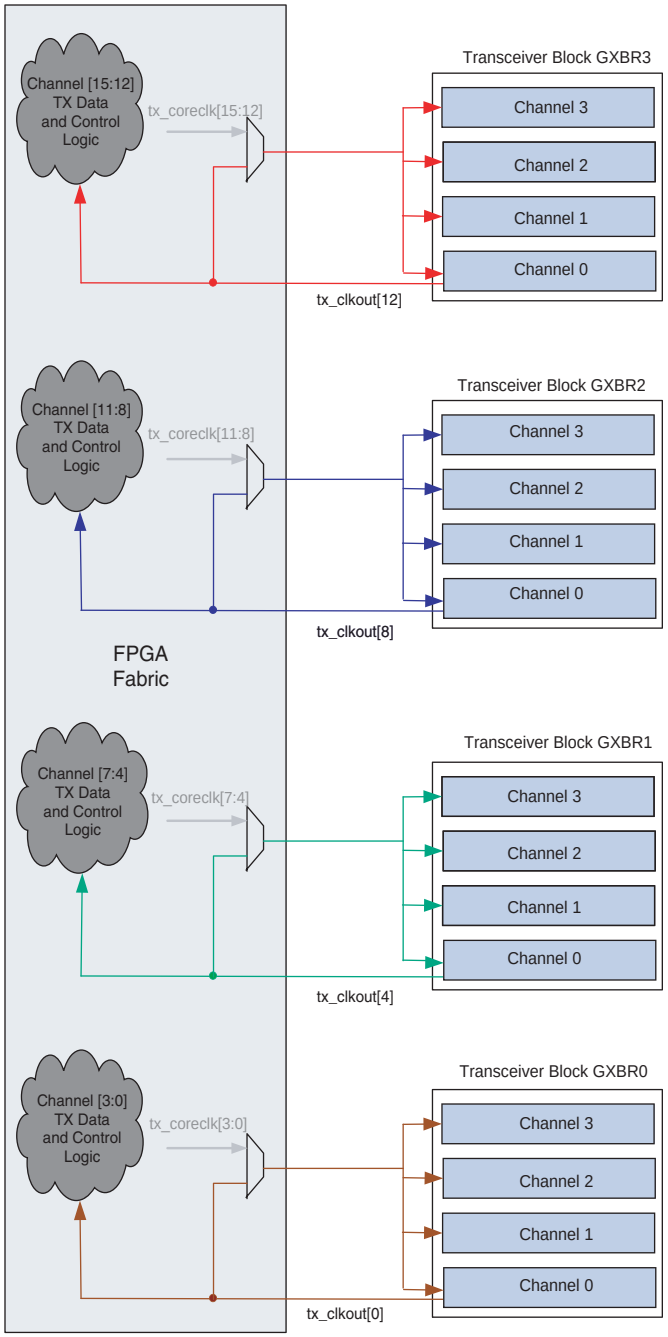
The Quartus II software uses a single tx_clkout signal to clock the transmitter phase compensation FIFO write port of all identical channels within a transceiver block. This results in one global and/or regional clock resource being used for each group of identical channels within a transceiver block.

For identical channels located across the transceiver blocks, the Quartus II software does not use a single tx_clkout signal to clock the write port of the transmitter phase compensation FIFOs for all channels. It uses one tx_clkout signal for each group of identical channels per transceiver block. This results in higher global and regional clock resource usage.

Example 4: Sixteen Identical Channels Across Four Transceiver Blocks

Figure 2–31 shows 16 identical transmitter channels located across four transceiver blocks. The Quartus II software uses tx_clkout from Channel 0 in each transceiver block to clock the write port of the transmitter phase compensation FIFO in all four channels in that transceiver block. This results in four global and/or regional clock resources being used, one for each transceiver block.

Figure 2–31. Sixteen Identical Channels Across Four Transceiver Blocks for Example 4 (1)



Note to Figure 2–31:

(1) The red lines represent `tx_clkout[12]`, the blue lines represent `tx_clkout[8]`, the green lines represent `tx_clkout[4]`, and the brown lines represent `tx_clkout[0]`.

Because all 16 channels are identical, using a single tx_clkout to clock the transmitter phase compensation FIFO in all 16 channels results in only one global or regional clock resource being used instead of four. To achieve this, you must choose the transmitter phase compensation FIFO write clocks instead of the Quartus II software automatic selection, as described in "User-Selected Transmitter Phase Compensation FIFO Write Clock" on page 2–58.

User-Selected Transmitter Phase Compensation FIFO Write Clock

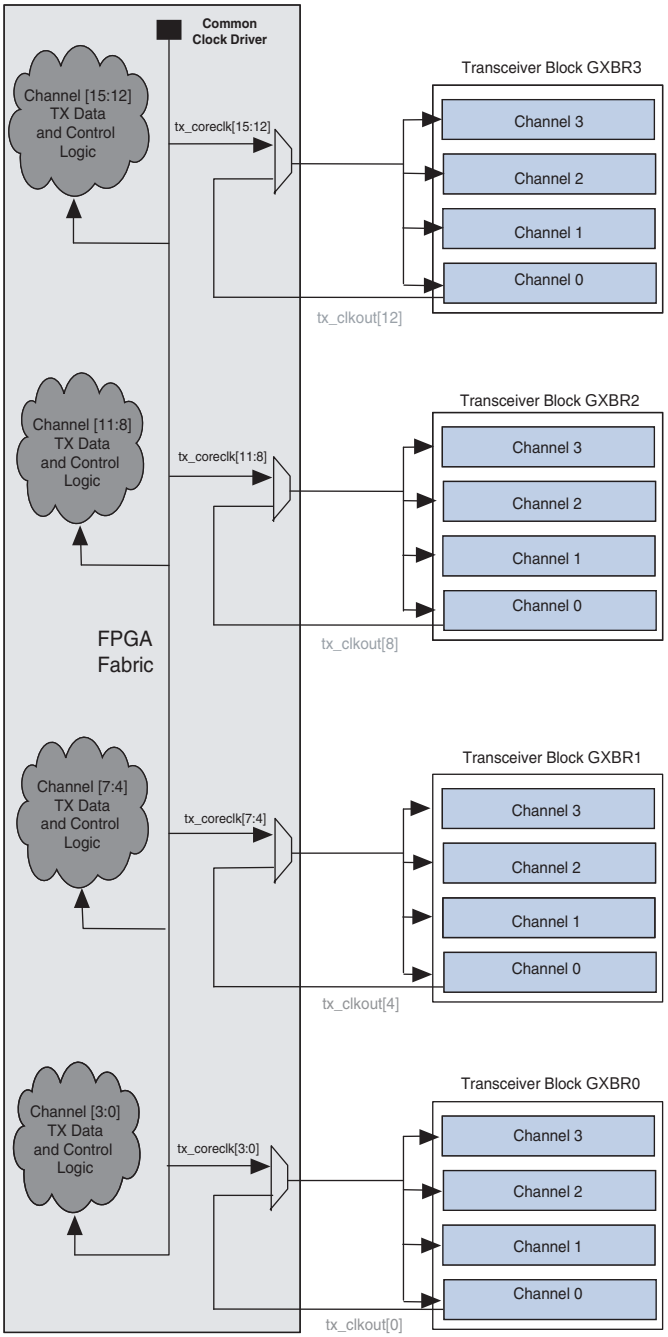
The ALTGX MegaWizard Plug-In Manager provides an optional port named tx_coreclk for each instantiated transmitter channel. If you enable this port, the Quartus II software does not automatically select the transmitter phase compensation FIFO write clock source. Instead, the signal that you drive on the tx_coreclk port of the channel clocks the write side of its transmitter phase compensation FIFO.

Use the flexibility of selecting the transmitter phase compensation FIFO write clock to reduce global and regional clock resource usage. You can connect the tx_coreclk ports of all identical channels in your design and drive them using a common clock driver that has 0 PPM frequency difference with respect to the FIFO read clocks of these channels. Use the common clock driver to clock the transmitter data and control logic in the FPGA fabric for all identical channels. This FPGA fabric-Transceiver interface clocking scheme uses only one global or regional clock resource for all identical channels in your design.

