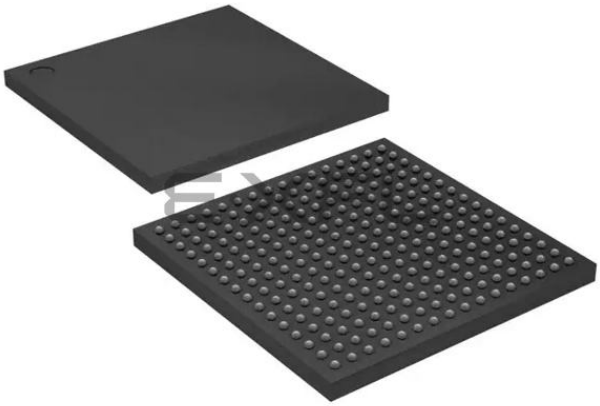


Welcome to E-XFL.COM

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	645
Number of Logic Elements/Cells	10320
Total RAM Bits	423936
Number of I/O	179
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
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/ep4ce10f17i8l

This chapter provides additional information about the document and Altera.

About this Handbook

This handbook provides comprehensive information about the Altera® Cyclone® IV family of devices.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."

High-Speed I/O Interface

Cyclone IV E I/Os are separated into eight I/O banks, as shown in Figure 6–9 on page 6–17. Cyclone IV GX I/Os are separated into six user I/O banks with the left side of the device as the transceiver block, as shown in Figure 6–10 on page 6–18. Each bank has an independent power supply. True output drivers for LVDS, RSDS, mini-LVDS, and PPDS are on the right I/O banks. On the Cyclone IV E row I/O banks and the Cyclone IV GX right I/O banks, some of the differential pin pairs (`p` and `n` pins) of the true output drivers are not located on adjacent pins. In these cases, a power pin is located between the `p` and `n` pins. These I/O standards are also supported on all I/O banks using two single-ended output with the second output programmed as inverted, and an external resistor network. True input buffers for these I/O standards are supported on the top, bottom, and right I/O banks except for I/O bank 9.

Configuration Scheme

A configuration scheme with different configuration voltage standards is selected by driving the MSEL pins either high or low, as shown in Table 8–3, Table 8–4, and Table 8–5.

☞ Hardwire the MSEL pins to V_{CCA} or GND without pull-up or pull-down resistors to avoid problems detecting an incorrect configuration scheme. Do not drive the MSEL pins with a microprocessor or another device.

Table 8–3. Configuration Schemes for Cyclone IV GX Devices (EP4CGX15, EP4CGX22, and EP4CGX30 [except for F484 Package])

Configuration Scheme	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
AS	1	0	1	Fast	3.3
	0	1	1	Fast	3.0, 2.5
	0	0	1	Standard	3.3
	0	1	0	Standard	3.0, 2.5
PS	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	0	Fast	1.8, 1.5
	0	0	0	Standard	3.3, 3.0, 2.5
JTAG-based configuration (2)	(3)	(3)	(3)	—	—

Notes to Table 8–3:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 1 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
AS	1	1	0	1	Fast	3.3
	1	0	1	1	Fast	3.0, 2.5
	1	0	0	1	Standard	3.3
	1	0	1	0	Standard	3.0, 2.5
PS	1	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	1	0	Fast	1.8, 1.5
	1	0	0	0	Standard	3.3, 3.0, 2.5
	0	0	0	0	Standard	1.8, 1.5
FPP	0	0	1	1	Fast	3.3, 3.0, 2.5
	0	1	0	0	Fast	1.8, 1.5
	0	0	0	1	Standard	3.3, 3.0, 2.5
	0	0	1	0	Standard	1.8, 1.5

Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 2 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
JTAG-based configuration (2)	(3)	(3)	(3)	(3)	—	—

Notes to Table 8–4:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

Smaller Cyclone IV E devices or package options (E144 and F256 packages) do not have the MSEL[3] pin. The AS Fast POR configuration scheme at 3.0- or 2.5-V configuration voltage standard and the AP configuration scheme are not supported in Cyclone IV E devices without the MSEL[3] pin. To configure these devices with other supported configuration schemes, select MSEL[2..0] pins according to the MSEL settings in Table 8–5.

Table 8–5. Configuration Schemes for Cyclone IV E Devices

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
AS	1	1	0	1	Fast	3.3
	0	1	0	0	Fast	3.0, 2.5
	0	0	1	0	Standard	3.3
	0	0	1	1	Standard	3.0, 2.5
AP	0	1	0	1	Fast	3.3
	0	1	1	0	Fast	1.8
	0	1	1	1	Standard	3.3
	1	0	1	1	Standard	3.0, 2.5
	1	0	0	0	Standard	1.8
PS	1	1	0	0	Fast	3.3, 3.0, 2.5
	0	0	0	0	Standard	3.3, 3.0, 2.5
FPP	1	1	1	0	Fast	3.3, 3.0, 2.5
	1	1	1	1	Fast	1.8, 1.5
JTAG-based configuration (2)	(3)	(3)	(3)	(3)	—	—

