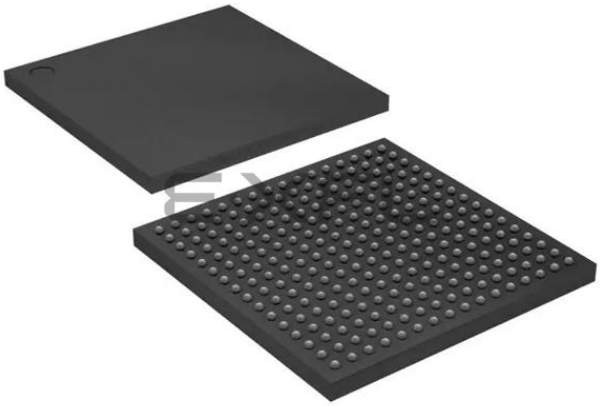


Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

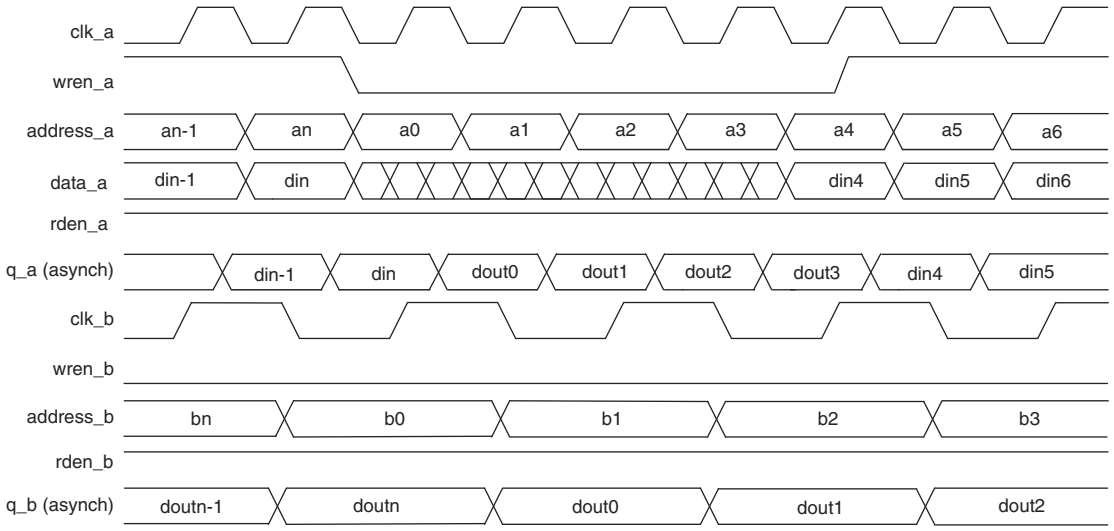
The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Active
Number of LABs/CLBs	1395
Number of Logic Elements/Cells	22320
Total RAM Bits	608256
Number of I/O	153
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce22f17i7

In true dual-port mode, you can access any memory location at any time from either port A or port B. However, when accessing the same memory location from both ports, you must avoid possible write conflicts. When you attempt to write to the same address location from both ports at the same time, a write conflict happens. This results in unknown data being stored to that address location. There is no conflict resolution circuitry built into the Cyclone IV devices M9K memory blocks. You must handle address conflicts external to the RAM block.

Figure 3–11 shows true dual-port timing waveforms for the write operation at port A and read operation at port B. Registering the outputs of the RAM simply delays the `q` outputs by one clock cycle.

Figure 3–11. Cyclone IV Devices True Dual-Port Timing Waveform



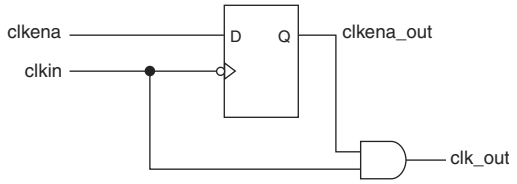
Shift Register Mode

Cyclone IV devices M9K memory blocks can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto-correlation and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift register block, which saves logic cell and routing resources.

The size of a ($w \times m \times n$) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n), and must be less than or equal to the maximum number of memory bits, which is 9,216 bits. In addition, the size of ($w \times n$) must be less than or equal to the maximum width of the block, which is 36 bits. If you need a larger shift register, you can cascade the M9K memory blocks.

Figure 5–7 shows how to implement the clkena signal with a single register.

Figure 5–7. clkena Implementation

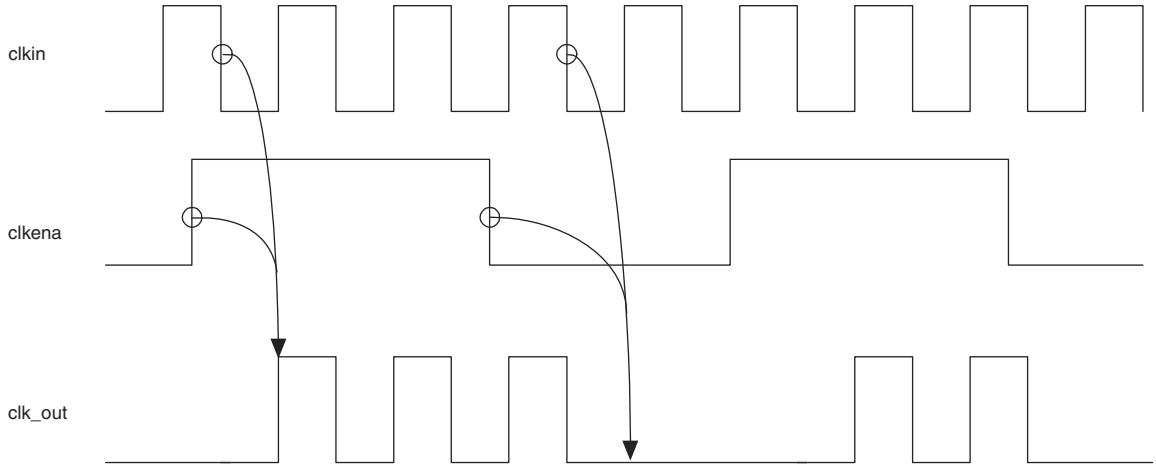


The clkena circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5–7.

Figure 5–8 shows the waveform example for a clock output enable. The clkena signal is sampled on the falling edge of the clock (clkin).

This feature is useful for applications that require low power or sleep mode.

Figure 5–8. clkena Implementation: Output Enable



The clkena signal can also disable clock outputs if the system is not tolerant to frequency overshoot during PLL resynchronization.

