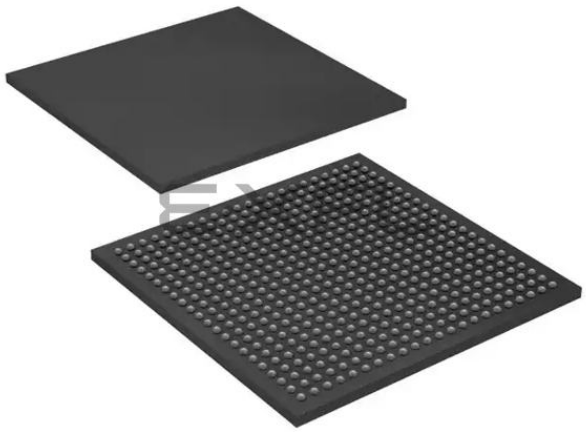


Welcome to E-XFL.COM

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.

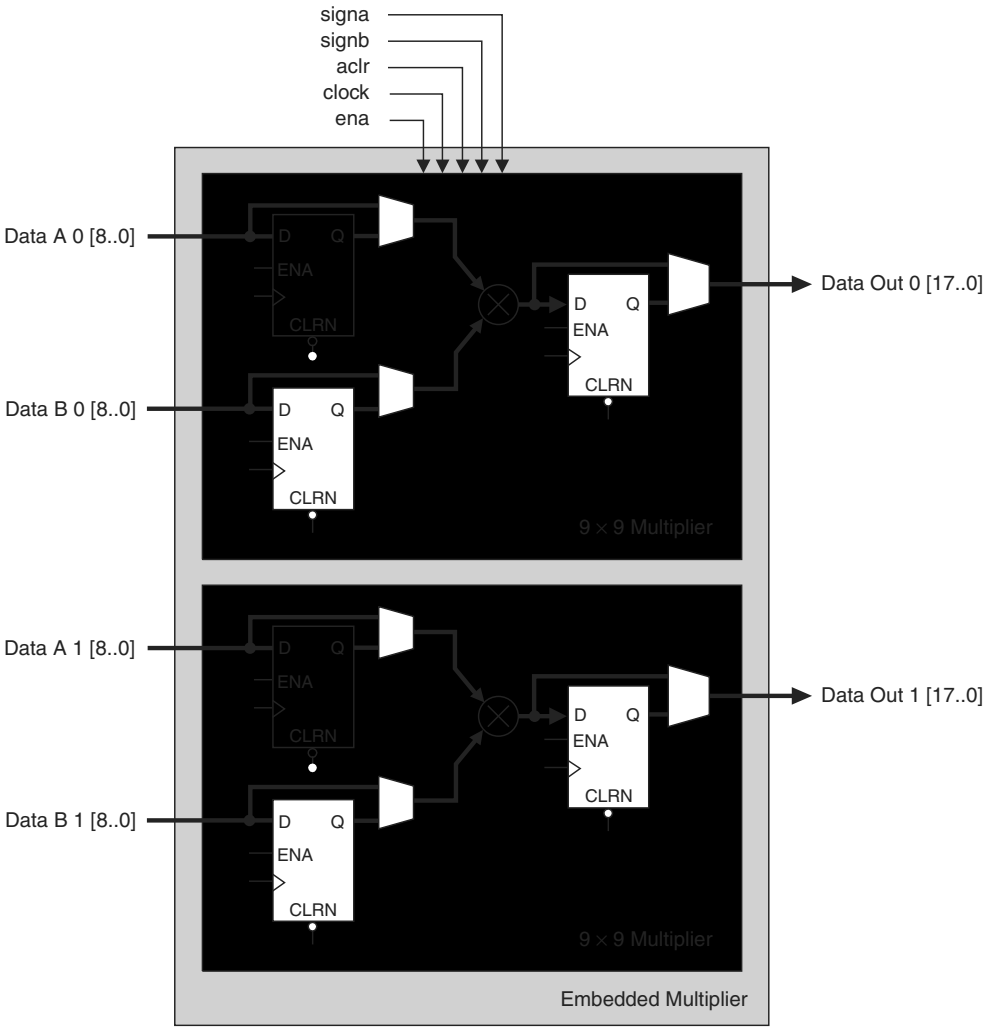
Details	
Product Status	Active
Number of LABs/CLBs	7155
Number of Logic Elements/Cells	114480
Total RAM Bits	3981312
Number of I/O	280
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce115f23c8n

9-Bit Multipliers

You can configure each embedded multiplier to support two 9 × 9 independent multipliers for input widths of up to 9 bits.

Figure 4–4 shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 4–4. 9-Bit Multiplier Mode



All 9-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Two 9 × 9 multipliers in the same embedded multiplier block share the same `signa` and `signb` signal. Therefore, all the `Data A` inputs feeding the same embedded multiplier must have the same sign representation. Similarly, all the `Data B` inputs feeding the same embedded multiplier must have the same sign representation.

- Low time count = 1 cycle
- `rselodd` = 1 effectively equals:
  - High time count = 1.5 cycles
  - Low time count = 1.5 cycles
  - Duty cycle = (1.5/3)% high time count and (1.5/3)% low time count

Scan Chain Description

Cyclone IV PLLs have a 144-bit scan chain.

Table 5–7 lists the number of bits for each component of the PLL.