Notes to Table 8–5:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

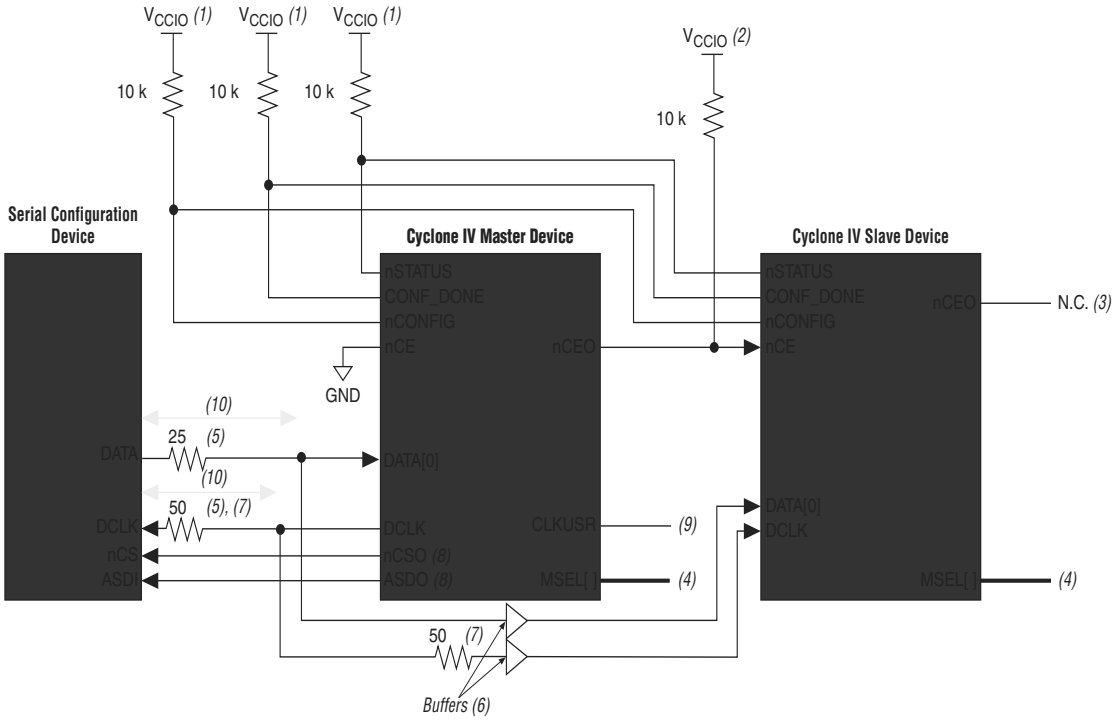
(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

Multi-Device AS Configuration

You can configure multiple Cyclone IV devices with a single serial configuration device. When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. If the last device in the chain is a Cyclone IV device, you can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA[0] pins of each device in the chain are connected together (Figure 8–3).

Figure 8–3. Multi-Device AS Configuration



Notes to Figure 8–3:

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

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of I/O bank in which the nCE pin resides.

(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device of the Cyclone IV device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and slave devices in PS mode, 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.

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

(6) Connect the repeater buffers between the master and slave devices of the Cyclone IV device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

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

(10) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

Programming Parallel Flash Memories

Supported parallel flash memories are external non-volatile configuration devices. They are industry standard microprocessor flash memories. For more information about the supported families for the commodity parallel flash, refer to Table 8–10 on page 8–22.

Cyclone IV E devices in a single- or multiple-device chain support in-system programming of a parallel flash using the JTAG interface with the flash loader megafunction. The board intelligent host or download cable uses the four JTAG pins on Cyclone IV E devices to program the parallel flash in system, even if the host or download cable cannot access the configuration pins of the parallel flash.

For more information about using the JTAG pins on Cyclone IV E devices to program the parallel flash in-system, refer to *AN 478: Using FPGA-Based Parallel Flash Loader (PFL) with the Quartus II Software*.

In the AP configuration scheme, the default configuration boot address is 0×010000 when represented in 16-bit word addressing in the supported parallel flash memory (Figure 8–12). In the Quartus II software, the default configuration boot address is 0×020000 because it is represented in 8-bit byte addressing. Cyclone IV E devices configure from word address 0×010000, which is equivalent to byte address 0×020000.

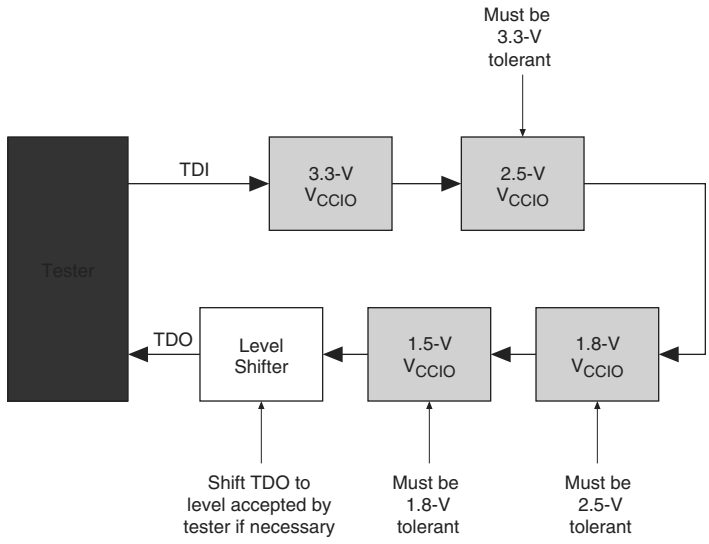
The Quartus II software uses byte addressing for the default configuration boot address. You must set the start address field to **0×020000**.

Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 3 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK (1)	N/A	PS, FPP, AS, AP (2)	Input (PS, FPP) (2)	In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target Cyclone IV device. Data is latched into the device on the rising edge of DCLK. In AS mode, DCLK is an output from the Cyclone IV device that provides timing for the configuration interface. It has an internal pull-up resistor (typically 25 kΩ) that is always active. In AP mode, DCLK is an output from the Cyclone IV E device that provides timing for the configuration interface. (2) In AS or AP configuration schemes, this pin is driven into an inactive state after configuration completes. Alternatively, in active schemes, you can use this pin as a user I/O during user mode. In PS or FPP schemes that use a control host, you must drive DCLK either high or low, whichever is more convenient. In passive schemes, you cannot use DCLK as a user I/O in user mode. Toggling this pin after configuration does not affect the configured device.
	I/O		Output (AS, AP)	
DATA[0] (1)	I/O	PS, FPP, AS, AP (2)	Input (PS, FPP, AS). Bidirectional (AP) (2)	Data input. In serial configuration modes, bit-wide configuration data is presented to the target Cyclone IV device on the DATA[0] pin. In AS mode, DATA[0] has an internal pull-up resistor that is always active. After AS configuration, DATA[0] is a dedicated input pin with optional user control. After PS or FPP configuration, DATA[0] is available as a user I/O pin. The state of this pin depends on the **Dual-Purpose Pin** settings. After AP configuration, DATA[0] is a dedicated bidirectional pin with optional user control. (2)
DATA[1]/ ASDO (1)	I/O	FPP, AS, AP (2)	Input (FPP). Output (AS). Bidirectional (AP) (2)	The DATA[1] pin functions as the ASDO pin in AS mode. Data input in non-AS mode. Control signal from the Cyclone IV device to the serial configuration device in AS mode used to read out configuration data. In AS mode, DATA[1] has an internal pull-up resistor that is always active. After AS configuration, DATA[1] is a dedicated output pin with optional user control. In a PS configuration scheme, DATA[1] functions as a user I/O pin during configuration, which means it is tri-stated. After FPP configuration, DATA[1] is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings. In an AP configuration scheme, for Cyclone IV E devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone IV E device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[1] is a dedicated bidirectional pin with optional user control. (2)

Figure 10–3 shows the JTAG chain of mixed voltages and how a level shifter is inserted in the chain.

Figure 10–3. JTAG Chain of Mixed Voltages



Boundary-Scan Description Language Support

The boundary-scan description language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1/IEEE Std. 1149.6 BST-capable device that can be tested.

For more information about how to download BSDL files for IEEE Std. 1149.1-compliant Cyclone IV E devices, refer to *IEEE Std. 1149.1 BSDL Files*.

For more information about how to download BSDL files for IEEE Std. 1149.6-compliant Cyclone IV GX devices, refer to *IEEE Std. 1149.6 BSDL Files*.

You can also generate BSDL files (pre-configuration and post-configuration) for IEEE Std. 1149.1/IEEE Std. 1149.6-compliant Cyclone IV devices with the Quartus® II software version 9.1 SP1 and later. For more information about the procedure to generate BSDL files using the Quartus II software, refer to *BSDL Files Generation in Quartus II*.

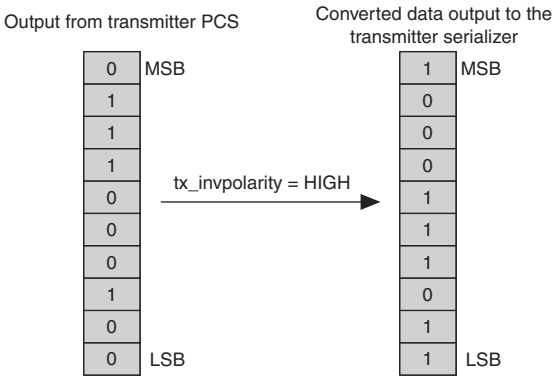
at time n + 2 is encoded as a positive disparity code group. In the same example, the current running disparity at time n + 5 indicates that the K28.5 in time n + 6 should be encoded with a positive disparity. Because tx_forcedisp is high at time n + 6, and tx_dispval is high, the K28.5 at time n + 6 is encoded as a negative disparity code group.

Miscellaneous Transmitter PCS Features

The transmitter PCS supports the following additional features:

- Polarity inversion—corrects accidentally swapped positive and negative signals from the serial differential link during board layout by inverting the polarity of each bit. An optional tx_invpolarity port is available to dynamically invert the polarity of every bit of the 8-bit or 10-bit input data to the serializer in the transmitter datapath. Figure 1–9 shows the transmitter polarity inversion feature.