Altera recommends using the clkena signals when switching the clock source to the PLLs or the GCLK. The recommended sequence is:

1. Disable the primary output clock by de-asserting the clkena signal.
2. Switch to the secondary clock using the dynamic select signals of the clock control block.
3. Allow some clock cycles of the secondary clock to pass before reasserting the clkena signal. The exact number of clock cycles you must wait before enabling the secondary clock is design-dependent. You can build custom logic to ensure glitch-free transition when switching between different clock sources.

Section II. I/O Interfaces

This section provides information about Cyclone® IV device family I/O features and high-speed differential and external memory interfaces.

This section includes the following chapters:

- Chapter 6, I/O Features in Cyclone IV Devices
- Chapter 7, External Memory Interfaces in Cyclone IV Devices

Revision History

Refer to each chapter for its own specific revision history. For information about when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the complete handbook.

RSDS, Mini-LVDS, and PPDS I/O Standard Support in Cyclone IV Devices

The RSDS, mini-LVDS, and PPDS I/O standards are used in chip-to-chip applications between the timing controller and the column drivers on the display panels such as LCD monitor panels and LCD televisions. Cyclone IV devices meet the National Semiconductor Corporation RSDS Interface Specification, Texas Instruments mini-LVDS Interface Specification, and National Semiconductor Corporation PPDS Interface Specification to support RSDS, mini-LVDS and PPDS output standards, respectively.

For Cyclone IV devices RSDS, mini-LVDS, and PPDS output electrical specifications, refer to the *Cyclone IV Device Datasheet* chapter.

For more information about the RSDS I/O standard, refer to the RSDS specification from the National Semiconductor website (www.national.com).

Designing with RSDS, Mini-LVDS, and PPDS

Cyclone IV I/O banks support RSDS, mini-LVDS, and PPDS output standards. The right I/O banks support true RSDS, mini-LVDS, and PPDS transmitters. On the top and bottom I/O banks, RSDS, mini-LVDS, and PPDS transmitters are supported using two single-ended output buffers with external resistors. The two single-ended output buffers are programmed to have opposite polarity.

Figure 6–15 shows an RSDS, mini-LVDS, or PPDS interface with a true output buffer.

Figure 6–15. Cyclone IV Devices RSDS, Mini-LVDS, or PPDS Interface with True Output Buffer on the Right I/O Banks

Figure 6–16 shows an RSDS, mini-LVDS, or PPDS interface with two single-ended output buffers and external resistors.

Figure 6–16. RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks *(1)*

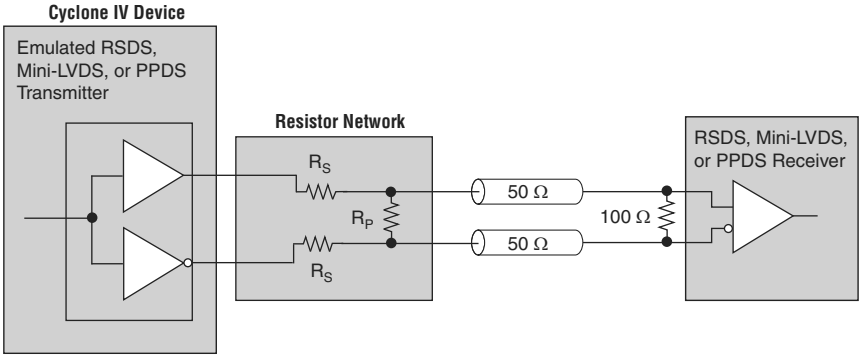


Table 7–2. Cyclone IV E Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 2 of 3)

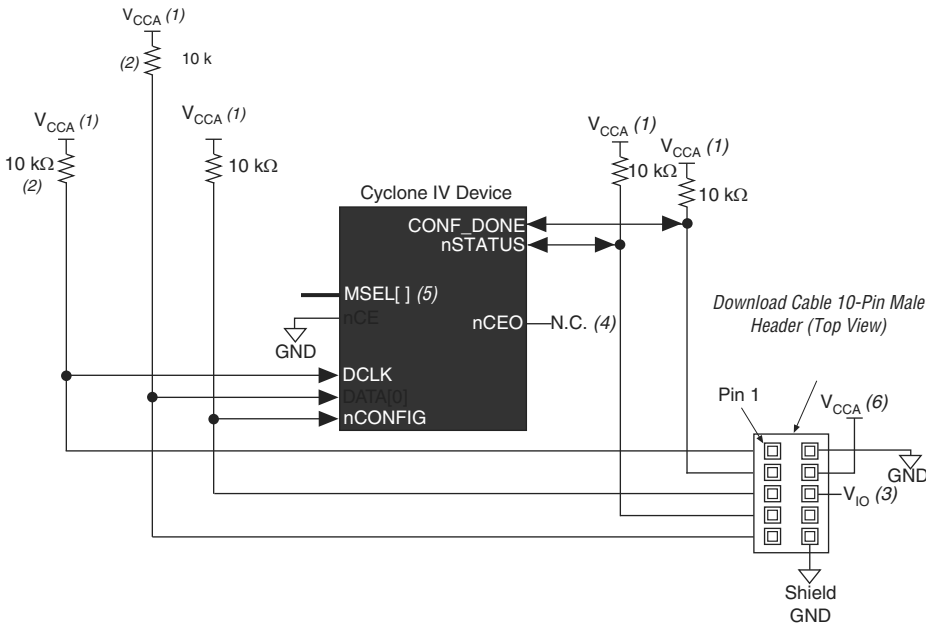
Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CE22	144-pin EQFP	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Bottom (1), (3)	1	0	0	0	—	—
		Top (1), (4)	1	0	0	0	—	—
	256-pin UBGA	Left (1)	1	1	0	0	—	—
		Right (2)	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
	256-pin FBGA	Left (1)	1	1	0	0	—	—
		Right (2)	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
EP4CE30	324-pin FBGA	Left (1)	2	2	1	1	0	0
		Right (2)	2	2	1	1	0	0
		Bottom	2	2	1	1	0	0
		Top	2	2	1	1	0	0
EP4CE30 EP4CE115	484-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	4	4	2	2	1	1
		Top	4	4	2	2	1	1
	780-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	6	6	2	2	1	1
		Top	6	6	2	2	1	1
EP4CE40	324-pin FBGA	Left	2	2	1	1	0	0
		Right	2	2	1	1	0	0
		Bottom	2	2	1	1	0	0
		Top	2	2	1	1	0	0

The programming hardware or download cable then places the configuration data one bit at a time on the DATA[0] pin of the device. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

When you use a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software if an error occurs. Additionally, the **Enable user-supplied start-up clock (CLKUSR)** option has no effect on device initialization, because this option is disabled in the **.sof** when programming the device with the Quartus II Programmer and download cable. Therefore, if you turn on the **CLKUSR** option, you do not have to provide a clock on CLKUSR when you configure the device with the Quartus II Programmer and a download cable.

