

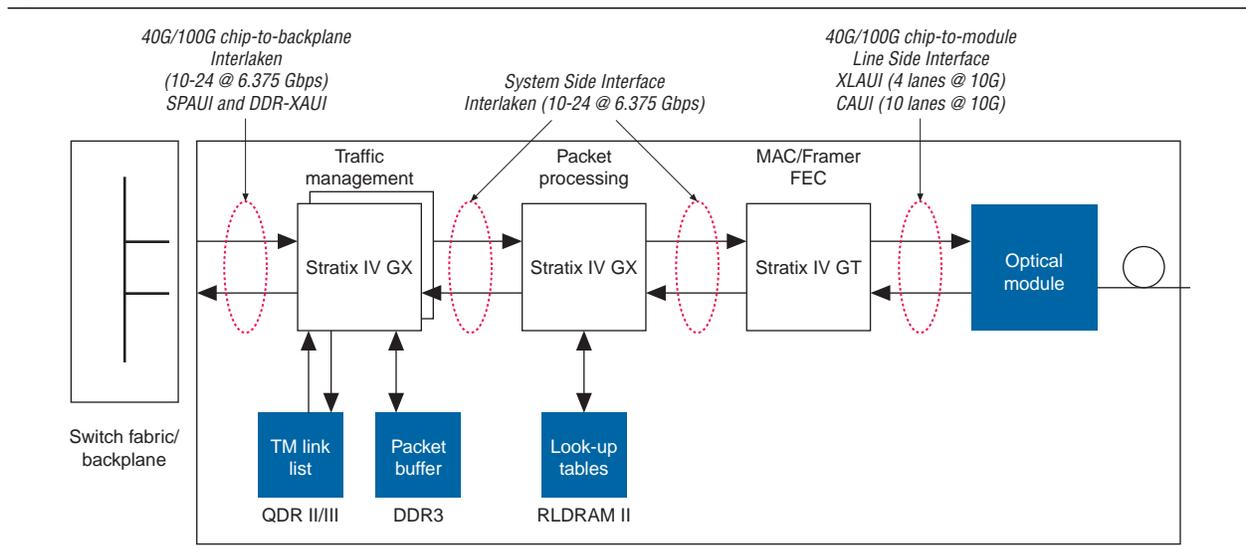
Introduction

This application note describes how to implement the Interlaken protocol in 40 Gbps and 100 Gbps applications with Stratix® IV transceivers (Stratix IV GX and Stratix IV GT devices). It describes in detail the transceiver configuration and clocking scheme to achieve the maximum channel utilization.

Interlaken is a high-speed communication protocol for high bandwidth (3 Gbps and higher) serial interface applications. The protocol allows multiple lanes to form a single Interlaken link, providing a scalable solution for 40 Gbps and 100 Gbps applications. The number of lanes for a given Interlaken link depends on the system requirements. The Interlaken protocol follows the CEI-6G-SR specification (www.oiforum.com). When multiple lanes are combined to form an Interlaken link, the protocol requires the skew between the transmitter lanes (measured at the transmitter output) to be within 67 unit intervals (UI).

Figure 1 shows an example of a 40G/100G line card solution that uses Altera® Stratix IV GT and Stratix IV GX devices.

Figure 1. Altera’s 40G/100G Line Card Solution



Stratix IV Transceiver Support for Interlaken Applications

Stratix IV transceivers offer an excellent solution for chip-to-chip or chip-to-backplane Interlaken applications. Stratix IV transceivers enable you to combine all the transceiver channels on one side of the device (24 in select device packages), providing an aggregate bandwidth of more than 100 Gbps. In device packages with less than 24 channels per side, you can combine the channels on both sides of the device and form a single Interlaken link that meets the transmitter skew requirements of the Interlaken protocol. This configuration is discussed in “Lane Borrowing” on page 11.

For more information about Stratix IV transceiver device packages and the transceiver channel counts, refer to the *Stratix IV Device Family Overview* chapter in volume 1 of the *Stratix IV Device Handbook*.

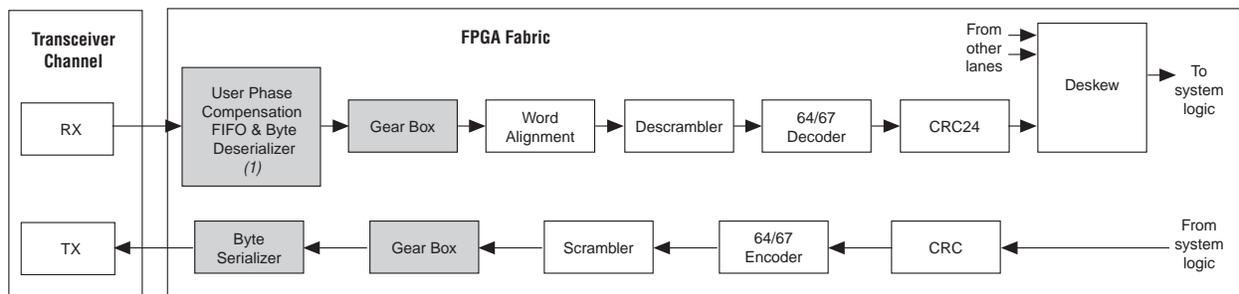
Altera offers Interlaken IP with the transceivers in Basic (PMA Direct mode).