Table 5–7. Cyclone IV PLL Reprogramming Bits

Block Name	Number of Bits		
	Counter	Other	Total
C4 *(1)*	16	2 *(2)*	18
C3	16	2 *(2)*	18
C2	16	2 *(2)*	18
C1	16	2 *(2)*	18
C0	16	2 *(2)*	18
M	16	2 *(2)*	18
N	16	2 *(2)*	18
Charge Pump	9	0	9
Loop Filter *(3)*	9	0	9
Total number of bits:			144

Notes to Table 5–7:

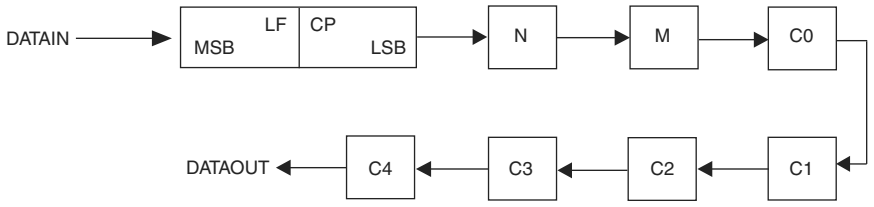
(1) LSB bit for C4 low-count value is the first bit shifted into the scan chain.

(2) These two control bits include `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

(3) MSB bit for loop filter is the last bit shifted into the scan chain.

Figure 5–24 shows the scan chain order of the PLL components.

Figure 5–24. PLL Component Scan Chain Order



For the specific sustaining current for each V_{CCIO} voltage level driven through the resistor and for the overdrive current used to identify the next driven input level, refer to the *Cyclone IV Device Datasheet* chapter.

Programmable Pull-Up Resistor

Each Cyclone IV device I/O pin provides an optional programmable pull-up resistor while in user mode. If you enable this feature for an I/O pin, the pull-up resistor holds the output to the V_{CCIO} level of the output pin's bank.

If you enable the programmable pull-up resistor, the device cannot use the bus-hold feature. Programmable pull-up resistors are not supported on the dedicated configuration, JTAG, and dedicated clock pins.

When the optional `DEV_OE` signal drives low, all I/O pins remains tri-stated even with the programmable pull-up option enabled.

Programmable Delay

The Cyclone IV IOE includes programmable delays to ensure zero hold times, minimize setup times, increase clock-to-output times, and delay the clock input signal.

A path in which a pin directly drives a register may require a programmable delay to ensure zero hold time, whereas a path in which a pin drives a register through combinational logic may not require the delay. Programmable delays minimize setup time. The Quartus II Compiler can program these delays to automatically minimize setup time while providing a zero hold time. Programmable delays can increase the register-to-pin delays for output registers. Each dual-purpose clock input pin provides a programmable delay to the global clock networks.

Table 6–1 shows the programmable delays for Cyclone IV devices.

Table 6–1. Cyclone IV Devices Programmable Delay Chain

Programmable Delay	Quartus II Logic Option
Input pin-to-logic array delay	Input delay from pin to internal cells
Input pin-to-input register delay	Input delay from pin to input register
Output pin delay	Delay from output register to output pin
Dual-purpose clock input pin delay	Input delay from dual-purpose clock pin to fan-out destinations

There are two paths in the IOE for an input to reach the logic array. Each of the two paths can have a different delay. This allows you to adjust delays from the pin to the internal logic element (LE) registers that reside in two different areas of the device. You must set the two combinational input delays with the input delay from pin to internal cells logic option in the Quartus II software for each path. If the pin uses the input register, one of the delays is disregarded and the delay is set with the input delay from pin to input register logic option in the Quartus II software.

The CLKIN/REFCLK pins are powered by dedicated $V_{CC\_CLKIN3A}$, $V_{CC\_CLKIN3B}$, $V_{CC\_CLKIN8A}$, and $V_{CC\_CLKIN8B}$ power supplies separately in their respective I/O banks to avoid the different power level requirements in the same bank for GPIO.

Table 6–10. Cyclone IV GX HSSI REFCLK I/O Standard Support Using GPIO CLKIN Pins *(1), (2)*

I/O Standard	HSSI Protocol	Coupling	Termination	VCC_CLKIN Level		I/O Pin Type		
				Input	Output	Column I/O	Row I/O	Supported I/O Banks
LVDS	All	Differential AC (Need off chip resistor to restore V_{CM})	Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
LVPECL	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
1.2V, 1.5V, 3.3V PCML	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
HCSL	PCIe	Differential DC	Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B

Notes to Table 6–10:

(1) The EP4CGX15, EP4CGX22, and EP4CGX30 devices have two pairs of dedicated clock input pins in banks 3A and 8A for HSSI input reference clock. I/O banks 3B and 8B are not available in EP4CGX15, EP4CGX22, and EP4CGX30 devices.

(2) The EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have four pairs of dedicated clock input pins in banks 3A, 3B, 8A, and 8B for HSSI input or single-ended clock input.

For more information about the AC-coupled termination scheme for the HSSI reference clock, refer to the *Cyclone IV Transceivers Architecture* chapter.

LVDS I/O Standard Support in Cyclone IV Devices

The LVDS I/O standard is a high-speed, low-voltage swing, low power, and GPIO interface standard. Cyclone IV devices meet the ANSI/TIA/EIA-644 standard with the following exceptions:

- The maximum differential output voltage (V_{OD}) is increased to 600 mV. The maximum V_{OD} for ANSI specification is 450 mV.
- The input voltage range is reduced to the range of 1.0 V to 1.6 V, 0.5 V to 1.85 V, or 0 V to 1.8 V based on different frequency ranges. The ANSI/TIA/EIA-644 specification supports an input voltage range of 0 V to 2.4 V.