Figure 8–17 shows PS configuration for Cyclone IV devices with a download cable.

Figure 8–17. PS Configuration Using a Download Cable



Notes to Figure 8–17:

(1) You must connect the pull-up resistor to the same supply voltage as the V_{CCA} supply.

(2) The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This is to ensure that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. With the USB-Blaster, ByteBlaster II, ByteBlaster MV, and EthernetBlaster, this pin is a no connect.

(4) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

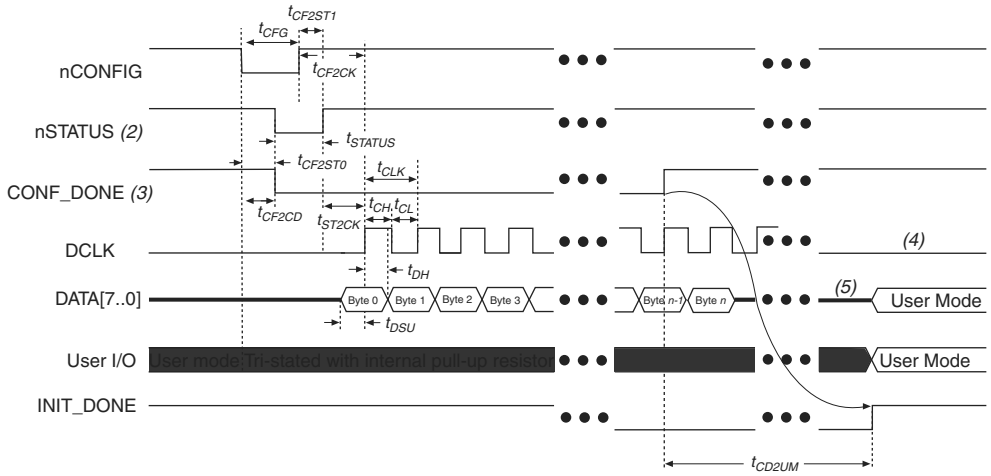
(5) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9 for PS configuration schemes. Connect the MSEL pins directly to V_{CCA} or GND.

(6) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

FPP Configuration Timing

Figure 8–22 shows the timing waveform for the FPP configuration when using an external host.

Figure 8–22. FPP Configuration Timing Waveform (1)



Notes to Figure 8–22:

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic-high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) After power up, the Cyclone IV device holds nSTATUS low during POR delay.

(3) After power up, before and during configuration, CONF_DONE is low.

(4) Do not leave DCLK floating after configuration. It must be driven high or low, whichever is more convenient.

(5) DATA[7..0] is available as a user I/O pin after configuration; the state of the pin depends on the dual-purpose pin settings.

Table 8–13 lists the FPP configuration timing parameters for Cyclone IV devices.

Table 8–13. FPP Timing Parameters for Cyclone IV Devices (Part 1 of 2)

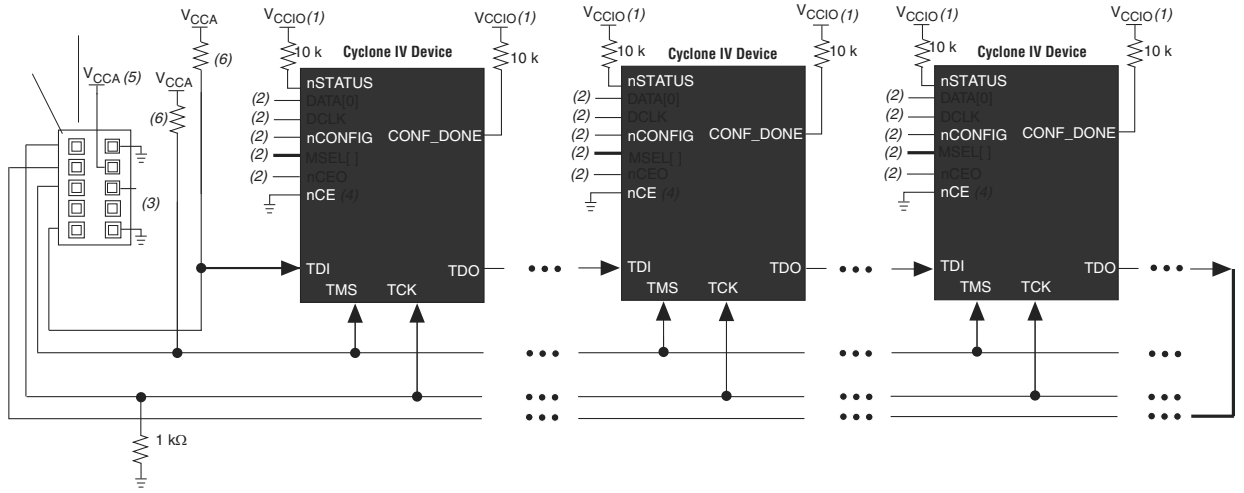
Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV (1)	Cyclone IV E (2)	Cyclone IV (1)	Cyclone IV E (2)	
t_{CF2CD}	nCONFIG low to CONF_DONE low	—		500		ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—		500		ns
t_{CFG}	nCONFIG low pulse width	500		—		ns
t_{STATUS}	nSTATUS low pulse width	45		230 (3)		µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—		230 (4)		µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	230 (3)		—		µs

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system with JTAG BST circuitry. Figure 8–25 and Figure 8–26 show multi-device JTAG configuration.

For devices using 2.5-, 3.0-, and 3.3-V V_{CCIO} supply, you must refer to Figure 8–25. All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot when using 2.5-, 3.0-, and 3.3- V V_{CCIO} supply. You must power up the V_{CC} of the download cable with a 2.5-V V_{CCA} supply. For device using V_{CCIO} of 1.2, 1.5 V, and 1.8 V, refer to Figure 8–26. You can power up the V_{CC} of the download cable with the supply from V_{CCIO}.