Implementing 100 Gbps Interlaken Links

This section explains the method for configuring 24 transceiver channels to implement an Interlaken link. The channels—16 regular and 8 clock multiplication unit (CMU) channels—are assumed to be on the same side of the device and configured to run at a 6.5 Gbps data rate.

Figure 2 shows the functional blocks required to implement an Interlaken protocol application in Stratix IV transceivers.

Figure 2. Functional Blocks Required to Implement an Interlaken Application



Note to Figure 2:

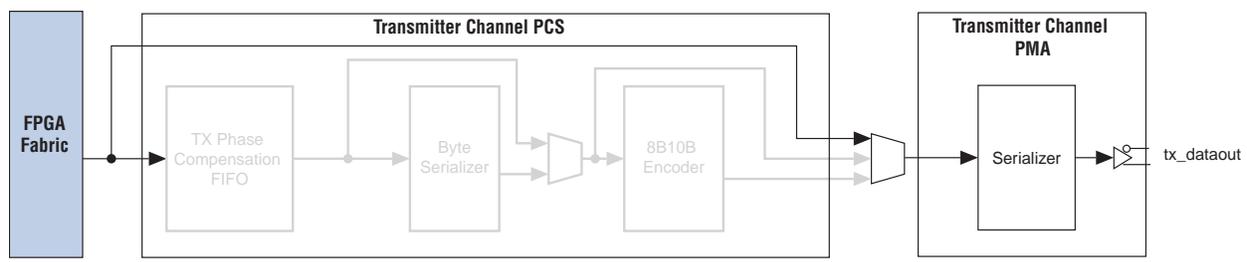
(1) A user phase compensation FIFO and byte deserializer are required for the CMU channels used as transceiver channels.

The gray blocks in Figure 2 are required to interface with the Stratix IV transceivers for this application. The functionality of the gear box and byte serialization logic is explained in “Gear Boxing Logic” on page 10.

Transceiver Channel Instances

To use all 24 transceiver channels, both the regular and the CMU channels must be configured. On the transmit side, to simplify the user logic and to maintain the transmit skew requirements, use the Basic (PMA Direct) mode on all the transmitter channels. In this mode, the parallel data from the FPGA fabric is directly provided to the transmitter serializer, as shown in Figure 3.

Figure 3. Transmitter Channel Datapath in Basic (PMA Direct) Mode



On the receive side, you can use one of the following configurations:

- Basic (PMA Direct) mode for all receiver channels
- Basic mode (enabling the phase compensation FIFO and the byte deserializer) in the regular channels and Basic (PMA Direct) mode in the CMU channels

Using the Basic (PMA Direct) mode simplifies the transceiver instantiation. It requires additional logic in the FPGA fabric to implement the byte deserializer and phase compensation FIFO. Also, because the data recovered from the individual channels is directly forwarded from the deserializer to the FPGA fabric, additional periphery clock (PCLK) routing resources and the manual placement of registers are required to meet the timing requirement between the FPGA fabric and the receiver interface.

Using the second option requires you to create multiple transceiver instantiations. With this configuration, you can use the byte deserializer and the phase compensation FIFO in the regular channels. The byte deserializer block reduces the FPGA fabric-receiver interface frequency. The phase compensation FIFO handles the 5 UI total wander requirement. For the CMU channels, implement the byte deserializer and the phase compensation FIFO in the FPGA fabric. The 8B10B decoder and the word aligner block (supporting a maximum of 20-bit word alignment) in the regular channels cannot be used, because the Interlaken protocol requires 64/67 encoding and a 64-bit synchronization pattern.

In this document, the transceiver instantiation and the clocking scheme for the second configuration option are discussed in detail. If you use the first method (all receiver channels in Basic [PMA Direct]), refer to [“FPGA Fabric–Receiver Interface Clocking” on page 8](#) for clocking scheme.

[Figure 4](#) and [Figure 5](#) illustrate the receive side datapath configuration for the regular and CMU channels.