For LVDS I/O standard electrical specifications in Cyclone IV devices, refer to the *Cyclone IV Device Datasheet* chapter.

devices. The internal oscillator is designed to ensure that its maximum frequency is guaranteed to meet EPCS device specifications. Cyclone IV devices offer the option to select CLKUSR as the external clock source for DCLK. You can change the clock source option in the Quartus II software in the **Configuration** tab of the **Device and Pin Options** dialog box.

EPCS1 does not support Cyclone IV devices because of its insufficient memory capacity.

Table 8–6. AS DCLK Output Frequency

Oscillator	Minimum	Typical	Maximum	Unit
40 MHz	20	30	40	MHz

In configuration mode, the Cyclone IV device enables the serial configuration device by driving the nCSO output pin low, which connects to the nCS pin of the configuration device. The Cyclone IV device uses the DCLK and DATA[1] pins to send operation commands and read address signals to the serial configuration device. The configuration device provides data on its DATA pin, which connects to the DATA[0] input of the Cyclone IV device.

All AS configuration pins (DATA[0], DCLK, nCSO, and DATA[1]) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors.

The timing parameters for AS mode are not listed here because the t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, t_{CF2ST1}, and t_{CD2UM} timing parameters are identical to the timing parameters for PS mode shown in Table 8–12 on page 8–36.

Table 8–8 provides the configuration time for AS configuration.

Table 8–8. AS Configuration Time for Cyclone IV Devices (1)

Symbol	Parameter	Cyclone IV E	Cyclone IV GX	Unit
t_{SU}	Setup time	10	8	ns
t_H	Hold time	0	0	ns
t_{CO}	Clock-to-output time	4	4	ns

Note to Table 8–8:

(1) For the AS configuration timing diagram, refer to the *Serial Configuration (EPCS) Devices Datasheet*.

Enabling compression reduces the amount of configuration data that is sent to the Cyclone IV device, which also reduces configuration time. On average, compression reduces configuration time by 50%.

Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash memory-based devices. You can program these devices in-system with the USB-Blaster™ or ByteBlaster™ II download cables. Alternatively, you can program them with the Altera Programming Unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices through the AS programming interface. During in-system programming, the download cable disables device access to the AS interface by driving the nCE pin high. Cyclone IV devices are also held in reset by a low level on nCONFIG. After programming is complete, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive V_{CC} and GND, respectively.

To perform in-system programming of a serial configuration device through the AS programming interface, you must place the diodes and capacitors as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V (Figure 8–6).

If you want to use the setup shown in Figure 8–6 to perform in-system programming of a serial configuration device and single- or multi-device AS configuration, you do not require a series resistor on the DATA line at the near end of the serial configuration device. The existing diodes and capacitors are sufficient.

Altera has developed the Serial FlashLoader (SFL), a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses its JTAG interface to access the EPCS JIC (JTAG Indirect Configuration Device Programming) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

For more information about implementing the SFL with Cyclone IV devices, refer to *AN 370: Using the Serial FlashLoader with the Quartus II Software*.

During device configuration, Cyclone IV E devices read configuration data using the parallel interface and configure their SRAM cells. This scheme is referred to as the AP configuration scheme because the device controls the configuration interface. This scheme contrasts with the FPP configuration scheme, where an external host controls the interface.

AP Configuration Supported Flash Memories

The AP configuration controller in Cyclone IV E devices is designed to interface with two industry-standard flash families—the Micron P30 Parallel NOR flash family and the Micron P33 Parallel NOR flash family. Unlike serial configuration devices, both of the flash families supported in AP configuration scheme are designed to interface with microprocessors. By configuring from an industry standard microprocessor flash which allows access to the flash after entering user mode, the AP configuration scheme allows you to combine configuration data and user data (microprocessor boot code) on the same flash memory.

The Micron P30 flash family and the P33 flash family support a continuous synchronous burst read mode at 40 MHz DCLK frequency for reading data from the flash. Additionally, the Micron P30 and P33 flash families have identical pin-out and adopt similar protocols for data access.

Cyclone IV E devices use a 40-MHz oscillator for the AP configuration scheme. The oscillator is the same oscillator used in the Cyclone IV E AS configuration scheme.

Table 8–10 lists the supported families of the commodity parallel flash for the AP configuration scheme.

Table 8–10. Supported Commodity Flash for AP Configuration Scheme for Cyclone IV E Devices *(1)*

Flash Memory Density	Micron P30 Flash Family *(2)*	Micron P33 Flash Family *(3)*
64 Mbit	✓	✓
128 Mbit	✓	✓
256 Mbit	✓	✓

Notes to Table 8–10:

(1) The AP configuration scheme only supports flash memory speed grades of 40 MHz and above.
(2) 3.3- , 3.0-, 2.5-, and 1.8-V I/O options are supported for the Micron P30 flash family.
(3) 3.3-, 3.0- and 2.5-V I/O options are supported for the Micron P33 flash family.

Configuring Cyclone IV E devices from the Micron P30 and P33 family 512-Mbit flash memory is possible, but you must properly drive the extra address and FLASH_nCE pins as required by these flash memories.