Figure 1–9. Transmitter Polarity Inversion



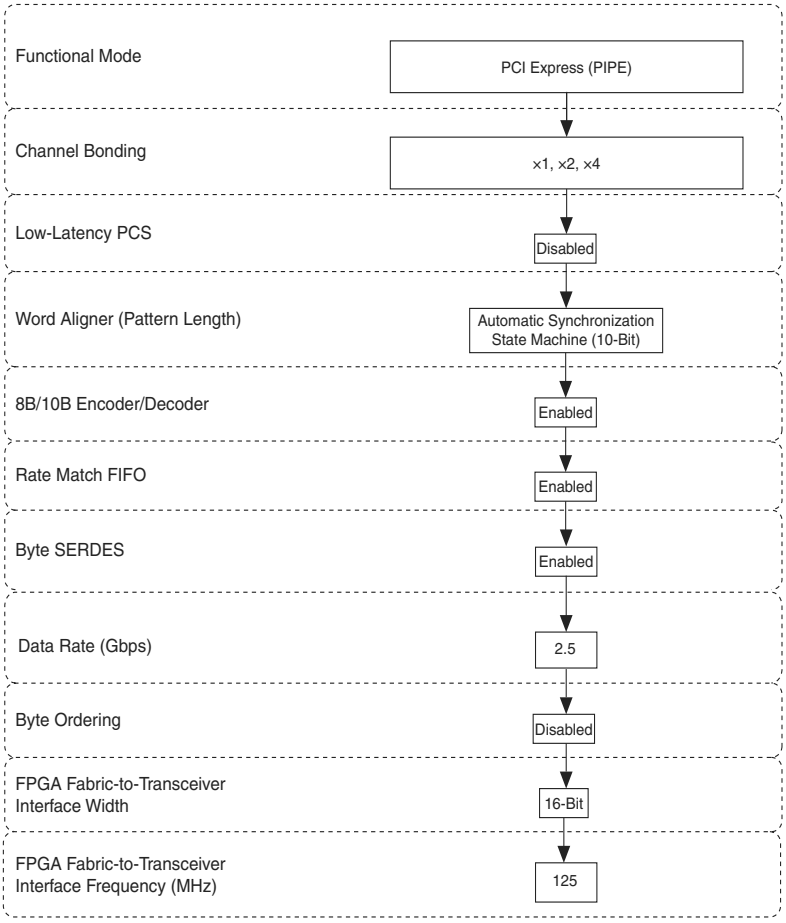
tx_invpolarity is a dynamic signal and might cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

Configuring the hard IP module requires using the PCI Express Compiler. When configuring the transceiver for PCIe implementation with hard IP module, the byte serializer and deserializer are not enabled, providing an 8-bit transceiver-PIPE-hard IP data interface width running at 250 MHz clock frequency.

For more information about PCIe implementation with hard IP module, refer to the *PCI Express Compiler User Guide*.

Figure 1–49 shows the transceiver configuration in PIPE mode.

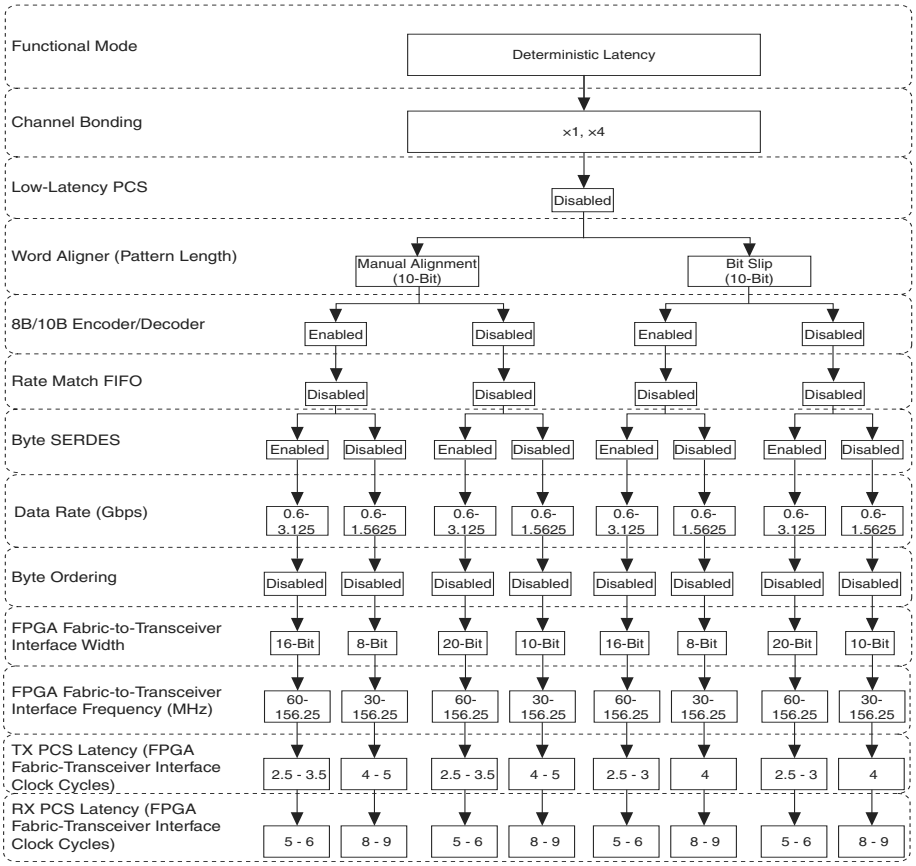
Figure 1–49. Transceiver Configuration in PIPE Mode



When configuring the transceiver into PIPE mode using ALTGX megafunction for PCIe implementation, the PHY-MAC, data link and transaction layers must be implemented in user logics. The PCIe hard IP block is bypassed in this configuration.

Figure 1–67 shows the transceiver configuration in Deterministic Latency mode.