Figure 4. Receive Side Datapath for Regular Channels

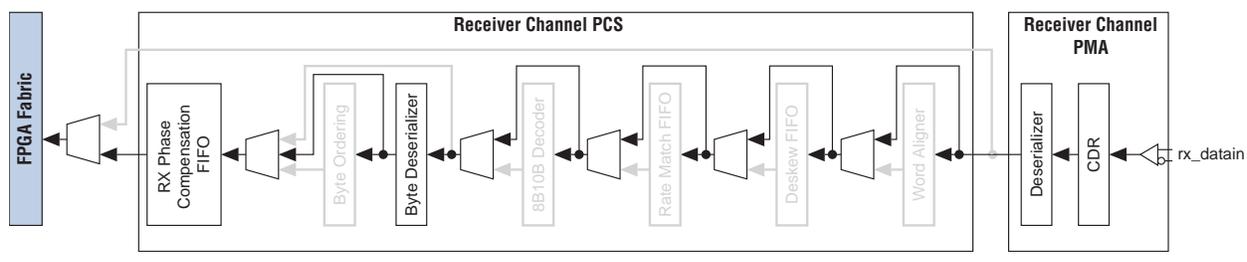
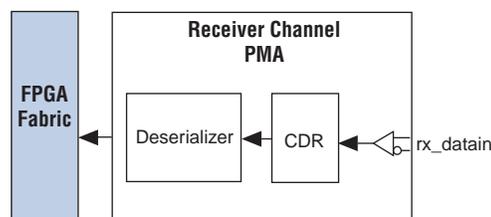


Figure 5. Receive Side Datapath for CMU Channels



Because multiple configurations are required to effectively use the transceiver functional blocks, create separate instances for regular and CMU channels and combine them in the design, as described in the following sections.

Transmitter Channel Instantiation

The transceiver configuration shown in this section assumes that 24 transceiver channels running at 6.5 Gbps are combined to form an Interlaken link.

In the ALTGX MegaWizard™ Plug-In Manager, create a Basic (PMA Direct) instance called INST_TX. Select the following options on the **General** tab:

- Which protocol will you be using?: **Basic (PMA Direct)**
- Which subprotocol will you be using?: **XN**
- What is the operation mode?: **Transmitter only**
- What is the number of channels?: **24**
- What is the deserializer block width?: **Double**
- What is the channel width?: **20**
- What is the effective data rate?: 6500

On the **PLL/Ports** tab, select the **Use Auxiliary Transmitter (ATX) PLL** option.



Not all the parameters in the ALTGX MegaWizard Plug-In Manager are listed. Refer to the *ALTGX Transceiver Setup Guide* in volume 3 of the *Stratix IV Device Handbook* for additional information about the options and settings in the ALTGX MegaWizard Plug-In Manager interface.

Receiver Channel Instantiation

In the ALTGX MegaWizard Plug-In Manager, create a Basic (PMA Direct) instance called INST_RX_CMU for the eight CMU channels. Select the following options on the **General** tab:

- Which protocol will you be using?: **Basic (PMA Direct)**
- Which subprotocol will you be using?: none
- What is the operation mode?: **Receiver only**
- What is the number of channels?: **8**
- What is the deserializer block width?: **Double**
- What is the channel width?: **20**
- What is the effective data rate?: 6500

Assign the rx_datain ports of this instance to the receive pins that correspond to CMU channels. The receive input pins for the CMU channels are named REFCLK_[L,R][0:7]p, GXB_CMURX_[L,R][0:7]p and REFCLK_[L,R][0:7]n, GXB_CMURX_[L,R][0:7]n.

Create a Basic instance called INST_RX_Regular for the 16 regular channels. Select the following options on the **General** tab:

- Which protocol will you be using?: **Basic**

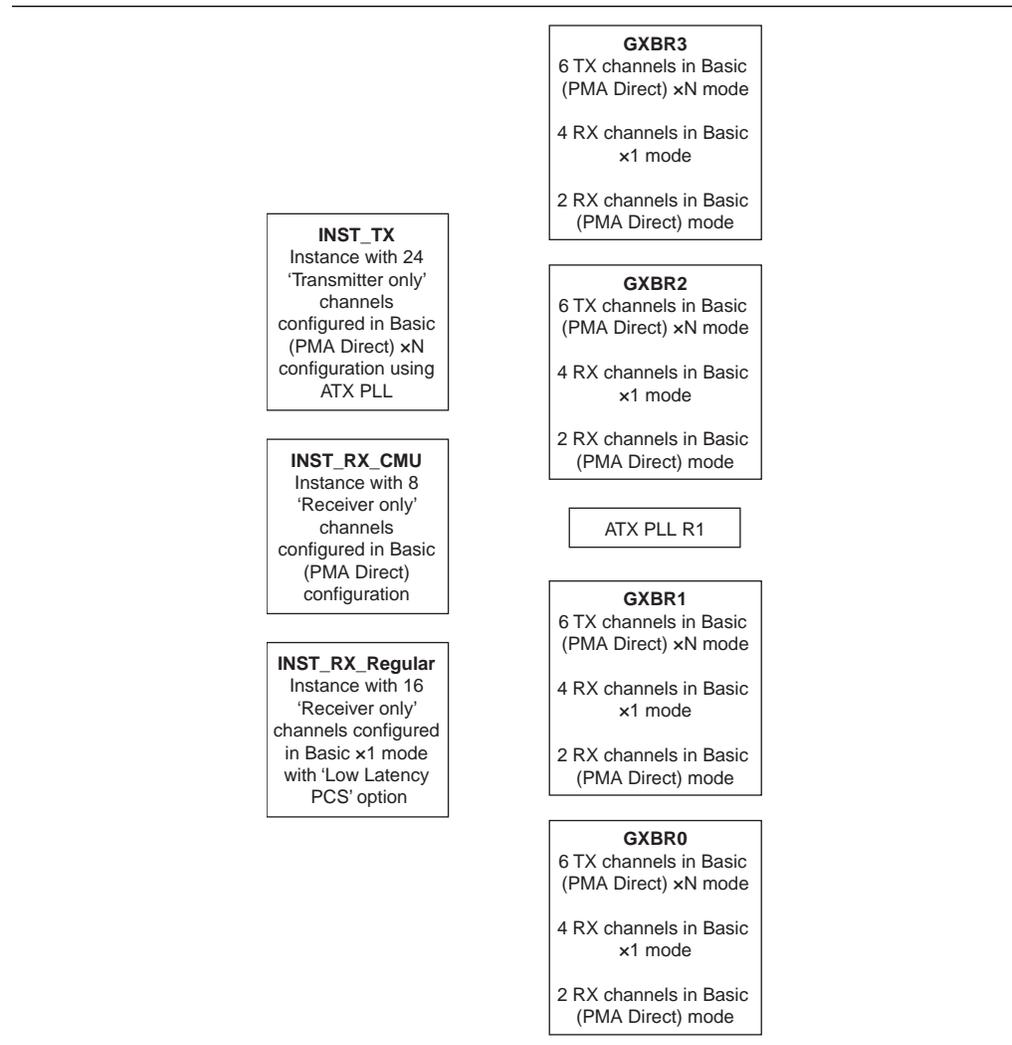
- Which subprotocol will you be using?: none
- What is the operation mode?: **Receiver only**
- What is the number of channels?: **16**
- What is the deserializer block width?: **Double**
- What is the channel width?: **40**
- What is the effective data rate?: 6500

On the **Basic/8B10B** tab, turn on **Enable Low Latency PCS mode**. (If this option is selected, the byte serializer/deserializer and TX/RX phase compensation FIFO are the only PCS functional blocks enabled in the transceiver hard macro.)

On the **Ports/Cal Blk** tab, turn on **Create 'rx_coreclk' port** to connect to the read clock of the RX phase compensation FIFO. The use of `rx_coreclk` is explained in ["Interface Clocking"](#) on page 7.

Assign the `rx_datain` ports of this instance to the pins that correspond to regular channels.

[Figure 6](#) illustrates the logical instantiations and physical placements of the transceiver channels. The `GXBR0`, `GXBR1`, `GXBR2`, and `GXBR3` blocks show the transceiver blocks on the right side of the device. The `ATX PLL R1` refers to the ATX PLL.

Figure 6. Logical Instantiations and Physical Channel Placements

Transmit Side Clock Generation and Input Reference Clocks

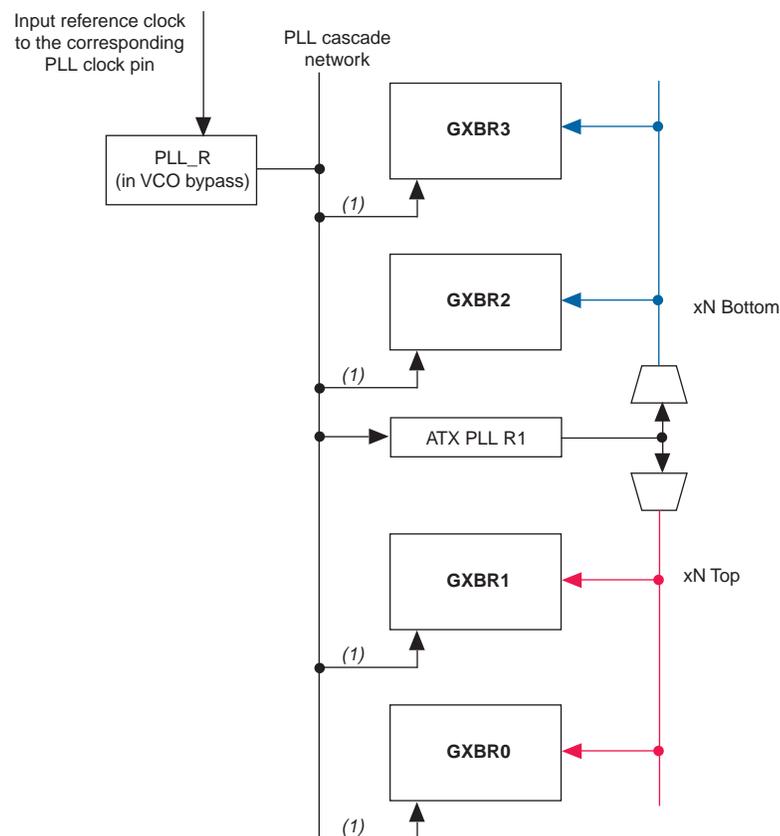
Because all the transceiver channels, including the CMU channels, are configured as transceiver channels, you must use the ATX PLL block to generate a high-speed clock to all the transmitter channels. Also, all the dedicated transceiver `refclk` pins on the side are used as receive data pins. Therefore, use an FPGA clock pin as an input reference clock source. To minimize jitter, use a Left/Right PLL in VCO Bypass mode to connect this input reference clock to the ATX PLL through the dedicated PLL cascade line (Figure 7).