To check for supported speed grades and package options, refer to the respective flash datasheets.

The AP configuration scheme in Cyclone IV E devices supports flash speed grades of 40 MHz and above. However, AP configuration for all these speed grades must be capped at 40 MHz. The advantage of faster speed grades is realized when your design in the Cyclone IV E devices accesses flash memory in user mode.

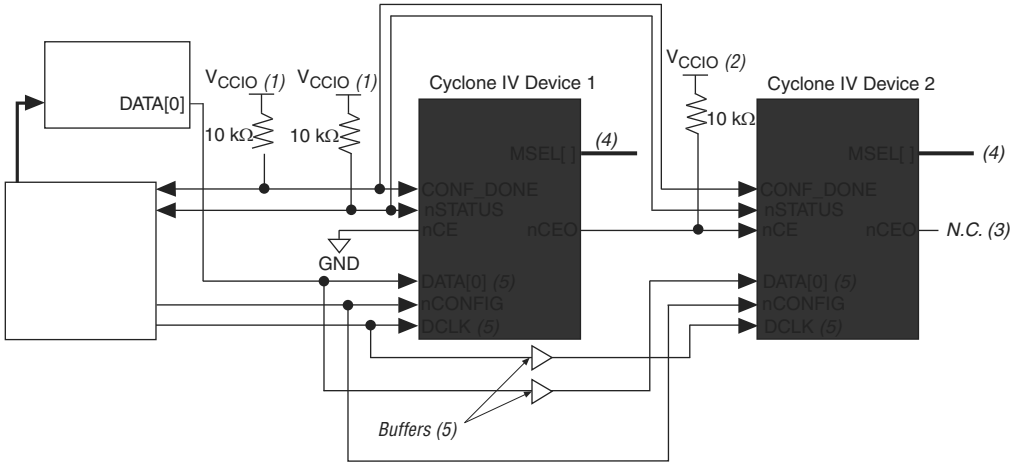
To ensure DCLK and DATA[0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. In the PS scheme, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified system frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

The external host device can also monitor CONF_DONE and INIT_DONE to ensure successful configuration. The CONF_DONE pin must be monitored by the external device to detect errors and to determine when programming is complete. If all configuration data is sent, but CONF_DONE or INIT_DONE has not gone high, the external device must reconfigure the target device.

Figure 8–14 shows how to configure multiple devices using an external host device. This circuit is similar to the PS configuration circuit for a single device, except that Cyclone IV devices are cascaded for multi-device configuration.

Figure 8–14. Multi-Device PS Configuration Using an External Host



Notes to Figure 8–14:

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

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

(3) 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.

(4) 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. Connect the MSEL pins directly to V_{CCA} or GND.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

You can perform JTAG testing on Cyclone IV devices before, during, and after configuration. Cyclone IV devices support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration. All other JTAG instructions can only be issued by first interrupting configuration and reprogramming I/O pins with the ACTIVE_DISENGAGE and CONFIG_IO instructions.

The CONFIG_IO instruction allows you to configure the I/O buffers through the JTAG port and interrupts configuration when issued after the ACTIVE_DISENGAGE instruction. This instruction allows you to perform board-level testing prior to configuring the Cyclone IV device or waiting for a configuration device to complete configuration. Prior to issuing the CONFIG_IO instruction, you must issue the ACTIVE_DISENGAGE instruction. This is because in Cyclone IV devices, the CONFIG_IO instruction does not hold nSTATUS low until reconfiguration, so you must disengage the active configuration mode controller when active configuration is interrupted. The ACTIVE_DISENGAGE instruction places the active configuration mode controllers in an idle state prior to JTAG programming. Additionally, the ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active configuration mode controller.

☞ You must follow a specific flow when executing the ACTIVE_DISENGAGE, CONFIG_IO, and ACTIVE_ENGAGE JTAG instructions in Cyclone IV devices.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins in Cyclone IV devices do not affect JTAG boundary-scan or programming operations. Toggling these pins do not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of Cyclone IV devices, consider the dedicated configuration pins. Table 8–15 describes how you must connect these pins during JTAG configuration.

Table 8–15. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Cyclone IV devices in the chain, nCE must be driven low by connecting it to GND, pulling it low through a resistor, or driving it by some control circuitry. For devices that are also in multi-device AS, AP, PS, or FPP configuration chains, you must connect the nCE pins to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Cyclone IV devices in the chain, nCEO is left floating or connected to the nCE of the next device.
MSEL	These pins must not be left floating. These pins support whichever non-JTAG configuration that you used in production. If you only use JTAG configuration, tie these pins to GND.
nCONFIG	Driven high by connecting to the V_{CCIO} supply of the bank in which the pin resides and pulling up through a resistor or driven high by some control circuitry.
nSTATUS	Pull to the V_{CCIO} supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin must be pulled up to the V_{CCIO} individually.
CONF_DONE	Pull to the V_{CCIO} supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to V_{CCIO} supply of the bank in which the pin resides individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Must not be left floating. Drive low or high, whichever is more convenient on your board.