Figure 1–67. Transceiver Configuration in Deterministic Latency Mode



Both CPRI and OBSAI protocols define the serial interface connecting the base station component (specifically channel cards) and remote radio heads (specifically radio frequency cards) in a radio base station system with fiber optics. The protocols require the accuracy of round trip delay measurement for single-hop and multi-hop connections to be within ± 16.276 ns. The Cyclone IV GX transceivers support the following CPRI and OBSAI line rates using Deterministic Latency mode:

- CPRI —614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps, and 3.072 Gbps
- OBSAI—768 Mbps, 1.536 Gbps, and 3.072 Gbps

For more information about deterministic latency implementation, refer to *AN 610: Implementing Deterministic Latency for CPRI and OBSAI Protocols in Stratix IV, HardCopy IV, Arria II GX, and Cyclone IV Devices*.

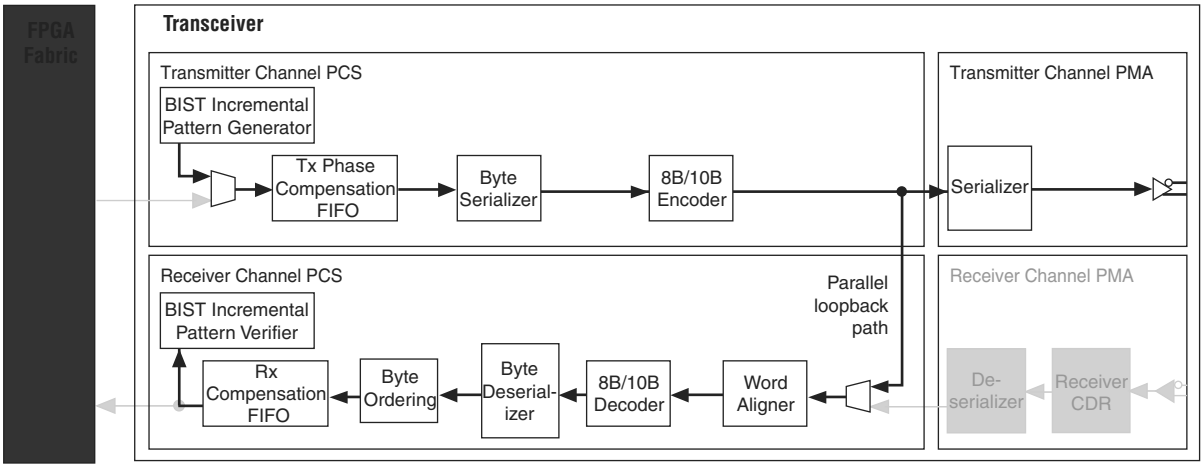
Registered Mode Phase Compensation FIFO

In Deterministic Latency mode, the RX phase compensation FIFO is set to registered mode while the TX phase compensation FIFO supports optional registered mode. When set into registered mode, the phase compensation FIFO acts as a register and eliminates the latency uncertainty through the FIFOs.

BIST

Figure 1–73 shows the datapath for BIST incremental data pattern test mode. The BIST incremental data generator and verifier are located near the FPGA fabric in the PCS block of the transceiver channel.

Figure 1–73. BIST Incremental Pattern Test Mode Datapath



The incremental pattern generator and verifier are 16-bits wide. The generated pattern increments from 00 to FF and passes through the TX PCS blocks, parallel looped back to RX PCS blocks, and checked by the verifier. The pattern is also available as serial data at the `tx_dataout` port. The differential output voltage of the transmitted serial data on the `tx_dataout` port is based on the selected V_{OD} settings. The incremental data pattern is not available to the FPGA logic at the receiver for verification.

The following are the transceiver channel configuration settings in this mode:

- PCS-FPGA fabric channel width: 16-bit
- 8B/10B blocks: Enabled
- Byte serializer/deserializer: Enabled
- Word aligner: Automatic synchronization state machine mode
- Byte ordering: Enabled

The `rx_bisterr` and `rx_bistdone` signals indicate the status of the verifier. The `rx_bisterr` signal is asserted and stays high when detecting an error in the data. The `rx_bistdone` signal is asserted and stays high when the verifier either receives a full cycle of incremental pattern or it detects an error in the receiver data. You can reset the incremental pattern generator and verifier by asserting the `tx_digitalreset` and `rx_digitalreset` ports, respectively.

Table 1–25. PRBS, High and Low Frequency Patterns, and Channel Settings (Part 2 of 2)

Patterns	Polynomial	8-bit Channel Width				10-bit Channel Width			
		Channel Width of 8 bits (1)	Word Alignment Pattern	Maximum Data Rate (Gbps) for F324 and Smaller Packages	Maximum Data Rate (Gbps) for F484 and Larger Packages	Channel Width of 10-bits (1)	Word Alignment Pattern	Maximum Data Rate (Gbps) for F324 and Smaller Packages	Maximum Data Rate (Gbps) for F484 and Larger Packages
Low Frequency (2)	1111100000	N	—	—	—	Y	—	2.5	3.125

Notes to Table 1–25:

(1) Channel width refers to the **What is the channel width?** option in the **General** screen of the ALTGX MegaWizard Plug-In Manager. Based on the selection, an 8 or 10 bits wide pattern is generated as indicated by a **Yes (Y)** or **No (N)**.

(2) A verifier and associated `rx_bistdone` and `rx_bisterr` signals are not available for the specified patterns.

You can enable the serial loopback option to loop the generated PRBS patterns to the receiver channel for verifier to check the PRBS patterns. When the PRBS pattern is received, the `rx_bisterr` and `rx_bistdone` signals indicate the status of the verifier. After the word aligner restores the word boundary, the `rx_bistdone` signal is driven high when the verifier receives a complete pattern cycle and remains asserted until it is reset using the `rx_digitalreset` port. After the assertion of `rx_bistdone`, the `rx_bisterr` signal is asserted for a minimum of three `rx_clkout` cycles when errors are detected in the data and deasserts if the following PRBS sequence contains no error. You can reset the PRBS pattern generator and verifier by asserting the `tx_digitalreset` and `rx_digitalreset` ports, respectively.