Example 5: Sixteen Identical Channels Across Four Transceiver Blocks

Figure 2–32 shows 16 identical transmitter channels located across four transceiver blocks. The tx_coreclk ports of all 16 transmitter channels are connected together and driven by a common clock driver. This common clock driver also drives the transmitter data and control logic of all 16 transmitter channels in the FPGA fabric. You use only one global or regional clock resource with this clocking scheme, compared to four global and regional clock resources needed without the tx_coreclk ports (the Quartus II software-selected transmitter phase compensation FIFO write clock).

Figure 2–32. Sixteen Identical Channels Across Four Transceiver Blocks for Example 5



Common Clock Driver Selection Rules

The common clock driver driving the `tx_coreclk` ports of all identical channels must have 0 PPM frequency difference with respect to the transmitter phase compensation FIFO read clocks of these channels. If there is any frequency difference between the FIFO write clock (`tx_coreclk`) and the FIFO read clock, the FIFO overflows or under-runs, resulting in corrupted data transfer between the FPGA fabric and the transmitter.

Table 2–15 lists the transmitter phase compensation FIFO read clocks that the Quartus II software selects in various configurations.

Table 2–15. Transmitter Phase Compensation FIFO Read Clocks

Configuration	Transmitter Phase Compensation FIFO Read Clock	
	Without Byte Serializer	With Byte Serializer
Non-Bonded Channel Configuration	Parallel transmitter PCS clock from the local clock divider in the associated channel (tx_clkout)	Divide-by-two version of the parallel transmitter PCS clock from the local clock divider in the associated channel (tx_clkout)
×4 Bonded Channel Configuration	Low-speed parallel clock from the CMU0 clock divider of the associated transceiver block (coreclkout)	Divide-by-two version of the low-speed parallel clock from the CMU0 clock divider of the associated transceiver block (coreclkout)
×8 Bonded Channel Configuration	Low-speed parallel clock from the CMU0 clock divider of the master transceiver block (coreclkout from master transceiver block)	Divide-by-two version of the low-speed parallel clock from the CMU0 clock divider of the master transceiver block (coreclkout from master transceiver block)

To ensure that you understand the 0 PPM clock driver rule, the Quartus II software expects the following set of user assignments whenever you use the tx_coreclk port to drive the transmitter phase compensation FIFO write clock:

- **GXB 0 PPM Core Clock Setting**

Failing to make this assignment correctly when using the tx_coreclk port results in a Quartus II compilation error.

The **GXB 0 PPM core clock** setting allows the following clock drivers to drive the tx_coreclk ports:

- tx_clkout in non-bonded channel configurations
- coreclkout in bonded channel configurations
- FPGA_CLK input pins
- Transceiver refclk pins
- Clock output from left and right and top and bottom PLLs (PLL_L, PLL_R, and PLL_T, PLL_B)

The Quartus II software does not allow gated clocks or clocks generated in FPGA logic to drive the tx_coreclk ports.

Because the **GXB 0 PPM core clock** setting allows the FPGA CLK input pins and transceiver refclk pins as the clock driver, the Quartus II compiler cannot determine if there is a 0 PPM difference between the FIFO write clock and read clock for each channel.

You must ensure that the clock driver for all connected tx_coreclk ports has a 0 PPM difference with respect to the FIFO read clock in those channels.

Table 2–16 lists the Quartus II assignments that you must make in the assignment editor.

Table 2–16. Quartus II Assignments

From	Full design hierarchy name of one of the following clock drivers that you choose to drive the `tx_coreclk` ports of all identical channels *(1)*: ■ `tx_clkout` ■ `coreclkout` ■ FPGA CLK input pins ■ Transceiver `refclk` pins ■ Clock output from the left and right or top and bottom PLLs ■ `tx_dataout` port of one of the identical channels
To	`tx_dataout` pins of all identical channels whose `tx_coreclk` ports are connected together and driven by the 0 PPM clock driver.
Assignment Name	GXB 0 PPM Core Clock Setting
Value	ON

Note to Table 2–13:

(1) You can find the full hierarchy name of the 0 PPM clock driver using the **Node Finder** feature in the Quartus II Assignment Editor.

For more implementation information, refer to "Configuration Example 2: Configuring Sixteen Identical Channels Across Four Transceiver Blocks" on page 2–75.

Basic (PMA Direct) mode

In Basic (PMA Direct) mode, each channel must be clocked by its own `tx_clkout`. As a result, the number of global and/or regional clock resources required is significantly higher. In Basic (PMA Direct) ×N mode, to save on global and/or regional clock resources, you may use `tx_clkout` from centrally located channels to clock all the channels. The `coreclkout` port is not available in Basic (PMA Direct) ×N mode.

FPGA Fabric-Receiver Interface Clocking

The receiver phase compensation FIFO compensates for the phase difference between the parallel receiver PCS clock (FIFO write clock) and the FPGA fabric clock (FIFO read clock). The receiver phase compensation FIFO read clock forms the FPGA fabric-Receiver interface clock. The FIFO write clock and read clock must have exactly the same frequency (0 PPM frequency difference).

Stratix IV transceivers provide the following two options for selecting the receiver phase compensation FIFO read clock:

- "Quartus II Software-Selected Receiver Phase Compensation FIFO Read Clock" on page 2–62
- "User-Selected Receiver Phase Compensation FIFO Read Clock" on page 2–68

User-selection is provided to share transceiver datapath interface clocks in order to reduce the global, regional, and periphery clock resource usage in your design.

Quartus II Software-Selected Receiver Phase Compensation FIFO Read Clock

If you do not select the rx_coreclk port in the ALTGX MegaWizard Plug-In Manager, the Quartus II software automatically selects the receiver phase compensation FIFO read clock for each channel in that ALTGX instance. The Quartus II software selects the FIFO read clock depending on the channel configuration. In non-bonded channel configurations, the FPGA fabric-receiver interface clocking has two scenarios:

- Receivers that do not use a rate matcher block (refer to "Non-Bonded Receiver Clocking Without Rate Matcher" on page 2–39)
- Receivers that use a rate matcher block (refer to "Non-Bonded Receiver Clocking with Rate Matcher" on page 2–41)

Non-Bonded Channel Configuration with Rate Matcher

In non-bonded channel configuration, the transceiver channels may or may not be identical. Identical transceiver channels are defined as channels that have exactly the same CMU PLL and receiver CDR input reference clock sources, exactly the same CMU PLL and receiver CDR configuration, and exactly the same PMA and PCS configuration.

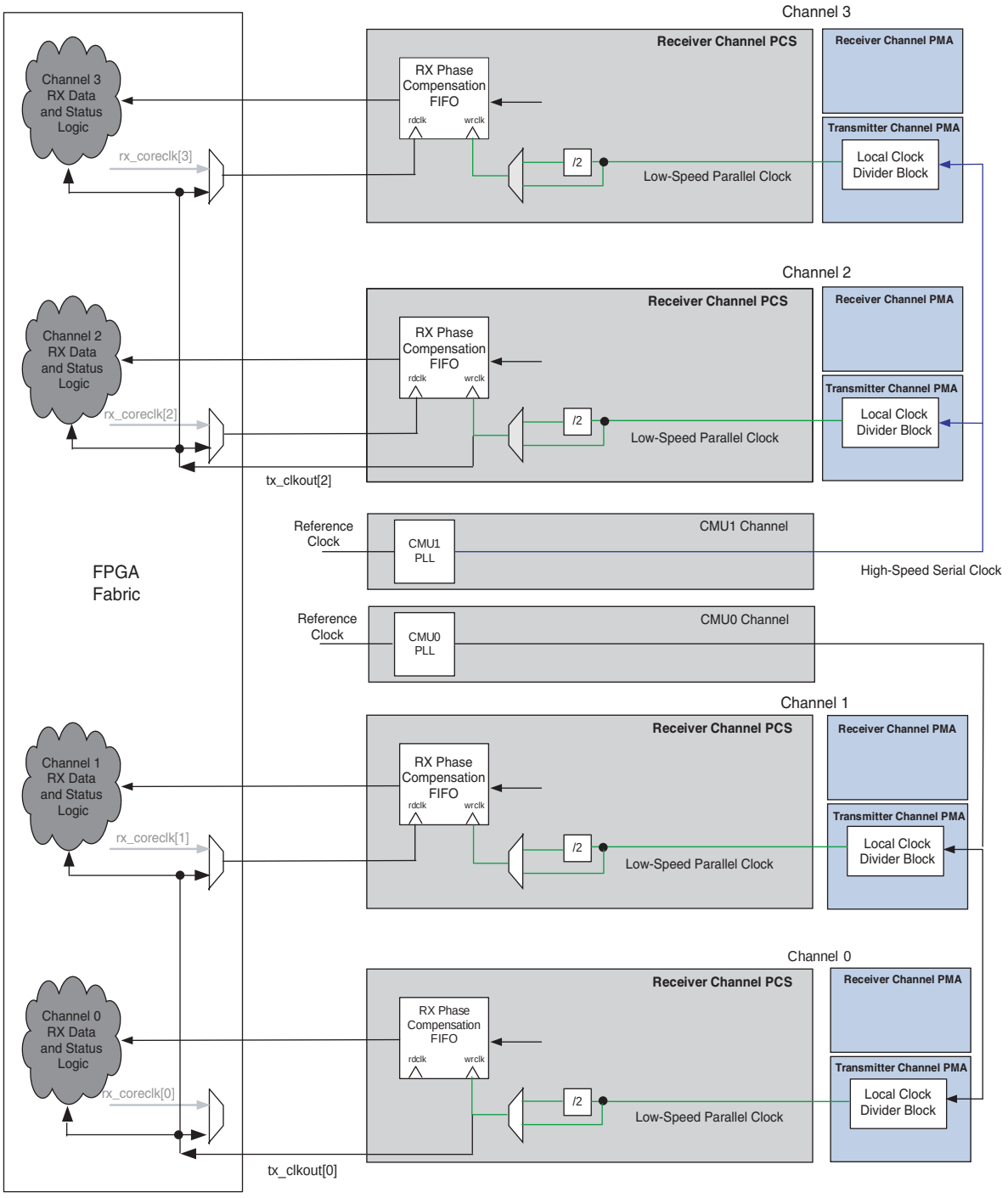
Example 6: Two Groups of Two Identical Channels in a Transceiver Block