 For more information about PLL cascading using the VCO Bypass option, refer to “Left and Right, Left, or Right PLL in VCO Bypass Mode” in the *Stratix IV Transceiver Clocking* chapter in volume 2 of the *Stratix IV Device Handbook*.

Altera recommends you use the ATX PLL R1 or the ATX PLL L1 (depending on the side used) to generate a clock for the transmit side to minimize transmit skew across the channels.

-  For more information about the ATX PLL block, refer to “Auxiliary Transmit (ATX) PLL Block” in the *Stratix IV Transceiver Architecture* chapter in volume 2 of the *Stratix IV Device Handbook*.
-  When multiple transceiver channels are bonded with the xN lines, you might have to change the number of transceiver channels bonded and the transceiver data rate transceiver supply levels— $V_{CCT_L/R}$, $V_{CCR_L/R}$, $V_{CCL_L/R}$ (applicable for Stratix IV GX transceivers).
-  For more information about the power supply requirements, refer to “x8 and xN Clock Line Timing Issue for Transceivers” for production devices in the *Stratix IV GX Errata Sheet* and *Stratix IV GT Errata Sheet*.

Figure 7. Input Reference Clocking Using Left/Right PLL in VCO Bypass Mode



Note to Figure 7:

(1) The input clock connects to the RX CDR in each channel.

Interface Clocking

The interface clocking requirements are different for the transmit and receive sides. The transmit side requires a specific clocking scheme to meet the FPGA fabric–transmitter PMA interface timing requirements. On the receive side, each channel has its recovered clock. To optimize the clock routing resources, Altera recommends the following clocking schemes.

FPGA Fabric–Transmitter PMA Interface Clocking

Because all the transmitter channels are configured in Basic (PMA Direct) mode at 6.5 Gbps, the FPGA fabric-transmitter interface clock frequency is 325 MHz. To meet the FPGA fabric-transmitter PMA interface timing, Altera recommends using pipeline registers between the transmitter PMA and the FPGA Fabric interface.

For more information about implementing and clocking these registers, refer to Register Method in *AN 580: Achieving Timing Closure in Basic (PMA Direct) Functional Mode*.

FPGA Fabric–Receiver Interface Clocking

The receive side clocking is different from the transmit side because the configuration contains a combination of regular channel instances (with phase compensation FIFOs) and CMU channel instances (without phase compensation FIFOs).

To meet timing between the receiver PMA and the FPGA fabric, capture the received parallel data in the FPGA core with positive-edge triggered registers.

Use the following multi-cycle constraint:

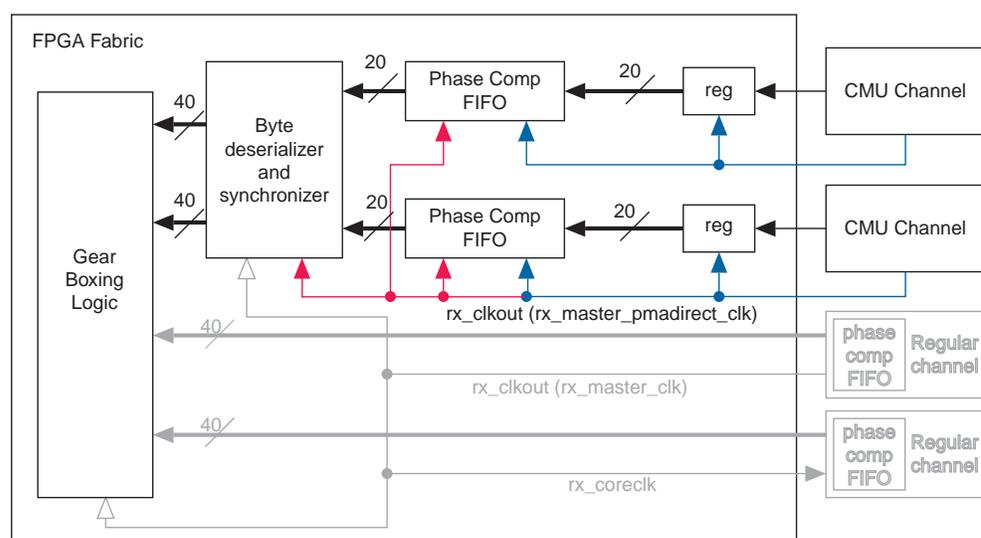
```
set_multicycle_path -setup -from [get_registers rx_data_reg*] 0
```

`rx_data_reg` are the registers used to capture the RX data from the `rx_dataout` port of the receiver PMA in the FPGA Fabric.