Table 2–3. Blocks Affected by Reset and Power-Down Signals (Part 2 of 2)

Transceiver Block	rx_digitalreset	rx_analogreset	tx_digitalreset	pll_areset	gxb_powerdown
Serializer	—	—	✓	—	✓
Transmitter Buffer	—	—	—	—	✓
Transmitter XAUI State Machine	—	—	✓	—	✓
Receiver Buffer	—	—	—	—	✓
Receiver CDR	—	✓	—	—	✓
Receiver Deserializer	—	—	—	—	✓
Receiver Word Aligner	✓	—	—	—	✓
Receiver Deskew FIFO	✓	—	—	—	✓
Receiver Clock Rate Compensation FIFO	✓	—	—	—	✓
Receiver 8B/10B Decoder	✓	—	—	—	✓
Receiver Byte Deserializer	✓	—	—	—	✓
Receiver Byte Ordering	✓	—	—	—	✓
Receiver Phase Compensation FIFO	✓	—	—	—	✓
Receiver XAUI State Machine	✓	—	—	—	✓
BIST Verifiers	✓	—	—	—	✓

Transceiver Reset Sequences

You can configure transceiver channels in Cyclone IV GX devices in various configurations. In all functional modes except XAUI functional mode, transceiver channels can be either bonded or non-bonded. In XAUI functional mode, transceiver channels must be bonded. In PCI Express® (PCIe®) functional mode, transceiver channels can be either bonded or non-bonded and need to follow a specific reset sequence.

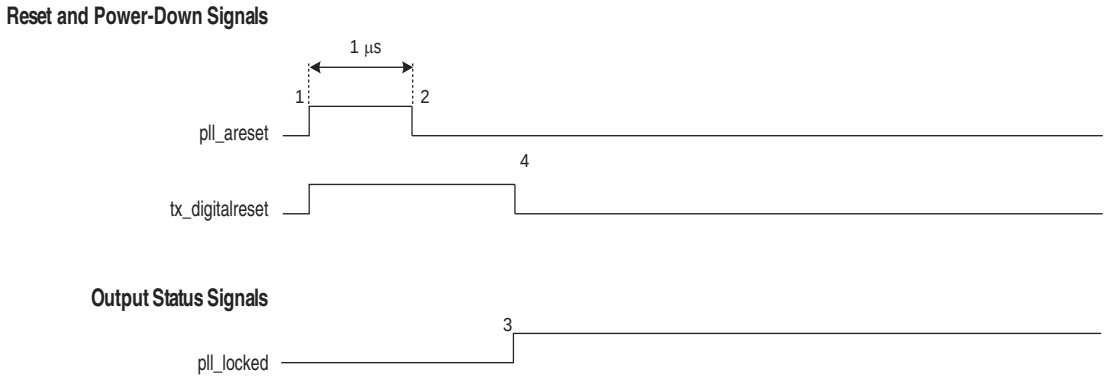
The two categories of reset sequences for Cyclone IV GX devices described in this chapter are:

- "All Supported Functional Modes Except the PCIe Functional Mode" on page 2–6—describes the reset sequences in bonded and non-bonded configurations.
- "PCIe Functional Mode" on page 2–17—describes the reset sequence for the initialization/compliance phase and the normal operation phase in PCIe functional modes.

Transmitter Only Channel

This configuration contains only a transmitter channel. If you create a **Transmitter Only** instance in the ALTGX MegaWizard Plug-In Manager in Basic ×4 functional mode, use the reset sequence shown in Figure 2–3.

Figure 2–3. Sample Reset Sequence for Bonded and Non-Bonded Configuration Transmitter Only Channels