Example 6 assumes channels 0 and 1, driven by the CMU0 PLL in a transceiver block, are identical. Also, channels 2 and 3, driven by the CMU1 PLL in the same transceiver block, are identical. In this case, the Quartus II software automatically drives the read port of the receiver phase compensation FIFO in channels 0 and 1 with the tx_clkout[0] signal. It also drives the read port of the receiver phase compensation FIFO in channels 2 and 3 with the tx_clkout[2] signal. Use the tx_clkout[0] signal to latch the receiver data and status signals from channels 0 and 1 in the FPGA fabric. Use the tx_clkout[2] signal to latch the receiver data and status signals from channels 2 and 3 in the FPGA fabric.

This configuration uses two FPGA global and/or regional clock resources, one for the tx_clkout[0] signal and the other for the tx_clkout[2] signal.

Figure 2–33 shows the FPGA fabric-Receiver interface clocking for Example 6.

Figure 2–33. FPGA Fabric-Receiver Interface Clocking for Example 6 (1)



Note to Figure 2–33:

(1) The green lines represent the low-speed parallel clock and the blue lines represent the high-speed serial clock.

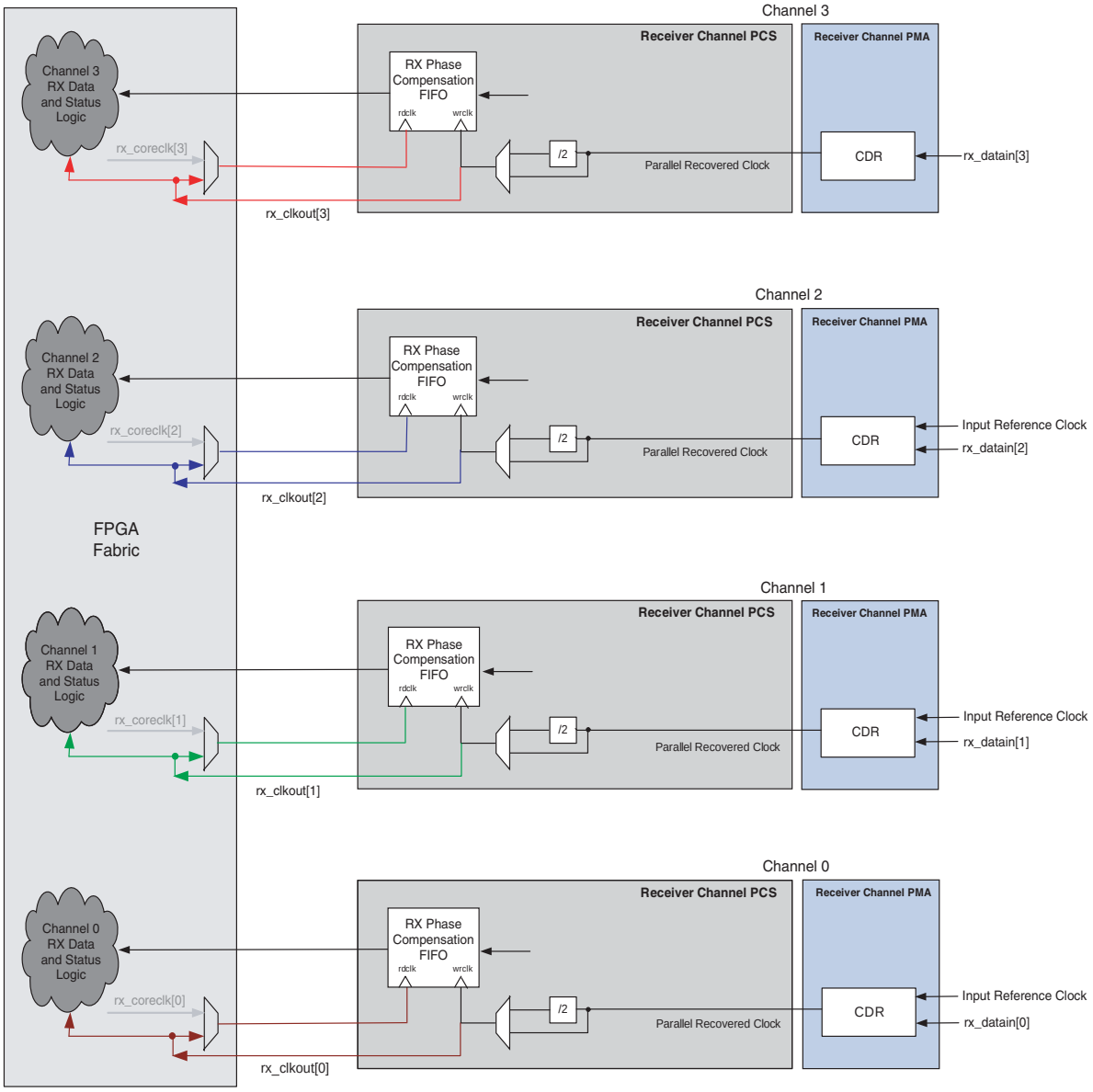
Non-Bonded Channel Configuration Without Rate Matcher

In non-bonded channel configuration without rate matcher, the Quartus II software cannot determine if the incoming serial data in all channels have a 0 PPM frequency difference. The Quartus II software automatically drives the read port of the receiver phase compensation FIFO in each channel with the recovered clock driven on the `rx_clkout` port of that channel. Use the `rx_clkout` signal from each channel to latch its receiver data and status signals in the FPGA fabric.

This configuration uses one FPGA global, regional clock, or both, resource per channel for the `rx_clkout` signal.

Figure 2–34 shows the FPGA fabric-Receiver interface clocking for non-bonded channel configurations without rate matcher.

Figure 2–34. FPGA Fabric-Receiver Interface Clocking for Non-Bonded Channel Configurations Without Rate Matcher (1)



Note to Figure 2–34:

(1) The red lines represent `rx_clkout[3]`, the blue lines represent `rx_clkout[2]`, the green lines represent `rx_clkout[1]`, and the brown lines represent `rx_clkout[0]`.