Figure 8–25. JTAG Configuration of Multiple Devices Using a Download Cable (2.5, 3.0, and 3.3-V V_{CCIO} Powering the JTAG Pins)



Notes to Figure 8–25:

(1) Connect these pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.

(2) Connect the nCONFIG and MSEL pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect the nCONFIG pin to logic-high and the MSEL pins to GND. In addition, pull DCLK and DATA[0] to either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the ByteBlasterMV cable, this pin is a no connect. In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

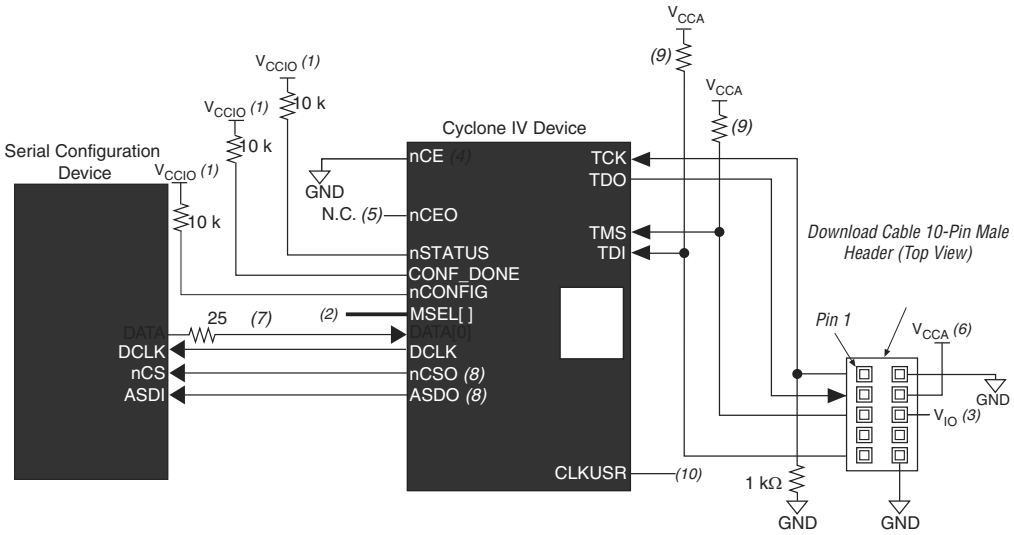
(4) You must connect the nCE pin to GND or driven low for successful JTAG configuration.

(5) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(6) Resistor value can vary from 1 kΩ to 10 kΩ.

If you configure a master device with an SFL design, the master device enters user mode even though the slave devices in the multiple device chain are not being configured. The master device enters user mode with a SFL design even though the CONF_DONE signal is externally held low by the other slave devices in chain. Figure 8–29 shows the JTAG configuration of a single Cyclone IV device with a SFL design.

Figure 8–29. Programming Serial Configuration Devices In-System Using the JTAG Interface



Notes to Figure 8–29:

(1) Connect the pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.

(2) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. The V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

(4) You must connect the nCE pin to GND or driven low for successful JTAG configuration.

(5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(6) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5- V V_{CCA} supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(7) Connect the series resistor at the near end of the serial configuration device.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(9) Resistor value can vary from 1 kΩ to 10 kΩ.

(10) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

ISP of the Configuration Device

In the second stage, the SFL design in the master device allows you to write the configuration data for the device chain into the serial configuration device with the Cyclone IV device JTAG interface. The JTAG interface sends the programming data for the serial configuration device to the Cyclone IV device first. The Cyclone IV device then uses the ASMI pins to send the data to the serial configuration device.

In some applications, it is necessary for a device to wake up very quickly to begin operation. Cyclone IV devices offer the Fast-On feature to support fast wake-up time applications. The MSEL pin settings determine the POR time (t_{POR}) of the device.

For more information about the MSEL pin settings, refer to the *Configuration and Remote System Upgrades in Cyclone IV Devices* chapter.

For more information about the POR specifications, refer to the *Cyclone IV Device Datasheet* chapter.