Chapter Revision Dates

The chapters in this document, Cyclone IV Device Handbook, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Cyclone IV Transceivers Architecture
Revised: *February 2015*
Part Number: *CYIV-52001-3.7*

Chapter 2. Cyclone IV Reset Control and Power Down
Revised: *September 2014*
Part Number: *CYIV-52002-1.4*

Chapter 3. Cyclone IV Dynamic Reconfiguration
Revised: *November 2011*
Part Number: *CYIV-52003-2.1*

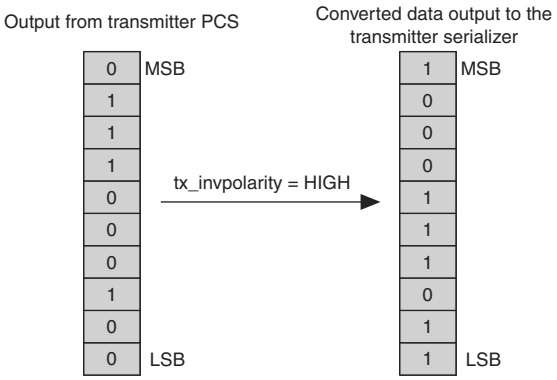
at time n + 2 is encoded as a positive disparity code group. In the same example, the current running disparity at time n + 5 indicates that the K28.5 in time n + 6 should be encoded with a positive disparity. Because tx_forcedisp is high at time n + 6, and tx_dispval is high, the K28.5 at time n + 6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

Byte Deserializer

The byte deserializer halves the FPGA fabric-transceiver interface frequency while doubles the parallel data width to the FPGA fabric.

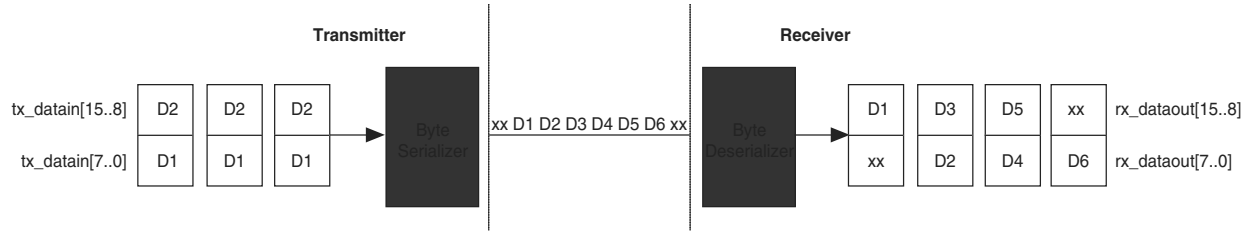
For example, when operating an EP4CGX150 receiver channel at 3.125 Gbps with deserialization factor of 10, the receiver PCS datapath runs at 312.5 MHz. The byte deserializer converts the 10-bit data at 312.5 MHz into 20-bit data at 156.25 MHz before forwarding the data to the FPGA fabric.

Byte Ordering

In the 16- or 20-bit FPGA fabric-transceiver interface, the byte deserializer receives one data byte (8 or 10 bits) and deserializes it into two data bytes (16 or 20 bits). Depending on when the receiver PCS logic comes out of reset, the byte ordering at the output of the byte deserializer may not match the original byte ordering of the transmitted data. The byte misalignment resulting from byte deserialization is unpredictable because it depends on which byte is being received by the byte deserializer when it comes out of reset.

Figure 1–23 shows a scenario where the most significant byte and the least significant byte of the two-byte transmitter data appears straddled across two word boundaries after the data is deserialized at the receiver.

Figure 1–23. Example of Byte Deserializer at the Receiver



The byte ordering block restores the proper byte ordering by performing the following actions:

- Look for the user-programmed byte ordering pattern in the byte-deserialized data
- Inserts a user-programmed pad byte if the user-programmed byte ordering pattern is found in the most significant byte position

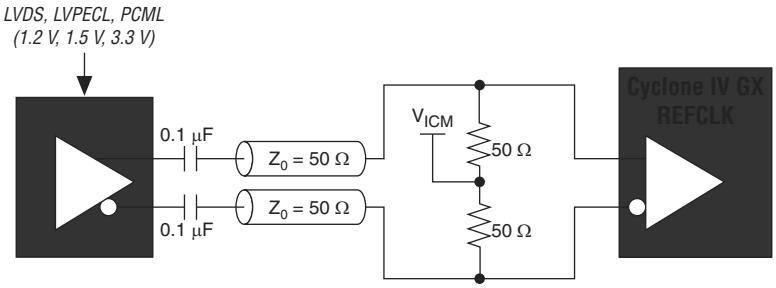
You must select a byte ordering pattern that you know appears at the least significant byte position of the parallel transmitter data.

The byte ordering block is supported in the following receiver configurations:

- 16-bit FPGA fabric-transceiver interface, 8B/10B disabled, and the word aligner in manual alignment mode. Program a custom 8-bit byte ordering pattern and 8-bit pad byte.
- 16-bit FPGA fabric-transceiver interface, 8B/10B enabled, and the word aligner in automatic synchronization state machine mode. Program a custom 9-bit byte ordering pattern and 9-bit pad byte. The MSB of the 9-bit byte ordering pattern and pad byte represents the control identifier of the 8B/10B decoded data.

Figure 1–27 shows an example of the termination scheme for AC-coupled connections for REFCLK pins.