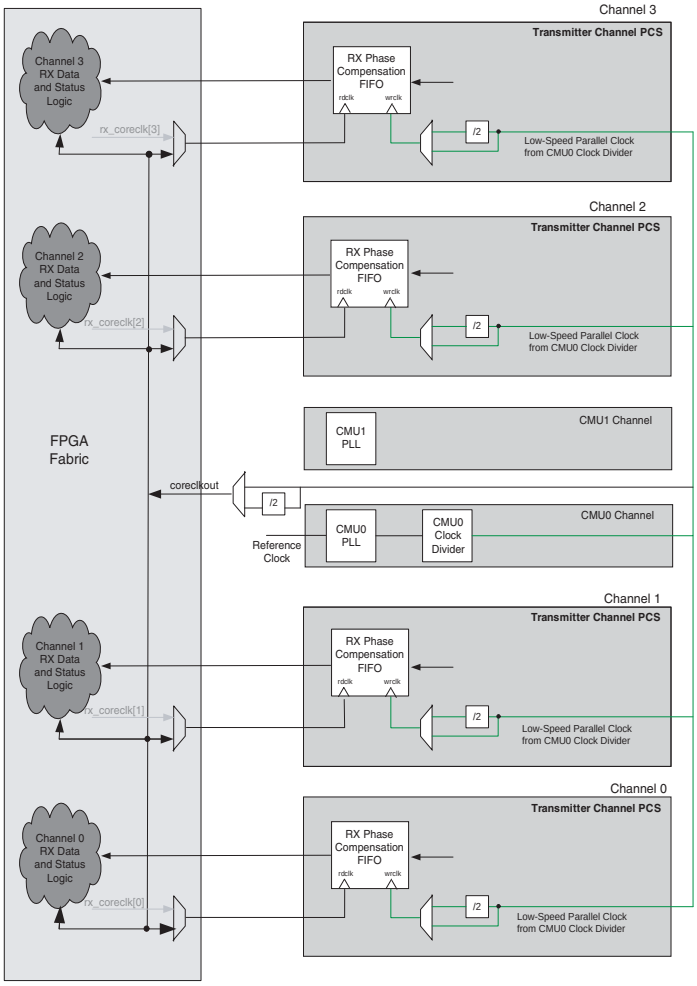
Bonded Channel Configuration

All bonded transceiver channel configurations have rate matcher in the receiver data path. In ×4 and ×8 bonded channel configurations, the Quartus II software automatically drives the read port of the receiver phase compensation FIFO in all channels with the `coreclkout` signal (from the master transceiver block in the case of ×8 bonded mode). Use the `coreclkout` signal to latch the receiver data and status signals from all channels in the FPGA fabric.

This configuration uses one FPGA global and/or regional clock resource per bonded link for the coreclkout signal.

Figure 2–35 shows the FPGA fabric-Receiver interface clocking in ×4 bonded channel configuration.

Figure 2–35. FPGA Fabric-Receiver Interface Clocking in a x4 Bonded Channel Configuration (1)



Note to Figure 2–35:

(1) The green lines represent low-speed parallel clock from the CMU0 clock divider.

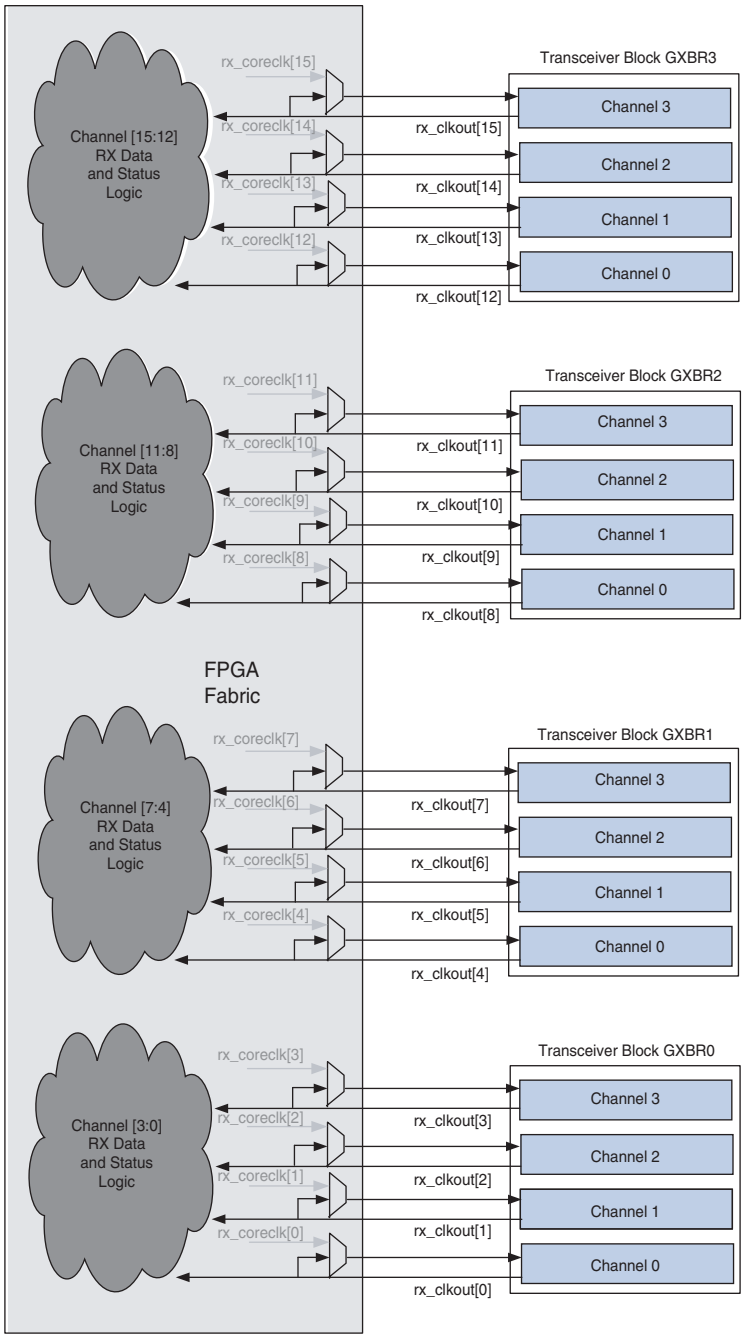
Limitations of the Quartus II Software-Selected Receiver Phase Compensation FIFO Read Clock

In non-bonded channel configurations without rate matcher, the Quartus II software cannot determine if the incoming serial data in all channels has a 0 PPM frequency difference. The Quartus II software uses the recovered clock rx_clkout signal from each channel to clock the read port of its receiver phase compensation FIFO. This results in one global, regional, or global and regional clock resource being used per channel for the rx_clkout signal.

Example 7: Sixteen Channels Across Four Transceiver Blocks

Figure 2–36 shows 16 non-bonded receiver channels without rate matcher, located across four transceiver blocks. The incoming serial data to all 16 channels have a 0 PPM frequency difference with respect to each other. The Quartus II software uses `rx_clkout` from each channel to clock the read port of its receiver phase compensation FIFO. This results in 16 global, regional, or global and regional clock resources being used, one for each channel.

Figure 2–36. Sixteen Non-Bonded Receiver Channels without Rate Match for Example 7



Because the recovered clock rx_clkout signals from all 16 channels have a 0 PPM frequency difference, you can use a single rx_clkout to clock the receiver phase compensation FIFO in all 16 channels. This results in only one global, regional, or global and regional clock resource being used instead of 16. To achieve this, you must select the receiver phase compensation FIFO read clocks instead of the Quartus II software default selection, as described in "User-Selected Receiver Phase Compensation FIFO Read Clock" on page 2–68.