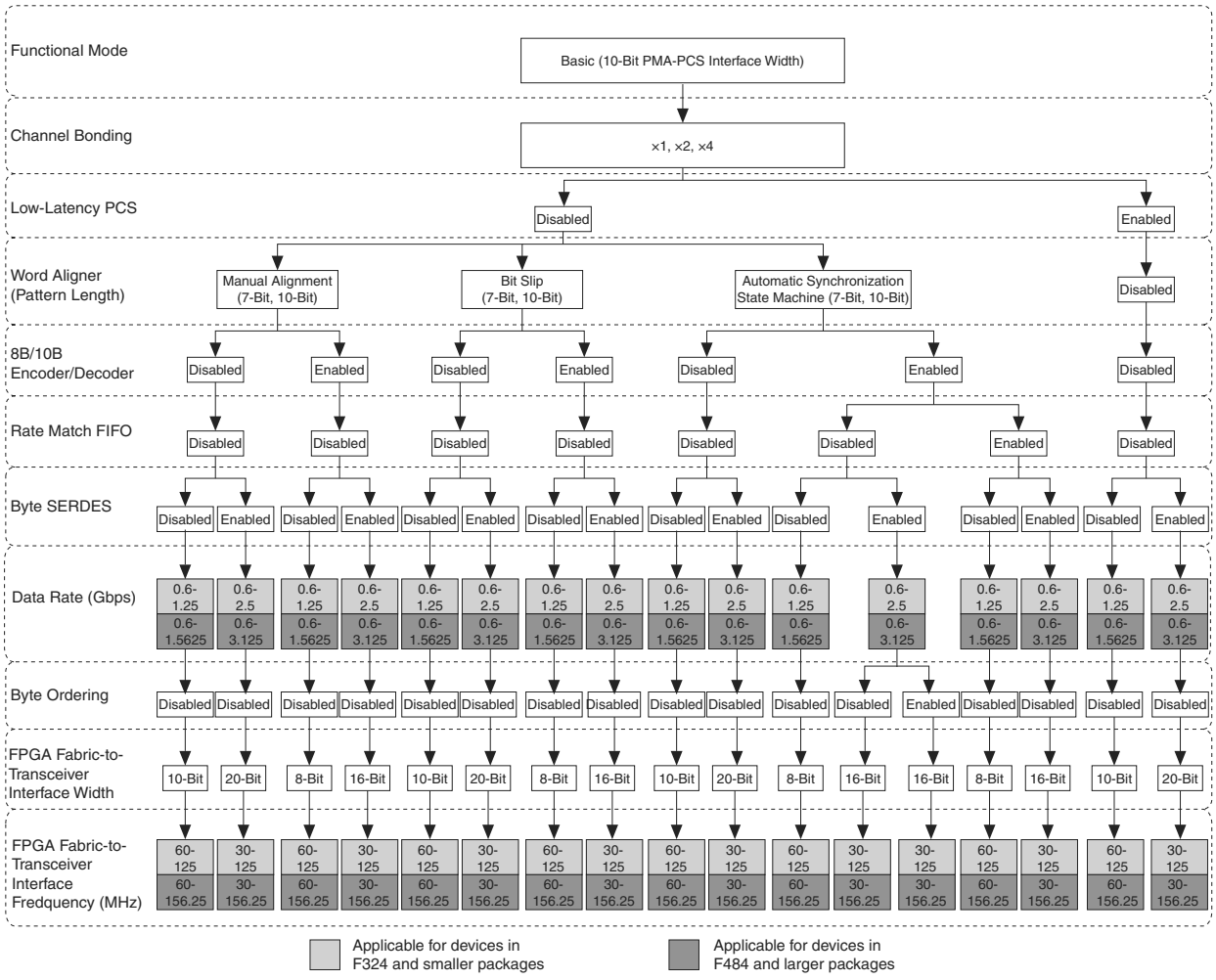
Document Revision History

Table 11–3 lists the revision history for this chapter.

Table 11–3. Document Revision History

Date	Version	Changes
May 2013	1.3	Updated Note (4) in Table 11–1.
July 2010	1.2	■ Updated for the Quartus II software version 10.0 release. ■ Updated "I/O Pins Remain Tri-stated During Power-Up" section. ■ Updated Table 11–1.
February 2010	1.1	Updated Table 11–1 and Table 11–2 for the Quartus II software version 9.1 SP1 release.
November 2009	1.0	Initial release.

Figure 1–46. Transceiver Configurations in Basic Mode with a 10-Bit Wide PMA-to-PCS Interface



Rate Match FIFO Operation in Basic Mode

In Basic mode, the rate match FIFO performs the following operations:

- Deletes a maximum of four skip patterns from a cluster, if there is one skip pattern left in the cluster after deletion
- Insert a maximum of four skip patterns in a cluster, if there are less than five skip patterns in the cluster after deletion
- Automatically deletes the data byte that causes the FIFO to go full and asserts the `rx_rmfifofull` flag synchronous to the subsequent data byte
- Automatically inserts /K30.7/ (9'h1FE) after the data byte that causes the FIFO to go empty and asserts the `rx-fifoempty` flag synchronous to the inserted /K30.7/ (9'h1FE)

Additional Options in Basic Mode

In Basic mode, the transceiver supports the following additional options:

- low-latency PCS operation

- transmitter in electrical idle
- receiver signal detect
- receiver spread spectrum clocking

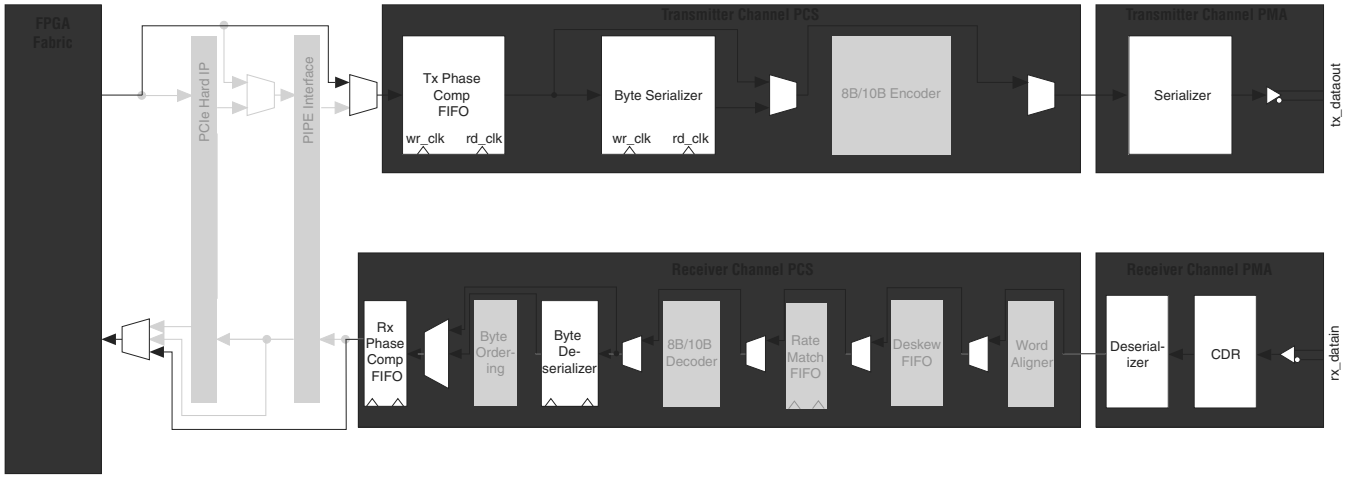
Low-Latency PCS Operation

When configured in low-latency PCS operation, the following blocks in the transceiver PCS are bypassed, resulting in a lower latency PCS datapath:

- 8B/10B encoder and decoder
- word aligner
- rate match FIFO
- byte ordering

Figure 1–47 shows the transceiver channel datapath in Basic mode with low-latency PCS operation.

Figure 1–47. Transceiver Channel Datapath in Basic Mode with Low-Latency PCS Operation



Transmitter in Electrical Idle

The transmitter buffer supports electrical idle state, where when enabled, the differential output buffer driver is tri-stated. During electrical idle, the output buffer assumes the common mode output voltage levels. For details about the electrical idle features, refer to "PCI Express (PIPE) Mode" on page 1–52.