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

You can minimize the clock resource utilization for the CMU channels and the regular channels. For each of the CMU channels, create an external phase compensation FIFO and use the `rx_clkout` port of one CMU channel to clock the read side of the external phase compensation FIFO. **Figure 8** illustrates the receive side clocking for the CMU channels.

Figure 8. Receive Side Clocking for CMU Channels (Note 1)



Note to Figure 8:

(1) To simplify the illustration, not all CMU channels are shown.

Therefore, design your deskew logic in the FPGA fabric taking these uncertainty and latency values into consideration.

Gear Boxing Logic

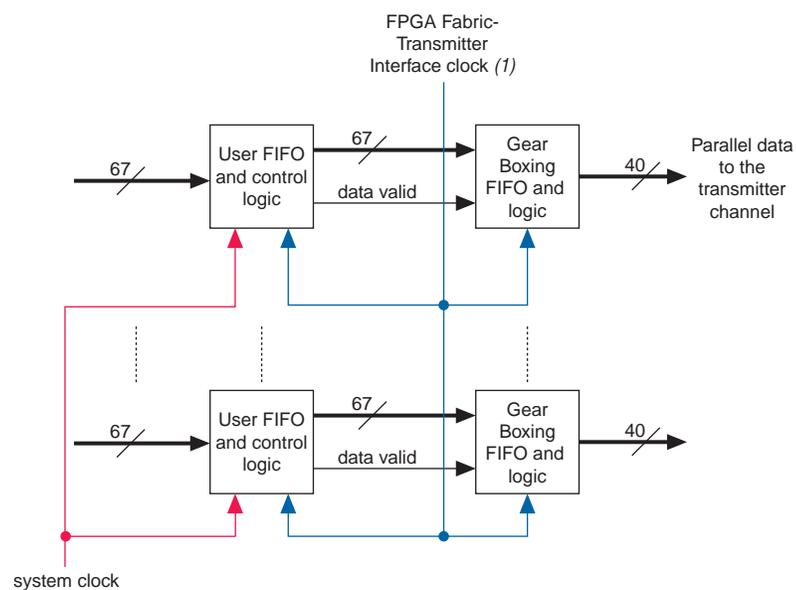
The Interlaken protocol requires 64/67 encoding-decoding on the transmit and receive datapath. The configuration discussed in [“Transceiver Channel Instances” on page 2](#) requires a 20-bit datapath interface for all the transmitter channels. To reduce the FPGA fabric-transmitter interface frequency of 325 MHz (6.5 Gbps/20), implement a byte serializer in the FPGA fabric to reduce the operating frequency of the transmitter side user logic to 162.5 MHz. The byte serializer effectively doubles the datapath width. On the receive side, the CMU channels are configured to the maximum datapath interface width of 20 bits. The regular channels have a 40-bit interface. Implement a byte deserializer in the FPGA fabric for the receive side interface for the CMU channels. To interface this 40-bit datapath to the 64/67-bit datapath domain, you must implement a gear boxing logic in the FPGA fabric.

Transmit Side Gear Boxing

[Figure 10](#) shows the user logic associated with gear boxing on the transmit side. The FIFO in the datapath provides the compensation between the two clock domains with different datapath widths.

As shown in [Figure 10](#), the write side and read side of the user FIFO are assumed to be generated by the same clock source. The read side of the user FIFO operates at the same clock frequency as the gear boxing FIFO and logic block. When the user FIFO becomes empty due to being read at the higher clock frequency, the control logic in the user FIFO deasserts the `data valid` signal to indicate to the gear boxing logic that the data can be ignored. With this method, the throughput of the link is unaffected.