User-Selected Receiver Phase Compensation FIFO Read Clock

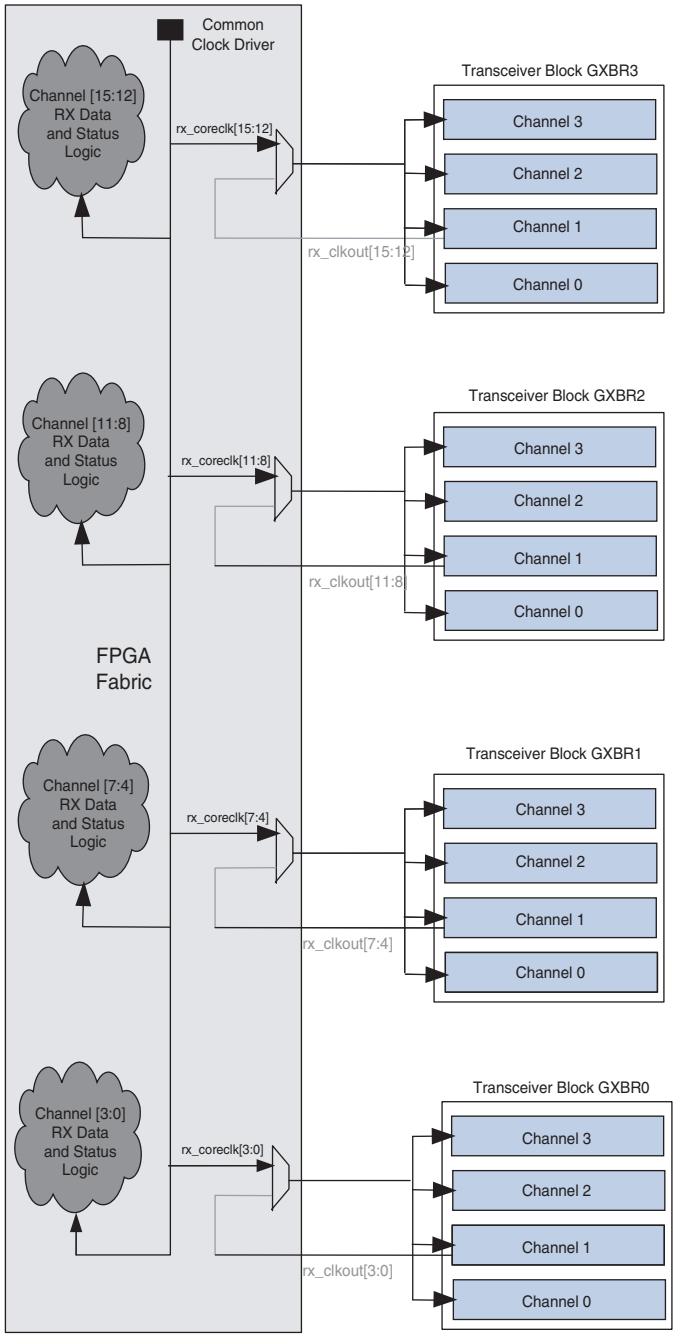
The ALTGX MegaWizard Plug-In Manager provides an optional port named rx_coreclk for each instantiated receiver channel. If you enable this port, the Quartus II software does not automatically select the receiver phase compensation FIFO read clock source. Instead, the signal that you drive on the rx_coreclk port of the channel clocks the read side of its receiver phase compensation FIFO.

You can use the flexibility of selecting the receiver phase compensation FIFO read clock to reduce the global, regional, or global and regional clock resource usage. You can connect the rx_coreclk ports of all the receiver channels in your design and drive them using a common clock driver that has a 0 PPM frequency difference with respect to the FIFO write clocks of these channels. Use this common clock driver to latch the receiver data and status signals in the FPGA fabric for these channels. This FPGA fabric-Transceiver interface clocking scheme uses only one global, regional, or global and regional clock resource for all channels.

Example 8: Sixteen Identical Channels Across Four Transceiver Blocks

Figure 2–37 shows 16 channels located across four transceiver blocks. The incoming serial data to all 16 channels has a 0 PPM frequency difference with respect to each other. The rx_coreclk ports of all 16 channels are connected together and driven by a common clock driver. This common clock driver also latches the receiver data and status logic of all 16 receiver channels in the FPGA fabric. Only one global, regional, or global and regional clock resource is used with this clocking scheme, compared to 16 global, regional, or global and regional clock resources needed without the rx_coreclk ports (the Quartus II software-selected receiver phase compensation FIFO read clock).

Figure 2–37. Sixteen Identical Channels Across Four Transceiver Blocks for Example 8



Common Clock Driver Selection Rules

The common clock driver driving the `rx_coreclk` ports of all channels must have a 0 PPM frequency difference with respect to the receiver phase compensation FIFO write clocks of these channels. If there is any frequency difference between the FIFO read clock (`rx_coreclk`) and the FIFO write clock, the FIFO overflows or under-runs, resulting in corrupted data transfer between the FPGA fabric and the receiver.

Table 2–17 lists the receiver phase compensation FIFO write clocks that the Quartus II software selects in various configurations.

Table 2–17. Receiver Phase Compensation FIFO Write Clocks

Configuration	Receiver Phase Compensation FIFO Write Clock	
	Without Byte Serializer	With Byte Serializer
Non-Bonded Channel Configuration with rate matcher	Low-speed parallel clock from the local clock divider in the associated channel (tx_clkout)	Divide-by-two version of the low-speed parallel clock from the local clock divider in the associated channel (tx_clkout)
Non-Bonded Channel Configuration without rate matcher	Parallel recovered clock from the receiver PMA in the associated channel (rx_clkout)	Divide-by-two version of the parallel recovered clock from the receiver PMA in the associated channel (rx_clkout)
×4-Bonded Channel Configuration	Low-speed parallel clock from the CMU0 clock divider of the associated transceiver block (coreclkout)	Divide-by-two version of the low-speed parallel clock from the CMU0 clock divider of the associated transceiver block (coreclkout)
×8-Bonded Channel Configuration	Low-speed parallel clock from the CMU0 clock divider of the master transceiver block (coreclkout from the master transceiver block)	Divide-by-two version of the low-speed parallel clock from the CMU0 clock divider of the master transceiver block (coreclkout from the master transceiver block)

To ensure that you understand the 0 PPM clock driver rule, the Quartus II software expects the following user assignment whenever you use the rx_coreclk port to drive the receiver phase compensation FIFO read clock:

- **GXB 0 PPM Core Clock Setting**

Failing to make this assignment correctly when using the rx_coreclk port results in a Quartus II compilation error.

The **GXB 0 PPM core clock setting** user assignment allows the following clock drivers to drive the rx_coreclk ports:

- tx_clkout in non-bonded channel configurations with rate matcher
- tx_clkout and rx_clkout in non-bonded configurations without rate matcher
- coreclkout in bonded channel configurations
- FPGA CLK input pins
- Transceiver refclk pins
- Clock output from left and right and top and bottom PLLs (PLL_L, PLL_R, and PLL_T, PLL_B)

The Quartus II software does not allow gated clocks or clocks generated in FPGA logic to drive the tx_coreclk ports.

Because the 0 PPM clock group assignment allows the FPGA CLK input pins and transceiver refclk pins as the clock driver, the Quartus II compiler cannot determine if there is a 0 PPM difference between the FIFO write clock and read clock for each channel.

☞ You must ensure that the clock driver for all the connected `rx_coreclk` ports has a 0 PPM difference with respect to the FIFO write clock in those channels.

Table 2–18 lists the Quartus II assignments that you must make.

Table 2–18. Quartus II Assignments

From	Full design hierarchy name of one of the following clock drivers that you choose to drive the `rx_coreclk` ports of all identical channels *(1)*: ■ `tx_clkout` ■ `rx_clkout` ■ `coreclkout` ■ FPGA CLK input pins ■ Transceiver `refclk` pins ■ Clock output from the left and right or top and bottom PLLs ■ `tx_dataout` port of one of the identical channels
To	`rx_datain` pins of all channels whose `rx_coreclk` ports are connected together and driven by the 0 PPM clock driver.
Assignment Name	GXB 0 PPM Core Clock Setting
Value	ON

Note to Table 2–18:

(1) You can find the full hierarchy name of the 0 PPM clock driver using the **Node Finder** feature in the Quartus II Assignment Editor.

For more implementation details, refer to "Configuration Example 3: Configuring Sixteen Channels Across Four Transceiver Blocks" on page 2–76.

Basic (PMA Direct) Mode

In Basic (PMA Direct) mode, each channel must be clocked by its own `rx_clkout`. As a result, the number of global and/or regional clock resources required is significantly higher. Bonding is not supported for receivers configured in Basic (PMA Direct) functional mode.

Using the CMU/ATX PLL for Clocking User Logic in the FPGA Fabric

Some designs that use multiple clock domains may run out of PLLs in the FPGA fabric. In such a scenario, if your design has CMU or ATX PLLs that are not being used, it may be possible to use them for clocking user logic in the FPGA fabric. However, the CMU PLLs and ATX PLLs do not have many features that are supported by the PLLs in the FPGA fabric.

The following are the supported features on CMU PLLs and ATX PLLs used as PLLs for clocking user logic in the FPGA fabric:

- Single clock output
- Programmable PLL bandwidth
- PLL PFD power down control
- Lock status signal

To use this feature, you must create an ALTGX instance with a single channel in **Transmitter Only** mode that uses the required CMU PLL or ATX PLL. To create the ALTGX instance, follow these steps:

1. Choose **Basic (PMA Direct) ×N mode** as the protocol.
2. Select **Transmitter Only** operation mode.
3. Select the input clock frequency.
4. Select the appropriate values of data rate and channel width based on the desired output clock frequency. To generate a 250 MHz clock using an input clock frequency of 50 MHz, select a channel width of **10** and a data rate of **2500 Mbps** (Equation 2–1).