Figure 1–27. AC-Coupled Termination Scheme for a Reference Clock

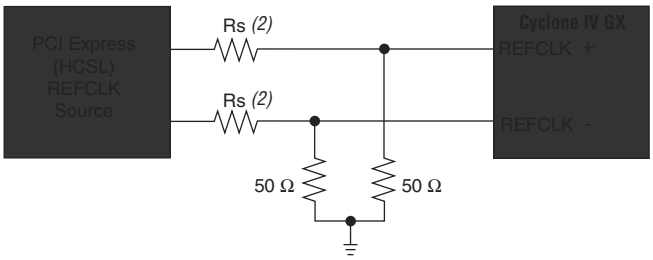


Note to Figure 1–27:

(1) For more information about the V_{ICM} value, refer to the *Cyclone IV Device Datasheet* chapter.

Figure 1–28 shows an example termination scheme for the REFCLK pin when configured as a **HCSL** input.

Figure 1–28. Termination Scheme for a Reference Clock When Configured as HCSL (1)



Notes to Figure 1–28:

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

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

Transceiver Channel Datapath Clocking

Channel datapath clocking varies with channel configuration options and PCS configurations. This section describes the clock distribution from the left PLLs for transceiver channels and the datapath clocking in various supported configurations.

Table 1–7 lists the clocks generated by the PLLs for transceiver datapath.

Table 1–7. PLL Clocks for Transceiver Datapath

Clock	Usage
CDR clocks	Receiver CDR unit
High-speed clock	Transmitter serializer block in PMA
Low-speed clock	Transmitter PCS blocks Receiver PCS blocks when rate match FIFO enabled

Table 1–27. Receiver Ports in ALTGX Megafunction for Cyclone IV GX (Part 1 of 3)

Block	Port Name	Input/ Output	Clock Domain	Description
RX PCS	rx_syncstatus	Output	Synchronous to `tx_clkout` (non-bonded modes with rate match FIFO), `rx_clkout` (non-bonded modes without rate match FIFO), `coreclkout` (bonded modes), or `rx_coreclk` (when using the optional `rx_coreclk` input)	Word alignment synchronization status indicator. This signal passes through the RX Phase Compensation FIFO. ■ Not available in bit-slip mode
	rx_patternde tect	Output	Synchronous to `tx_clkout` (non-bonded modes with rate match FIFO), `rx_clkout` (non-bonded modes without rate match FIFO), `coreclkout` (bonded modes), or `rx_coreclk` (when using the optional `rx_coreclk` input)	Indicates when the word alignment logic detects the alignment pattern in the current word boundary. This signal passes through the RX Phase Compensation FIFO.
	rx_bitslip	Input	Asynchronous signal. Minimum pulse width is two parallel clock cycles.	Bit-slip control for the word aligner configured in bit-slip mode. ■ At every rising edge, word aligner slips one bit into the received data stream, effectively shifting the word boundary by one bit.
	rx_rlv	Output	Asynchronous signal. Driven for a minimum of two recovered clock cycles in configurations without byte serializer and a minimum of three recovered clock cycles in configurations with byte serializer.	Run-length violation indicator. ■ A high pulse indicates that the number of consecutive 1s or 0s in the received data stream exceeds the programmed run length violation threshold.
	rx_invpolarity	Input	Asynchronous signal. Minimum pulse width is two parallel clock cycles.	Generic receiver polarity inversion control. ■ A high level to invert the polarity of every bit of the 8- or 10-bit data to the word aligner.
	rx_enapattern align	Input	Asynchronous signal.	Controls the word aligner operation configured in manual alignment mode.
	rx_rmfifodata inserted	Output	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	Rate match FIFO insertion status indicator. ■ A high level indicates the rate match pattern byte is inserted to compensate for the ppm difference in the reference clock frequencies between the upstream transmitter and the local receiver.
	rx_rmfifodata deleted	Output	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	Rate match FIFO deletion status indicator. ■ A high level indicates the rate match pattern byte is deleted to compensate for the ppm difference in the reference clock frequencies between the upstream transmitter and the local receiver.

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.

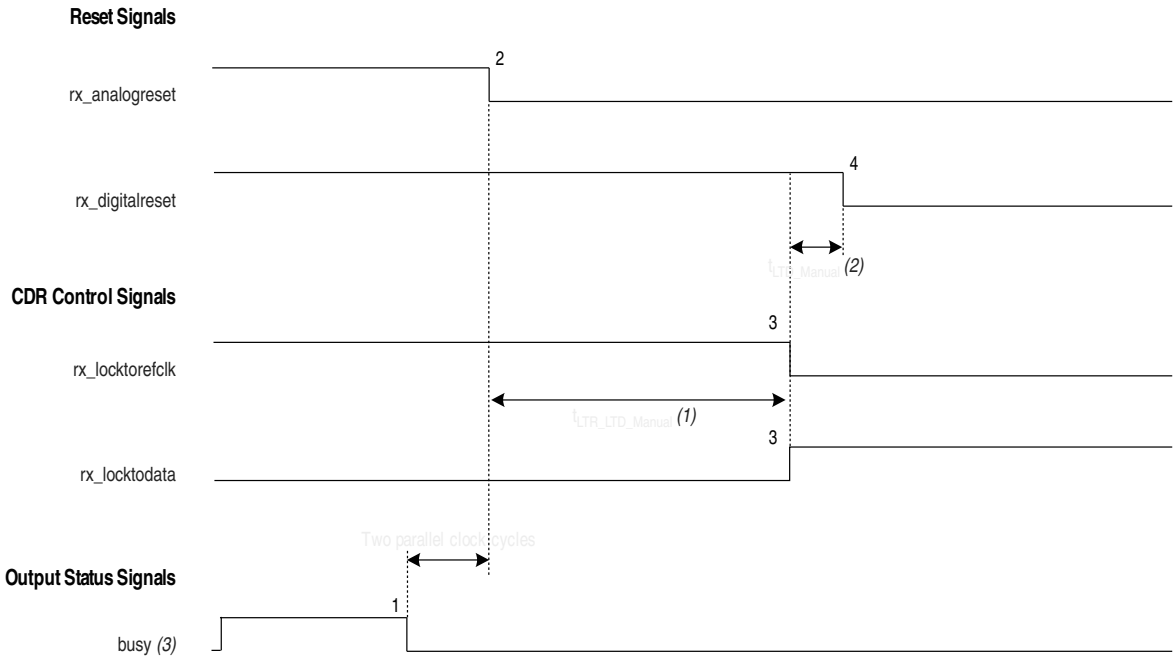
The two categories of reset sequences for Cyclone IV GX devices described in this chapter are:

- "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.

Receiver Only Channel—Receiver CDR in Manual Lock Mode

This configuration contains only a receiver channel. If you create a **Receiver Only** instance in the ALTGX MegaWizard Plug-In Manager with receiver CDR in manual lock mode, use the reset sequence shown in Figure 2–7.

Figure 2–7. Sample Reset Sequence of Receiver Only Channel—Receiver CDR in Manual Lock Mode



Notes to Figure 2–7:

(1) For $t_{LTR\_LTD\_Manual}$ duration, refer to the *Cyclone IV Device Datasheet* chapter.

(2) For t_{LTD_Manual} 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 2–7, perform the following reset procedure for the receiver CDR in manual lock mode:

1. After power up, wait for the busy signal to be asserted.
2. Keep the rx_digitalreset and rx_locktorefclk signals asserted and the rx_locktodata signal deasserted during this time period.
3. After deassertion of the busy signal (marker 1), wait for two parallel clock cycles to deassert the rx_analogreset signal (marker 2). After rx_analogreset deassert, rx_pll_locked will assert.
4. Wait for at least $t_{LTR\_LTD\_Manual}$, then deassert the rx_locktorefclk signal. At the same time, assert the rx_locktodata signal (marker 3).
5. Deassert rx_digitalreset at least t_{LTD_Manual} (the time between markers 3 and 4) after asserting the rx_locktodata signal. At this point, the receiver is ready to receive data.

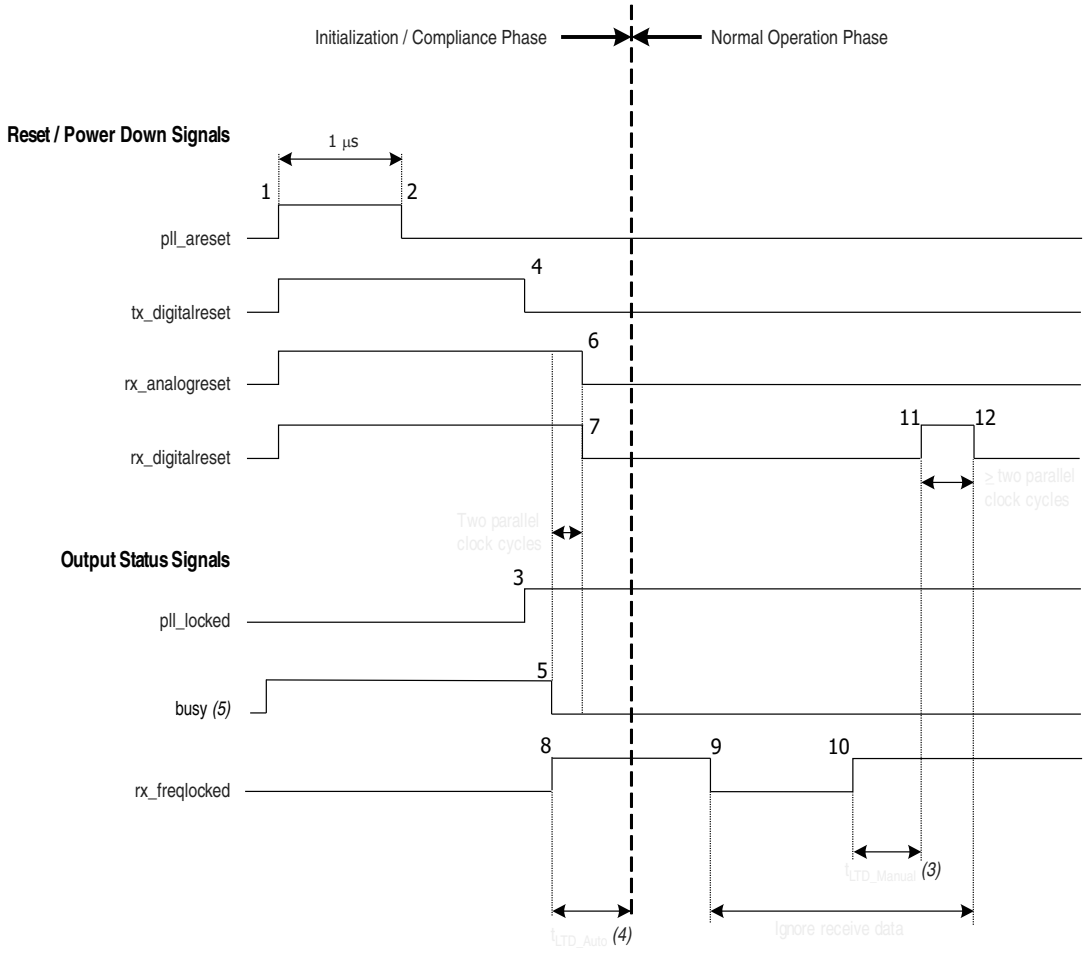
PCIe Functional Mode

You can configure PCIe functional mode with or without the receiver clock rate compensation FIFO in the Cyclone IV GX device. The reset sequence remains the same whether or not you use the receiver clock rate compensation FIFO.