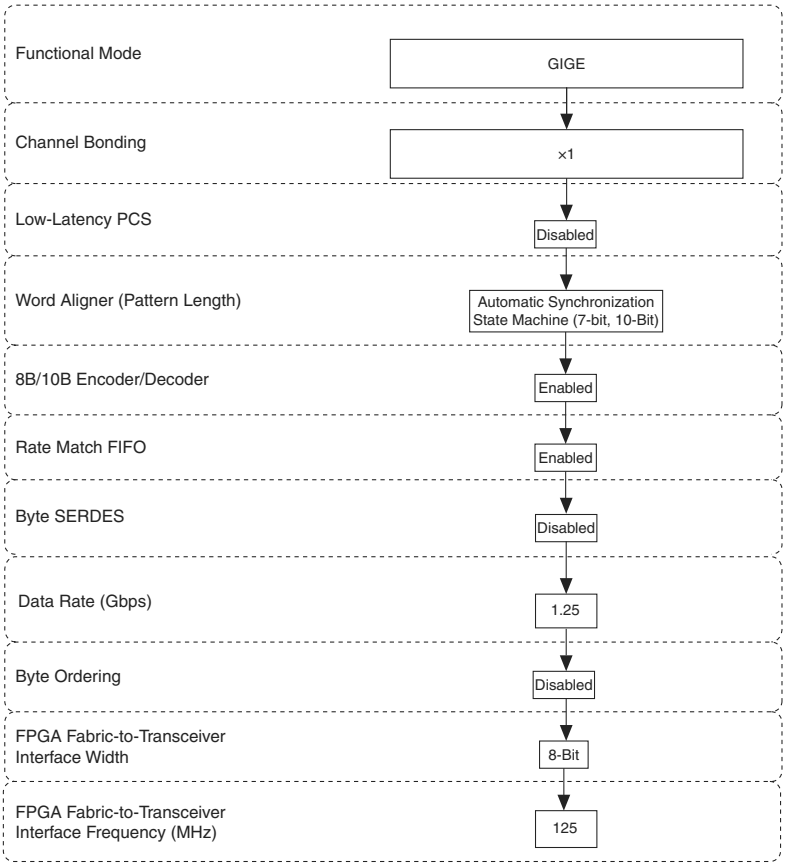
The transmitter in electrical idle feature is required for compliance to the version 2.00 of PHY Interface for the PCI Express (PIPE) Architecture specification for PCIe protocol implementation.

Signal Detect at Receiver

Signal detect at receiver is only supported when 8B/10B encoder/decoder block is enabled.

Figure 1–56 shows the transceiver configuration in GIGE mode.

Figure 1–56. Transceiver Configuration in GIGE Mode



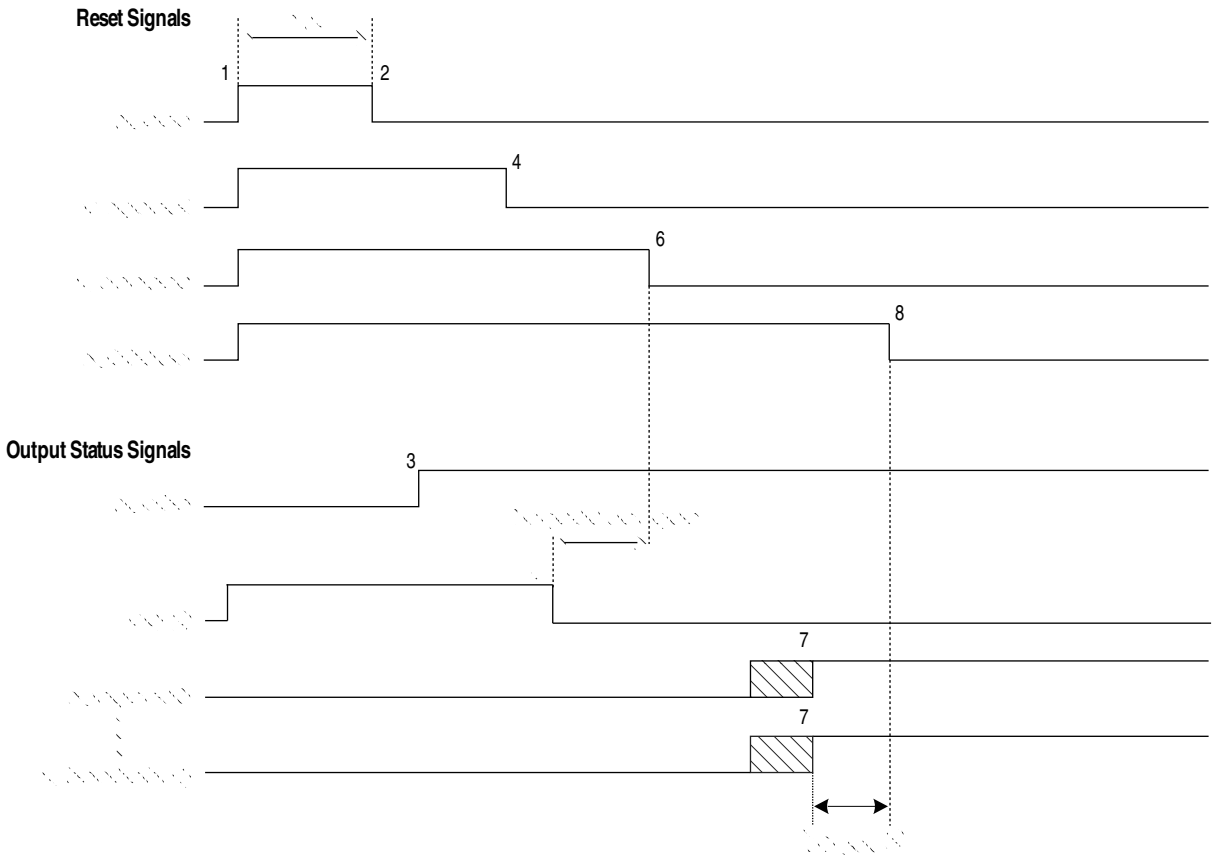
When configured in GIGE mode, three encoded comma (/K28.5/) code groups are transmitted automatically after deassertion of `tx_digitalreset` and before transmitting user data on the `tx_datain` port. This could affect the synchronization state machine behavior at the receiver.

Depending on when you start transmitting the synchronization sequence, there could be an even or odd number of encoded data (/Dx.y/) code groups transmitted between the last of the three automatically sent /K28.5/ code groups and the first /K28.5/ code group of the synchronization sequence. If there is an even number of /Dx.y/ code groups received between these two /K28.5/ code groups, the first /K28.5/ code group of the synchronization sequence begins at an odd code group boundary. An IEEE802.3-compliant GIGE synchronization state machine treats this as an error condition and goes into the Loss-of-Sync state.

Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode

This configuration contains both a transmitter and receiver channel. When the receiver CDR is in automatic lock mode, use the reset sequence shown in Figure 2–4.

Figure 2–4. Sample Reset Sequence for Bonded Configuration Receiver and Transmitter Channels—Receiver CDR in Automatic Lock Mode



Notes to Figure 2–4:

(1) The number of `rx_freqlocked[n]` signals depend on the number of channels configured. `n`=number of channels.