Equation 2–1.

$$f_{out} = \frac{\text{data rate}}{\text{channel width}}$$

5. You can select the PLL bandwidth by choosing **Tx PLL bandwidth mode**.
6. You can instantiate the `pll_locked` port to indicate the PLL lock status.
7. You can instantiate `pll_powerdown` or `gxb_powerdown` to enable the PLL PFD power down control.
8. Use `tx_clkout` of the ALTGX instance as the clock source for clocking user logic in the FPGA fabric.

Configuration Examples

This section describes the following examples:

- "Configuration Example 1: Configuring 24 Channels in Basic (PMA Direct) ×N Mode in the EP4S100G5F45 Device" on page 2–73
- "Configuration Example 2: Configuring Sixteen Identical Channels Across Four Transceiver Blocks" on page 2–75
- "Configuration Example 3: Configuring Sixteen Channels Across Four Transceiver Blocks" on page 2–76
- "Configuration Example 4: Configuring Left and Right, Left, or Right PLL in VCO Bypass Mode" on page 2–78

Configuration Example 1: Configuring 24 Channels in Basic (PMA Direct) ×N Mode in the EP4S100G5F45 Device

Each transceiver block has four regular channels and two CMU channels that you can configure in Basic (PMA Direct) ×N mode. The EP4S100G5F45 device has four transceiver blocks located on each side of the device allowing configuration of up to 24 channels in Basic (PMA Direct) ×N mode.

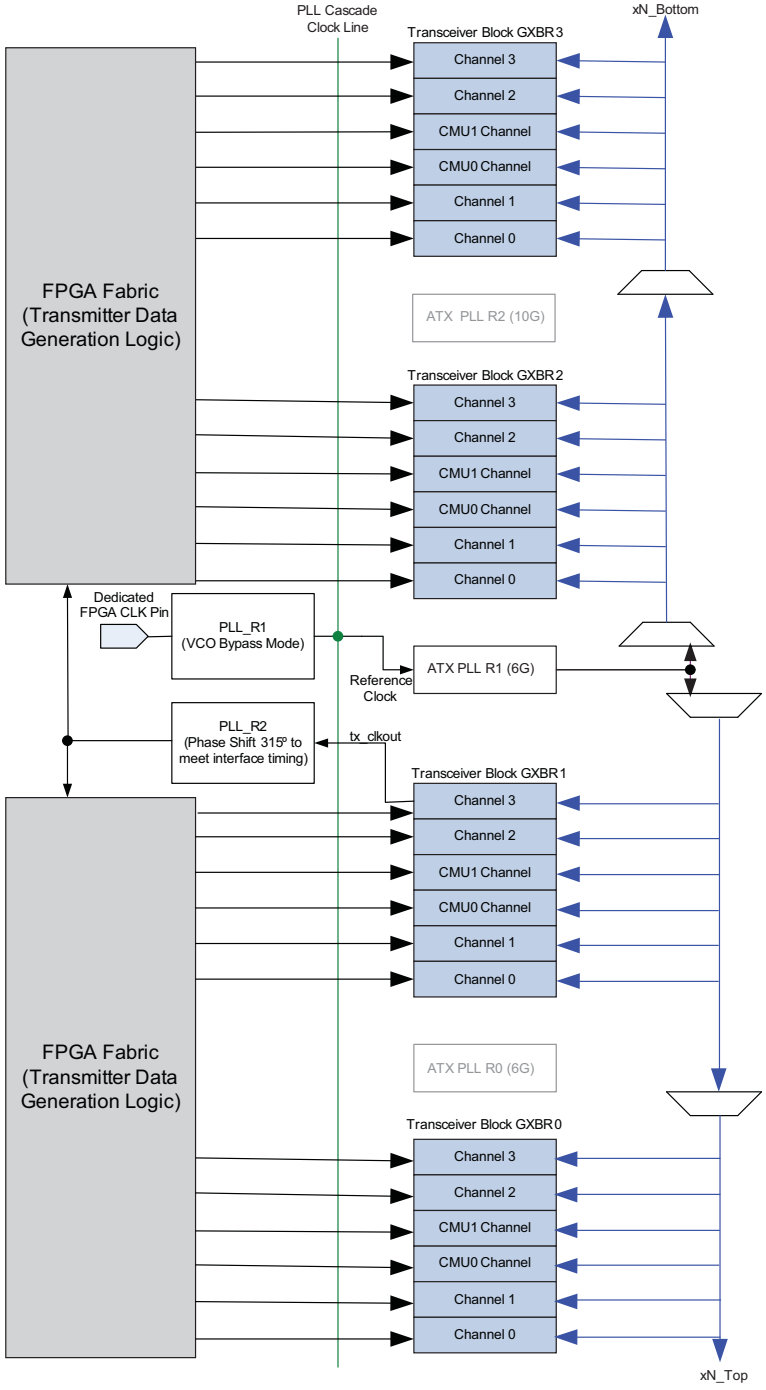
When all 24 channels on one side of the device are configured in Basic (PMA Direct) ×N mode, all eight CMU channels (two in each transceiver block) are configured as PMA-Only channels.

Use the `refclk` pins in each of the four transceiver blocks as receiver serial data input pins and configure the CMU PLLs as receiver CDRs when the CMU channel is configured as a PMA-Only channel. Due to the non-availability of CMU PLLs, you must use the 6G ATX PLL to generate the high-speed serial and low-speed parallel transceiver clocks for all 24 channels. Due to the non-availability of a `refclk` pin, you must use the left and right, or left or right PLL in VCO bypass mode to provide the reference clock through the PLL cascade clock line.

For more information about left and right PLL VCO bypass mode, refer to "Configuration Example 4: Configuring Left and Right, Left, or Right PLL in VCO Bypass Mode" on page 2–78.

Figure 2–38 shows 24 channels on the right side of the EP4S100G5F45 device configured in Basic (PMA Direct) ×N mode running at 6.5 Gbps with a 20-bit FPGA fabric-PMA interface width. Because all 24 channels on the right side of the device are configured in Basic (PMA Direct) ×N mode, the right `PLL_R1` configured in VCO bypass mode is used to provide the input reference clock to the 6G ATX PLL. The 6G ATX PLL generates the high-speed serial and low-speed parallel transceiver clocks that are distributed to the 24 channels though the `×N_Top` and `×N_Bottom` clock network. Because the data rate of 6.5 Gbps requires a left and right, or left or right PLL to meet FPGA fabric-Transmitter PMA interface timing, `tx_clkout` from one of the 24 channels is phase shifted by 315° using `PLL_R2`. The phase shifted output clock from `PLL_R2` is used to clock the FPGA fabric logic that generates the transmitter parallel data and control signals.

Figure 2–38. Twenty-Four Channels on the Right Side of the EP4S100G5F45 Device Configured in Basic (PMA Direct) ×N Mode for Configuration Example 1 (1)



Note to Figure 2–38:

(1) The green line represents the PLL cascade clock line and the blue lines represent the 6G ATX PLL block.

Configuration Example 2: Configuring Sixteen Identical Channels Across Four Transceiver Blocks

This example relates to "User-Selected Receiver Phase Compensation FIFO Read Clock" on page 2–68.

Figure 2–39 shows 16 identical transmitter channels located across four transceiver blocks. The `tx_coreclk` ports of all 16 transmitter channels are connected together and driven by the `tx_clkout[4]` signal from channel 0 in transceiver block `GXBR1`. The `tx_clkout[4]` signal also drives the transmitter data and control logic of all 16 transmitter channels in the FPGA fabric. With this clocking scheme, only one global clock resource is used by the `tx_clkout[4]` signal.