Figure 10. Transmit Side Gear Boxing Interface



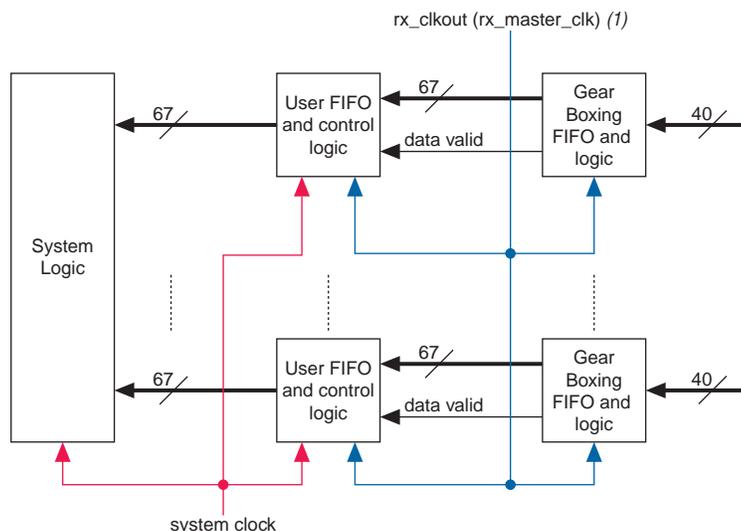
Note to Figure 10:

- (1) Refer to [“FPGA Fabric–Transmitter PMA Interface Clocking” on page 8](#) for more information about the source of this clock.

Receive Side Gear Boxing

The receive side gear boxing logic shown in [Figure 11](#) is similar to that of the transmit side. The gear boxing FIFO logic and the write side of the user FIFO for all the receiver channels operate in the `rx_clkout` (`rx_master_clk`) domain. This method minimizes the FPGA clock resource usage. Refer to [Figure 9 on page 9](#) for the source of `rx_clkout`. The frequency of the system clock depends on the implementation and the size of the user FIFO logic. Consider running the system clock at a faster rate than the `rx_master_clk` domain to avoid clock rate compensation between the system and the `rx_master_clk` domains.

Figure 11. Receive Side Gear Boxing Logic

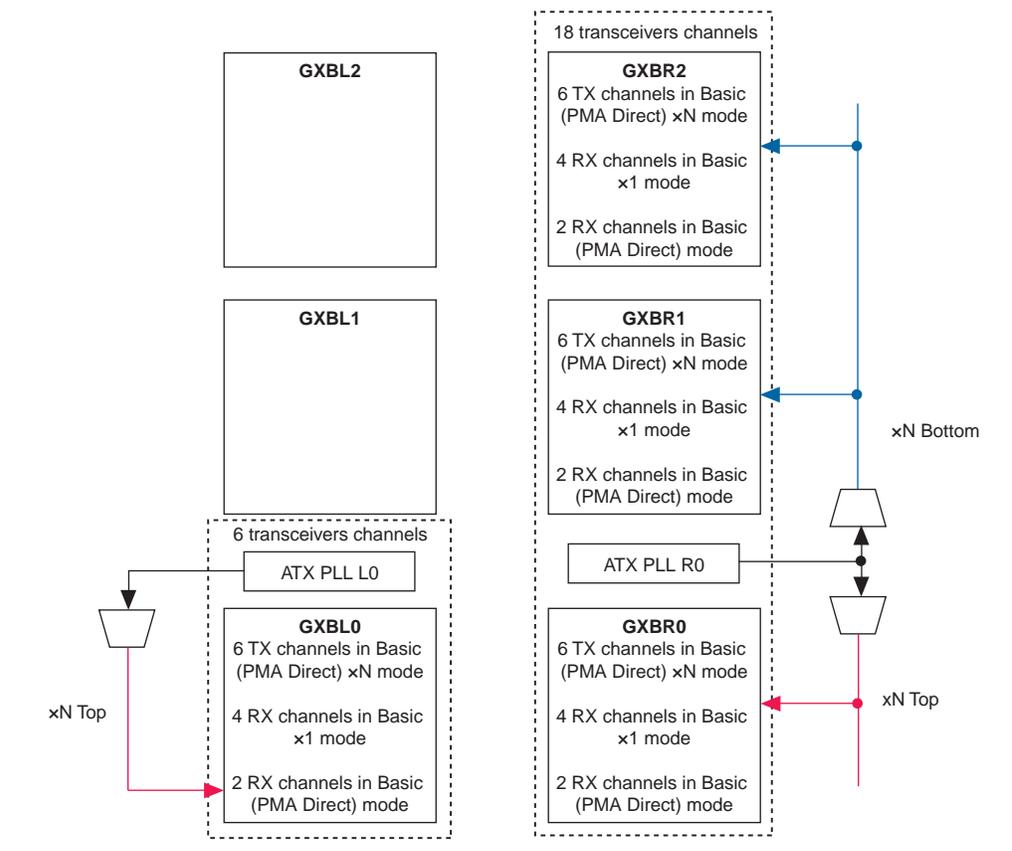


Note to Figure 11:

(1) Refer to [Figure 9 on page 9](#) for the source of this clock.

Lane Borrowing

In Stratix IV device packages that have fewer than 24 transceiver channels per side, you can use the transceiver channels on the other side of the device to implement 100 Gbps Interlaken links. The transmitter channels on the two sides of the device are clocked by separate PLLs. [Figure 12](#) shows an example of 24-channel Interlaken link implementation by using channels on both sides of the device.