(2) For t_{LTD_Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.

(3) The `busy` signal is asserted and deasserted only during initial power up when offset cancellation occurs. In subsequent reset sequences, the `busy` signal is asserted and deasserted only if there is a read or write operation to the ALTGX_RECONFIG megafunction.

As shown in Figure 2–4, perform the following reset procedure for the receiver CDR in automatic lock mode configuration:

1. After power up, assert `pll_areset` for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset`, `rx_analogreset`, and `rx_digitalreset` signals asserted during this time period. After you deassert the `pll_areset` signal, the multipurpose PLL starts locking to the input reference clock.
3. After the multipurpose PLL locks, as indicated by the `pll_locked` signal going high, deassert the `tx_digitalreset` signal. At this point, the transmitter is ready for data traffic.

Table 3–3. Cyclone IV GX Supported Dynamic Reconfiguration Mode (Part 2 of 2)

Dynamic Reconfiguration Supported Mode	Operational Mode			Quartus II Instances			.mif Requirements
	Transmitter Only	Receiver Only	Transmitter and Receiver Only	ALTGX	ALTGX_ RECONFIG	ALTPLL_ RECONFIG	
Channel Reconfiguration							
Channel Interface	✓	✓	✓	✓	✓	—	✓
Data Rate Division in Receiver Channel	—	✓	✓	✓	✓	—	✓
PLL Reconfiguration	✓	✓	✓	✓	—	✓	✓

The following modes are available for dynamically reconfiguring the Cyclone IV transceivers:

- "PMA Controls Reconfiguration Mode" on page 3–13
- "Transceiver Channel Reconfiguration Mode" on page 3–21
  - Channel interface (**.mif** based)
  - Data rate division in receiver channel (**.mif** based)

The following sections describe each of these modes in detail.

The following modes are unsupported for dynamic reconfiguration:

- Dynamically enable/disable PRBS or BIST
- Switch between a receiver-only channel and a transmitter-only channel
- Switch between a ×1 mode to a bonded ×4 mode

PMA Controls Reconfiguration Mode

You can dynamically reconfigure the following PMA controls for all supported transceiver configurations channels as configured in the ALTGX instances:

- Pre-emphasis settings
- Equalization settings (channel reconfiguration mode does not support equalization settings)
- DC gain settings
- V_{OD} settings

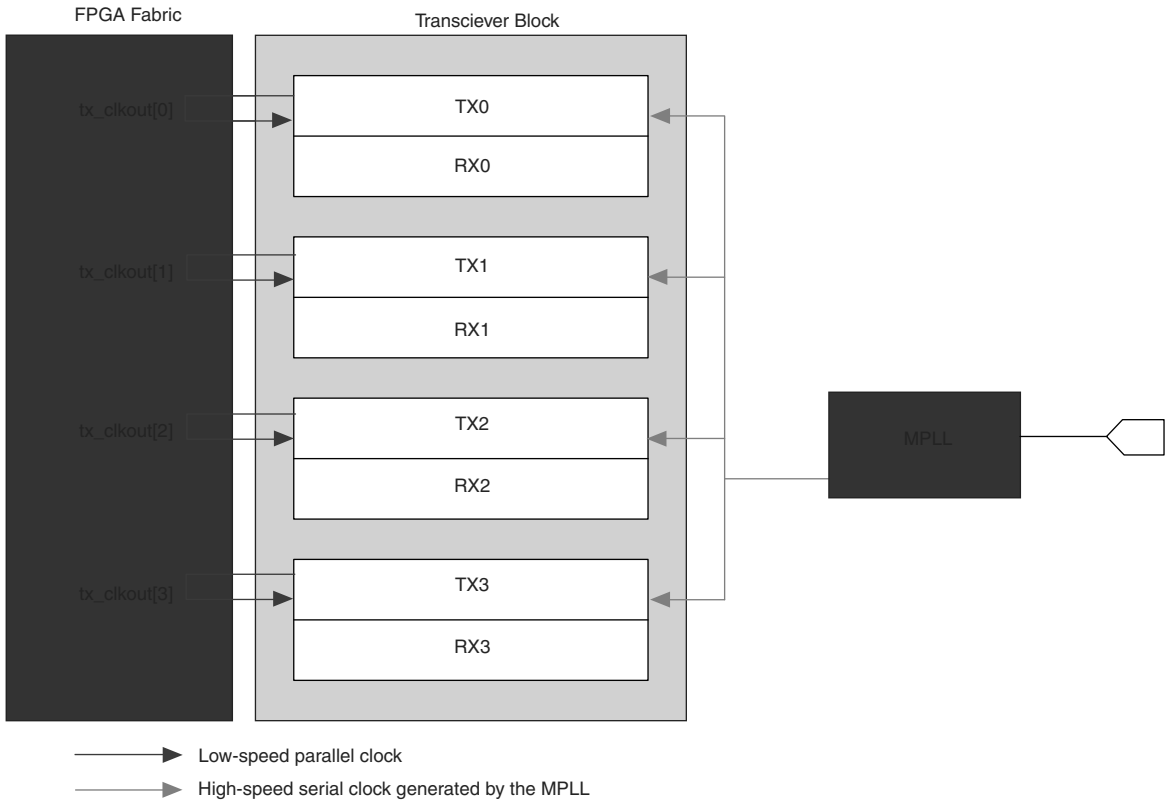
You can use the analog reconfiguration feature to dynamically reconfigure the transceivers channels setting in either the transmitter or the receivers in the PMA blocks. You can update the PMA controls on-the-fly based on the desired input. You can perform both read and write transaction separately for this analog reconfiguration mode.

Option 2: Use the Respective Channel Transmitter Core Clocks

- Enable this option if you want the individual transmitter channel `tx_clkout` signals to provide the write clock to their respective Transmit Phase Compensation FIFOs.
- This option is typically enabled when each transceiver channel is reconfigured to a different functional mode using channel reconfiguration.

Figure 3–12 shows how each transmitter channel's `tx_clkout` signal provides a clock to the Transmit Phase Compensation FIFOs of the respective transceiver channels.