Figure 2–39. Sixteen Identical Channels Across Four Transceiver Blocks for Configuration Example 2

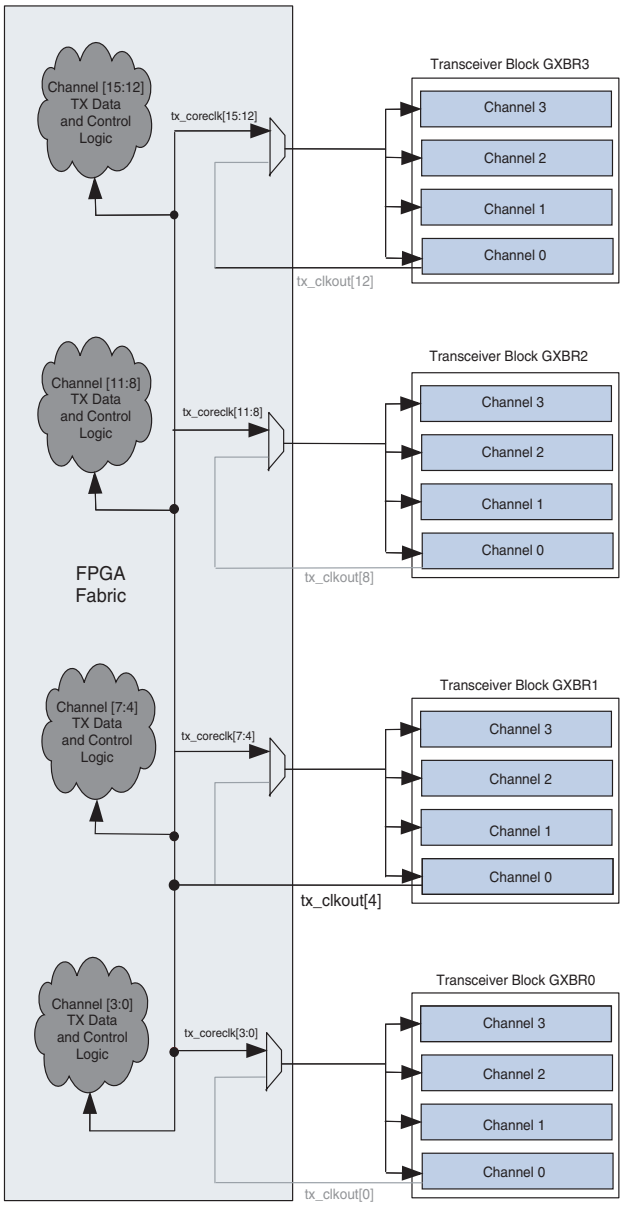


Table 2–19 lists the Quartus II assignments that you must make for the clocking scheme shown in Figure 2–38.

Table 2–19. Quartus II Assignments

From	top_level/top_xcvr_instance1/altgx_component/tx_clkout[4] (1)
To	tx_dataout[15..0]
Assignment Name	GXB 0 PPM Core Clock Setting
Value	ON

Note to Table 2–19:

(1) This is an example design hierarchy path for the tx_clkout[4] signal.

Configuration Example 3: Configuring Sixteen Channels Across Four Transceiver Blocks

This example relates to "User-Selected Receiver Phase Compensation FIFO Read Clock" on page 2–68.

Figure 2–40 shows 16 non-bonded channels without rate matcher located across four transceiver blocks. The incoming serial data to all 16 channels has a 0 PPM frequency difference with respect to each other. The rx_coreclk ports of all 16 channels are connected together and driven by rx_clkout[9] in transceiver block GXBR2. rx_clkout[9] also clocks the receiver data and status signals of all 16 channels in the FPGA fabric. With this clocking scheme, only one global, regional, or global and regional clock resource is used by rx_clkout[9].

Figure 2–40. Sixteen Channels Across Four Transceiver Blocks for Configuration Example 3

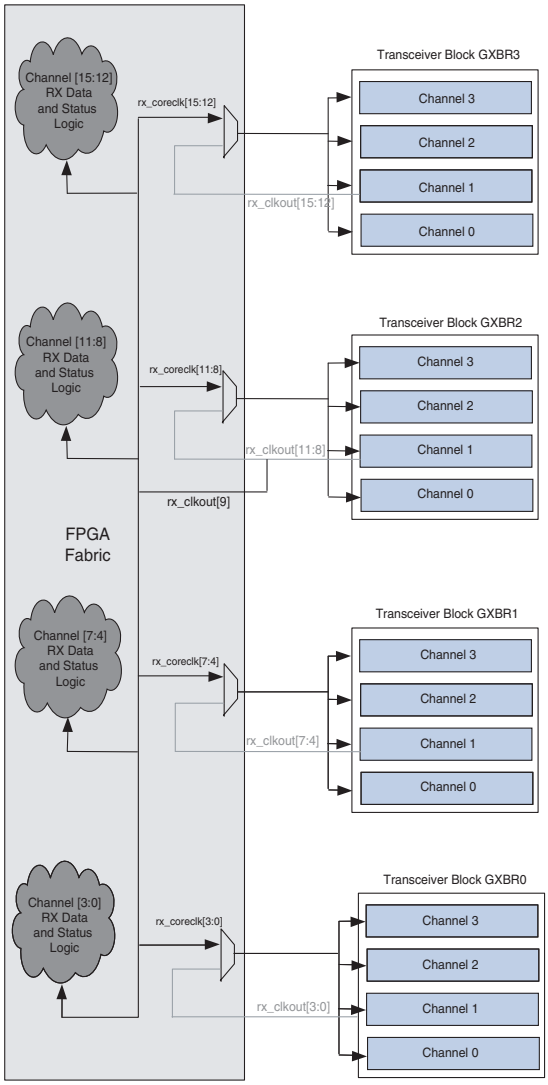


Table 2–20 lists the Quartus II assignments that you must make for the clocking scheme shown in Figure 2–40.

Table 2–20. Quartus II Assignments for Appendix Example 4

From	`top_level/top_xcvr_instance1/altgx_component/rx_clkout[9]` *(1)*
To	`rx_datain[15..0]`
Assignment Name	GXB 0 PPM Core Clock Setting
Value	ON

Note to Table 2–20:

(1) This is an example design hierarchy path for the `rx_clkout[9]` signal.

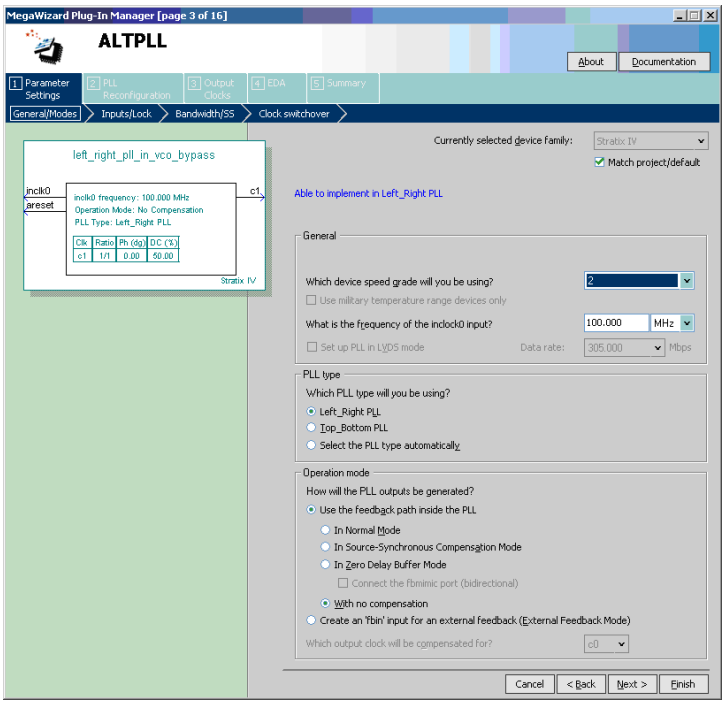
Configuration Example 4: Configuring Left and Right, Left, or Right PLL in VCO Bypass Mode

This example relates to "Left and Right, Left, or Right PLL in VCO Bypass Mode" on page 2–17.