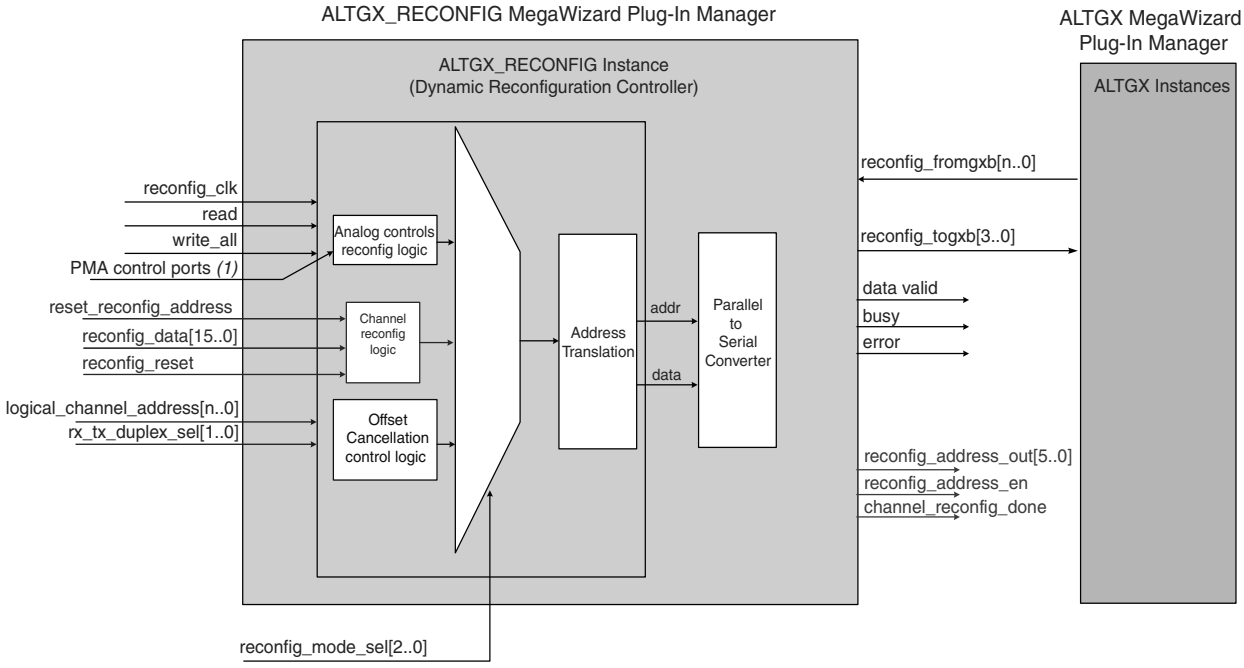


As shown in Figure 2–3, perform the following reset procedure for the **Transmitter Only** channel configuration:

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` signal asserted during this time period. After you de-assert the `pll_areset` signal, the multipurpose PLL starts locking to the transmitter input reference clock.
3. When the multipurpose PLL locks, as indicated by the `pll_locked` signal going high (marker 3), de-assert the `tx_digitalreset` signal (marker 4). At this point, the transmitter is ready for transmitting data.

Figure 3–1 shows a conceptual view of the dynamic reconfiguration controller architecture. For a detailed description of the inputs and outputs of the ALTGX_RECONFIG instance, refer to "Error Indication During Dynamic Reconfiguration" on page 3–36.

Figure 3–1. Dynamic Reconfiguration Controller



Note to Figure 3–1:

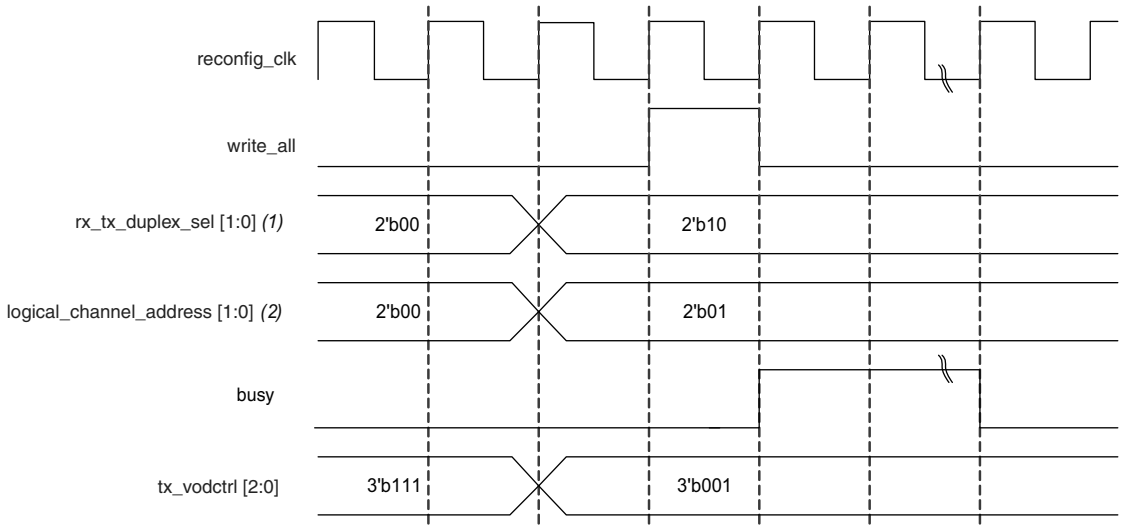
(1) The PMA control ports consist of the V_{OD}, pre-emphasis, DC gain, and manual equalization controls.

Only PMA reconfiguration mode supports manual equalization controls.

You can use one ALTGX_RECONFIG instance to control multiple transceiver blocks. However, you cannot use multiple ALTGX_RECONFIG instances to control one transceiver block.

Figure 3–4 shows the write transaction waveform for Method 1.

Figure 3–4. Write Transaction Waveform—Use 'logical_channel_address port' Option



Notes to Figure 3–4:

(1) In this waveform example, you are writing to only the transmitter portion of the channel.

(2) In this waveform example, the number of channels connected to the dynamic reconfiguration controller is four. Therefore, the `logical_channel_address` port is 2 bits wide.

Read Transaction

For example, to read the existing V_{OD} values from the transmit V_{OD} control registers of the transmitter portion of a specific channel controlled by the ALTGX_RECONFIG instance, perform the following steps:

1. Set the `logical_channel_address` input port to the logical channel address of the transceiver channel whose PMA controls you want to read (for example, `tx_vodctrl_out`).
2. Set the `rx_tx_duplex_sel` port to **2'b10** so that only the transmit PMA controls are read from the transceiver channel.
3. Ensure that the `busy` signal is low before you start a read transaction.
4. Assert the read signal for one `reconfig_clk` clock cycle. This initiates the read transaction.