Figure 3–12. Option 2 for Transmitter Core Clocking (Channel Reconfiguration Mode)



Receiver core clocking refers to the clock that is used to read the parallel data from the Receiver Phase Compensation FIFO into the FPGA fabric. You can use one of the following clocks to read from the Receive Phase Compensation FIFO:

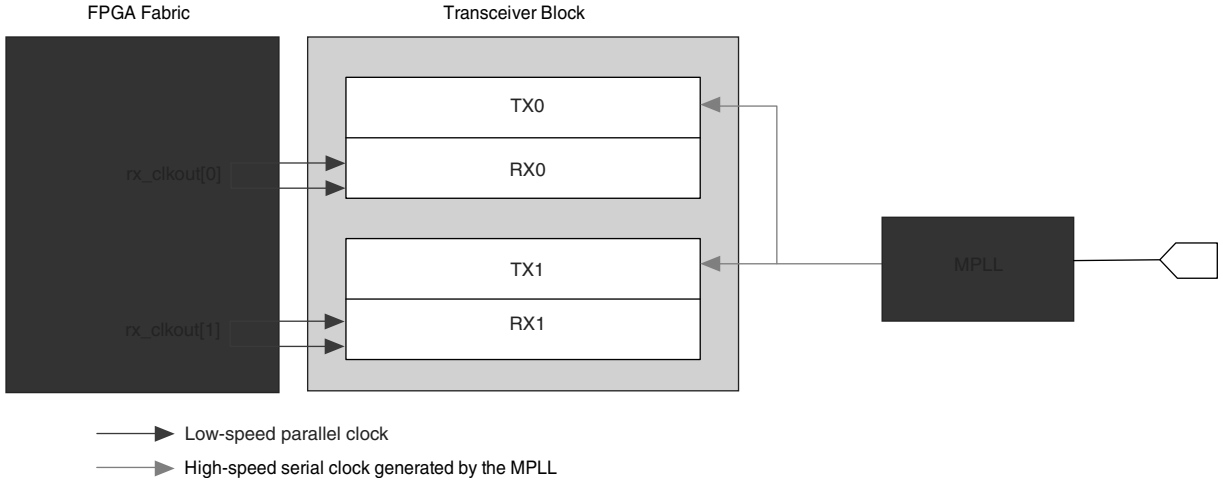
- `rx_coreclk`—you can use a clock of the same frequency as `rx_clkout` from the FPGA fabric to provide the read clock to the Receive Phase Compensation FIFO. If you use `rx_coreclk`, it overrides the `rx_clkout` options in the ALTGX MegaWizard Plug-In Manager.
- `rx_clkout`—the Quartus II software automatically routes `rx_clkout` to the FPGA fabric and back into the Receive Phase Compensation FIFO.

Option 3: Use the Respective Channel Receiver Core Clocks

- Enable this option if you want the individual channel's `rx_clkout` signal to provide the read clock to its respective Receive Phase Compensation FIFO.
- This option is typically enabled when the channel is reconfigured from a Basic or Protocol configuration with or without rate matching to another Basic or Protocol configuration with or without rate matching.

Figure 3–15 shows the respective `rx_clkout` of each channel clocking the respective receiver channels of a transceiver block.

Figure 3–15. Option 3 for Receiver Core Clocking (Channel Reconfiguration Mode)



PLL Reconfiguration Mode

Cyclone IV GX device support the PLL reconfiguration support through the ALTPLL_RECONFIG MegaWizard. You can use this mode to reconfigure the multipurpose PLL or general purpose PLL used to clock the transceiver channel without affecting the remaining blocks of the channel. When you reconfigure the multipurpose PLL or general purpose PLL of a transceiver block to run at a different data rate, all the transceiver channels listening to this multipurpose PLL or general purpose PLL also get reconfigured to the new data rate. Channel settings are not affected. When you reconfigure the multipurpose PLL or general purpose PLL to support a different data rate, you must ensure that the functional mode of the transceiver channel supports the reconfigured data rate.

The PLL reconfiguration mode can be enabled by selecting the **Enable PLL Reconfiguration** option in the ALTGX MegaWizard under **Reconfiguration Setting** tab. For multipurpose PLL or general purpose PLL reconfiguration, **.mif** files are required to dynamically reconfigure the PLL setting in order to change the output frequency of the transceiver PLL to support different data rates.

Transceiver Performance Specifications

Table 1–21 lists the Cyclone IV GX transceiver specifications.

Table 1–21. Transceiver Specification for Cyclone IV GX Devices (Part 1 of 4)

Symbol/ Description	Conditions	C6			C7, I7			C8			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Reference Clock											
Supported I/O Standards	1.2 V PCML, 1.5 V PCML, 3.3 V PCML, Differential LVPECL, LVDS, HCSL										
Input frequency from REFCLK input pins	—	50	—	156.25	50	—	156.25	50	—	156.25	MHz
Spread-spectrum modulating clock frequency	Physical interface for PCI Express (PIPE) mode	30	—	33	30	—	33	30	—	33	kHz
Spread-spectrum downspread	PIPE mode	—	0 to –0.5%	—	—	0 to –0.5%	—	—	0 to –0.5%	—	—
Peak-to-peak differential input voltage	—	0.1	—	1.6	0.1	—	1.6	0.1	—	1.6	V
V_{ICM} (AC coupled)	—	1100 ± 5%			1100 ± 5%			1100 ± 5%			mV
V_{ICM} (DC coupled)	HCSL I/O standard for PCIe reference clock	250	—	550	250	—	550	250	—	550	mV
Transmitter REFCLK Phase Noise *(1)*	Frequency offset = 1 MHz – 8 MHZ	—	—	–123	—	—	–123	—	—	–123	dBc/Hz
Transmitter REFCLK Total Jitter *(1)*		—	—	42.3	—	—	42.3	—	—	42.3	ps
R_{ref}	—	—	2000 ± 1%	—	—	2000 ± 1%	—	—	2000 ± 1%	—	Ω
Transceiver Clock											
cal_blk_clk clock frequency	—	10	—	125	10	—	125	10	—	125	MHz
fixedclk clock frequency	PCIe Receiver Detect	—	125	—	—	125	—	—	125	—	MHz
reconfig_clk clock frequency	Dynamic reconfiguration clock frequency	2.5/ 37.5 *(2)*	—	50	2.5/ 37.5 *(2)*	—	50	2.5/ 37.5 *(2)*	—	50	MHz
Delta time between reconfig_clk	—	—	—	2	—	—	2	—	—	2	ms
Transceiver block minimum power-down pulse width	—	—	1	—	—	1	—	—	1	—	µs

Table 1–47. Document Revision History

Date	Version	Changes
February 2010	1.1	■ Updated Table 1–3 through Table 1–44 to include information for Cyclone IV E devices and Cyclone IV GX devices for Quartus II software version 9.1 SP1 release. ■ Minor text edits.
November 2009	1.0	Initial release.