To configure the left and right, left, or right PLL in VCO bypass mode, follow these steps:

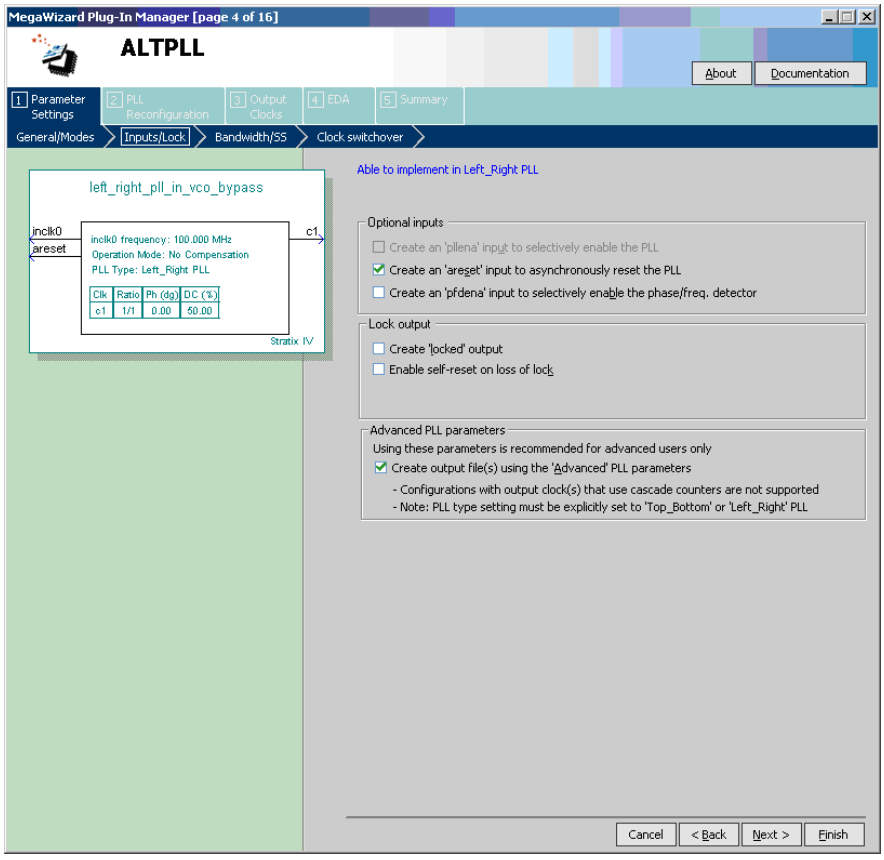
1. Under the **General/Modes** tab, enter the desired input reference clock frequency.
   a. Under **PLL Type**, select **Left_Right_PLL**.
   b. Under **Operation** mode, select the **With no compensation** option (Figure 2–41).

Figure 2–41. No Compensation Option Used for Configuration Example 4



2. Under the **Inputs/Lock** tab, select **Create output file(s) using the 'Advanced' PLL parameters** (Figure 2–42).

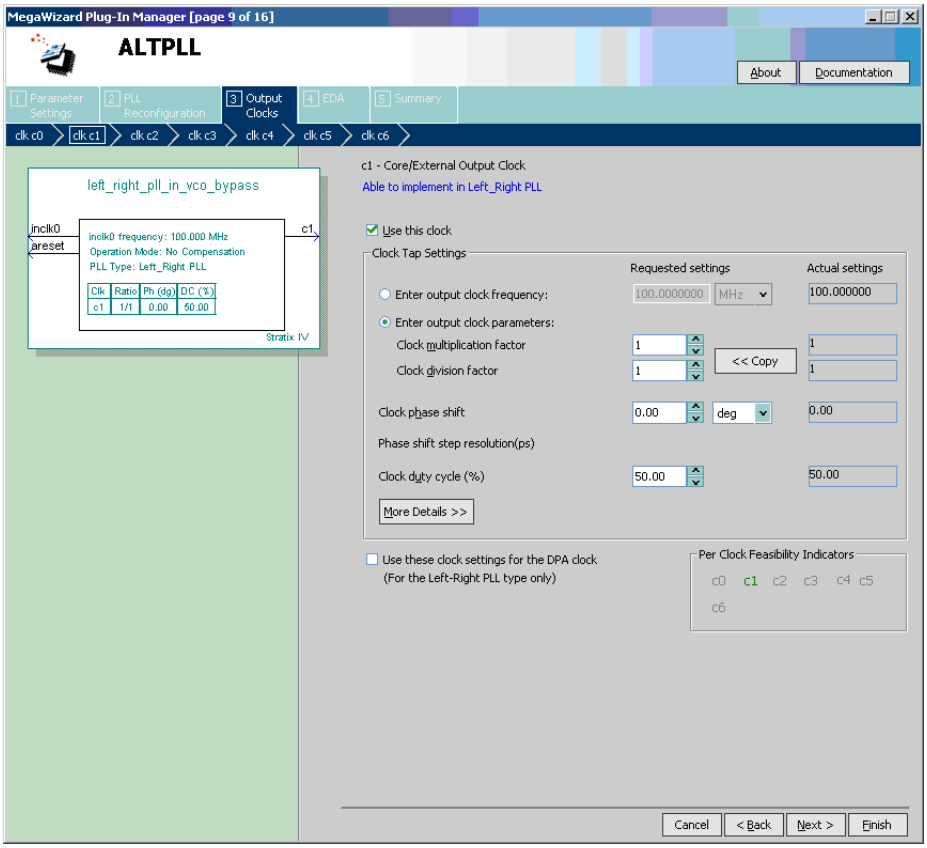
Figure 2–42. Create Output File(s) Using the 'Advanced' PLL Parameters Option Use for Configuration Example 4



3. Under the **Output Clocks** tab turn off **Use this clock** for clk c0.
4. Turn on **Use this clock** for clk c1 (Figure 2–43).

The **VCO bypass** option is only enabled for clock output c1.

Figure 2–43. Use This Clock Option Used for Configuration Example 4



5. Click **Finish** for the MegaWizard Plug-In Manager to generate the verilog **.v** file for the ALTPLL instantiation.
6. Next, from the command line, go to the directory where you have the ALTPLL instance files (**.v** or **.vhdl**) and type the following command:

```
qmegawiz -silent -wiz_override="c1_test_source=1,c1_mode=BYPASS,clk1_counter=C1" pll0.v
```

This command places your ALTPLL instance in VCO bypass mode. Revisit the **.v** or **.vhdl** file associated with the ALTPLL instance. Examine the file which is automatically updated to incorporate the PLL in a VCO bypass mode.

VCO bypass mode is not supported in the **.mif** file. Therefore, you can not manually modify the **.mif** file to set the PLL in VCO bypass mode.

7. Finally, connect `clk c1output` of the left and right, left, or right PLL to the input reference clock port of the ATX PLL used to generate the transceiver clocks.

Document Revision History

Table 2–21 lists the revision history for this chapter.

Table 2–21. Document Revision History

Date	Version	Changes
September 2012	3.4	■ Updated the "Non-Bonded Channel Configurations" section to close FB #65105.
December 2011	3.3	■ Updated Table 2–2. ■ Updated the "Left and Right, Left, or Right PLL in VCO Bypass Mode" section.
February 2011	3.2	■ Updated Table 2–4. ■ Updated Figure 2–7, Figure 2–18, Figure 2–19, Figure 2–20. ■ Updated the "Configuration Example 4: Configuring Left and Right, Left, or Right PLL in VCO Bypass Mode" section. ■ Applied new template. ■ Updated chapter title. ■ Applied new template.
March 2010	3.1	■ Updated Table 2–4. ■ Updated Figure 2–7, Figure 2–8, Figure 2–16, and Figure 2–21. ■ Updated the "Transceiver Channel Datapath Clocking" and "Configuration Example 3: Configuring Sixteen Channels Across Four Transceiver Blocks" sections. ■ Added a note to the "refclk0 and refclk1 Pins" section. ■ Changed "datapath clocks" to "datapath interface clocks". ■ Minor text edits.
November 2009	3.0	■ Added Figure 2–1, Figure 2–12,and Figure 2–13. ■ Added Table 2–1, Table 2–2, Table 2–8, and Table 2–2. ■ Updated Table 2–5 and Table 2–14. ■ Updated all graphics. ■ Updated all sections. ■ Added Stratix IV GT information. ■ Re-organized information. ■ Minor text edits.
June 2009	2.2	■ Updated Figure 2–5 and Figure 2–7. ■ Updated the "Transceiver Data Rates Supported in Basic (PMA Direct) Mode", ■ "FPGA Fabric PLLs-Transceiver PLLs Cascading in the 780-Pin Package", "FPGA Fabric PLLs-Transceiver PLLs Cascading in the 1152-Pin Package", sections. ■ Removed Table 2-5, Table 2-6, Table 2-7 ■ Removed Figure 2-17 and Figure 2-18. ■ Minor text edits.
March 2009	2.1	Minor updates.
November 2008	20	Update to chapter.
May 2008	1.0	Initial release.