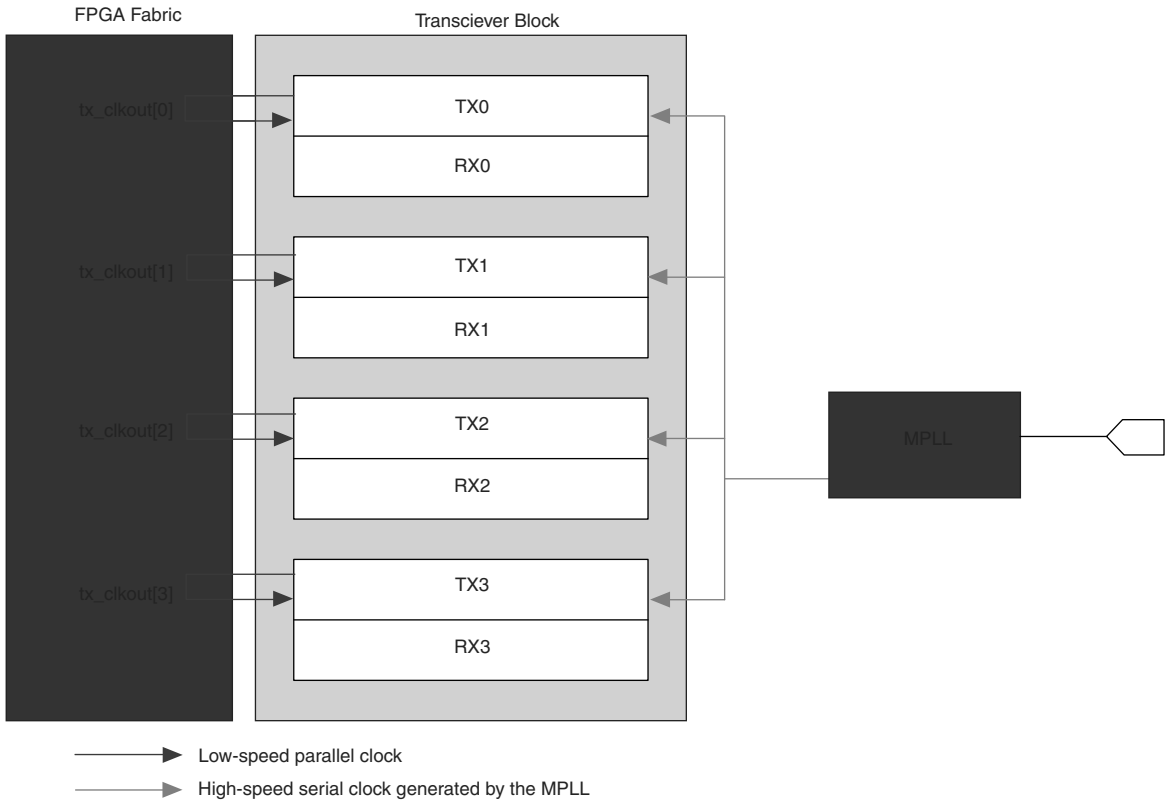
The `busy` output status signal is asserted high to indicate that the dynamic reconfiguration controller is busy reading the PMA control values. When the read transaction has completed, the `busy` signal goes low. The `data_valid` signal is asserted to indicate that the data available at the read control signal is valid.

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.

Example 1–1 shows how to calculate the change of 50-Ω I/O impedance from 25°C at 3.0 V to 85°C at 3.15 V.

Example 1–1. Impedance Change

$\Delta R_V = (3.15 – 3) \times 1000 \times –0.026 = –3.83$

$\Delta R_T = (85 – 25) \times 0.262 = 15.72$

Because ΔR_V is negative,

$MF_V = 1 / (3.83/100 + 1) = 0.963$

Because ΔR_T is positive,

$MF_T = 15.72/100 + 1 = 1.157$

$MF = 0.963 \times 1.157 = 1.114$

$R_{final} = 50 \times 1.114 = 55.71\ \Omega$

Pin Capacitance

Table 1–11 lists the pin capacitance for Cyclone IV devices.

Table 1–11. Pin Capacitance for Cyclone IV Devices *(1)*

Symbol	Parameter	Typical – Quad Flat Pack (QFP)	Typical – Quad Flat No Leads (QFN)	Typical – Ball-Grid Array (BGA)	Unit
C_{IOTB}	Input capacitance on top and bottom I/O pins	7	7	6	pF
C_{IOLR}	Input capacitance on right I/O pins	7	7	5	pF
C_{LVDSLR}	Input capacitance on right I/O pins with dedicated LVDS output	8	8	7	pF
C_{VREFLR} *(2)*	Input capacitance on right dual-purpose VREF pin when used as V_{REF} or user I/O pin	21	21	21	pF
C_{VREFTB} *(2)*	Input capacitance on top and bottom dual-purpose VREF pin when used as V_{REF} or user I/O pin	23 *(3)*	23	23	pF
C_{CLKTB}	Input capacitance on top and bottom dedicated clock input pins	7	7	6	pF
C_{CLKLR}	Input capacitance on right dedicated clock input pins	6	6	5	pF

Notes to Table 1–11:

(1) The pin capacitance applies to FBGA, UBGA, and MBGA packages.

(2) When you use the VREF pin as a regular input or output, you can expect a reduced performance of toggle rate and t_{CO} because of higher pin capacitance.

(3) C_{VREFTB} for the EP4CE22 device is 30 pF.