Figure 12. Using Channels from Both Sides to Implement a 100 Gbps Interlaken Link

If you design a system that uses transceiver channels on both sides of the device, consider the following parameters:

- Input reference clock from dedicated `refclk` pins—Route the clock trace from the input reference clock source with separate physical traces to transceiver PLLs on both sides of the device. This routing must occur at the board level because the input reference clock pins that are connected to the transceivers on one side cannot be routed through the device to clock the transceiver PLLs on the other side.
- Channel Trace length—Match the trace lengths of the channels on opposite sides of the device to meet the 67 UI skew requirements.
- Ensure that additional device resources such as clock routing and PLLs are available to implement the design.

Implementing 40 Gbps Interlaken Links

To implement the Interlaken protocol for 40 Gbps bandwidth using Stratix IV transceivers, use eight regular transceiver channels at 6.5 Gbps with a 40-bit datapath. The phase compensation FIFO and the byte serializer in the transceiver PCS can be used. Create the transceiver instance with a Basic $\times 8$ configuration. In this configuration, the transmitter PCS of all the channels are synchronized and the CMU0

channel of the master transceiver block provides high-speed serial and low-speed parallel clocks to all the transmitter channels. This configuration provides a simple clocking interface to the FPGA fabric and reduces the transmit lane-to-lane skew compared to a non-bonded configuration. Create the Basic $\times 8$ configuration by selecting the following options on the **General** tab:

- Which protocol will you be using?: **Basic**
- Which subprotocol will you be using?: **$\times 8$**
- What is the operation mode?: **Transmitter and Receiver**
- What is the number of channels?: **8**
- What is the deserializer block width?: **Double**
- What is the channel width?: **40**
- What is the effective data rate?: **6500**

On the **Basic/8B10B** tab, turn on **Enable Low Latency PCS mode**.

On the **Ports/Cal Blk** tab, turn on **Create 'rx_coreclk' port**.

The CMU PLL in the master transceiver block of this $\times 8$ link generates high-speed serial and low-speed parallel clocks for the transmitter channels. Only the selected adjacent transceiver blocks shown in [Table 1](#) can be paired for the $\times 8$ configuration.

Table 1. Adjacent Transceiver Blocks for the $\times 8$ Configuration

Number of Transceiver Blocks Per Side	Transceiver Blocks that Can Be Set Up in $\times 8$ Configuration	Master Transceiver Block
2	QL0 and QL1	QL0
	QR0 and QR1	QR0
3	QL0 and QL1	QL0
	QR0 and QR1	QR0
4	QL0 and QL1	QL0
	QL2 and QL3	QL2
	QR0 and QR1	QR0
	QR2 and QR3	QR2

Use one of the `coreclkout [1:0]` ports of the transceiver instance to clock the transmit side logic. On the receive side, connect the `rx_clkout` of one channel to the `rx_coreclk` ports of the other channels. To compile a design that uses `rx_coreclk` for clocking, use the Quartus II Assignment Editor (on the Assignments menu) to set a "GXB 0PPM core clocking setting" assignment. Refer to ["FPGA Fabric-Receiver Interface Clocking" on page 8](#) for additional information.



For more information about the clocking scheme for basic $\times 8$ mode, refer to "FPGA Fabric-Transmitter Interface Clocking" and "FPGA Fabric-Receiver Interface Clocking" in the [Stratix IV Transceiver Clocking](#) chapter in volume 2 of the *Stratix IV Device Handbook*.

-  When multiple transceiver channels are bonded with the $\times N$ lines, you might have to change the number of transceiver channels bonded and the transceiver data rate transceiver supply levels— $V_{CCT_L/R}$, $V_{CCR_L/R}$, $V_{CCL_L/R}$ (applicable for Stratix IV GX transceivers).
-  For more information about the power supply requirements, refer to “ $\times 8$ and $\times N$ Clock Line Timing Issue for Transceivers” for production devices in the *Stratix IV GX Errata Sheet* and *Stratix IV GT Errata Sheet*.

Summary

In this application note, specific transceiver configurations and a clocking scheme are recommended to implement the Interlaken protocol. Stratix IV transceivers supported by the Quartus II software offer flexible solutions to implement the Interlaken protocol between chip-to-chip and chip-to-backplane applications.

Document Revision History

Table 2 shows the revision history for this application note.

Table 2. Document Revision History

Date and Document Version	Changes Made
December 2009 v1.1	Updated the following sections: <ul style="list-style-type: none"> ■ “Transmit Side Clock Generation and Input Reference Clocks” on page 6 ■ “FPGA Fabric–Receiver Interface Clocking” on page 8 ■ “Implementing 40 Gbps Interlaken Links” on page 12
June 2009 v1.0	Initial release.



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera’s standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