PCIe Reset Sequence

The PCIe protocol consists of an initialization/compliance phase and a normal operation phase. The reset sequences for these two phases are described based on the timing diagram in Figure 2–10.

Figure 2–10. Reset Sequence of PCIe Functional Mode (1), (2)



Notes to Figure 2–10:

(1) This timing diagram is drawn based on the PCIe Gen 1 ×1 mode.

(2) For bonded PCIe Gen 1 ×2 and ×4 modes, there will be additional `rx_freqlocked[n]` signal. n=number of channels.

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

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

(5) 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.

3. Cyclone IV Dynamic Reconfiguration

CYIV-52003-2.1

Cyclone® IV GX transceivers allow you to dynamically reconfigure different portions of the transceivers without powering down any part of the device. This chapter describes and provides examples about the different modes available for dynamic reconfiguration.

You can use the ALTGX_RECONFIG and ALTPLL_RECONFIG controller instance to reconfigure the physical medium attachment (PMA) controls, physical coding sublayer (PCS), multipurpose phase locked loops (PLLs), and general purpose PLLs.

This chapter contains the following sections:

- "Glossary of Terms" on page 3–1
- "Dynamic Reconfiguration Controller Architecture" on page 3–2
- "Dynamic Reconfiguration Modes" on page 3–12
- "Error Indication During Dynamic Reconfiguration" on page 3–36
- "Functional Simulation of the Dynamic Reconfiguration Process" on page 3–37

Glossary of Terms

Table 3–1 lists the terms used in this chapter:

Table 3–1. Glossary of Terms Used in this Chapter (Part 1 of 2)

Term	Description
ALTGX_RECONFIG Instance	Dynamic reconfiguration controller instance generated by the ALTGX_RECONFIG MegaWizard™ Plug-In Manager.
ALTGX Instance	Transceiver instance generated by the ALTGX MegaWizard Plug-In Manager.
ALTPLL_RECONFIG Instance	Dynamic PLL reconfiguration controller instance generated by the ALTPLL_RECONFIG Megawizard Plug-In Manager
Logical Channel Addressing	Used whenever the concept of logical channel addressing is explained. This term does not refer to the `logical_channel_address` port available in the ALTGX_RECONFIG MegaWizard Plug-In Manager.

Table 3–5. rx_dataoutfull[31..0] FPGA Fabric-Transceiver Channel Interface Signal Descriptions (Part 2 of 3)

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
16-bit FPGA fabric-Transceiver Channel Interface with PCS-PMA set to 8/10 bits	Two 8-bit unencoded Data (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[23:16]- rx_dataout (MSByte)
	The following signals are used in 16-bit 8B/10B modes:
	Two Control Bits rx_dataoutfull[8] - rx_ctrldetect (LSB) and rx_dataoutfull[24]- rx_ctrldetect (MSB)
	Two Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) and rx_dataoutfull[25]- rx_errdetect (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull [10] - rx_syncstatus (LSB) and rx_dataoutfull[26] - rx_syncstatus (MSB)
	Two Receiver Disparity Error Bits rx_dataoutfull [11] - rx_disperr (LSB) and rx_dataoutfull[27] - rx_disperr (MSB)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[28]- rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[29]: Rate Match FIFO deletion status indicator (rx_rmfifodatadeleted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[14] and rx_dataoutfull[30]: Rate Match FIFO insertion status indicator (rx_rmfifodatainserted) in non-PCI Express (PIPE) functional modes
	Two 2-bit PCI Express (PIPE) Functional Mode Status Bits rx_dataoutfull[14:13] - rx_pipestatus (LSB) and rx_dataoutfull[30:29] - rx_pipestatus (MSB)
	rx_dataoutfull[15] and rx_dataoutfull[31]: 8B/10B running disparity indicator (rx_runningdisp)

IOE Programmable Delay

Table 1–40 and Table 1–41 list the IOE programmable delay for Cyclone IV E 1.0 V core voltage devices.

Table 1–40. IOE Programmable Delay on Column Pins for Cyclone IV E 1.0 V Core Voltage Devices *(1), (2)*

Parameter	Paths Affected	Number of Setting	Min Offset	Max Offset					Unit
				Fast Corner		Slow Corner			
				C8L	I8L	C8L	C9L	I8L	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	2.054	1.924	3.387	4.017	3.411	ns
Input delay from pin to input register	Pad to I/O input register	8	0	2.010	1.875	3.341	4.252	3.367	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.641	0.631	1.111	1.377	1.124	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.971	0.931	1.684	2.298	1.684	ns

Notes to Table 1–40:

(1) The incremental values for the settings are generally linear. For the exact values for each setting, use the latest version of the Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software.

Table 1–41. IOE Programmable Delay on Row Pins for Cyclone IV E 1.0 V Core Voltage Devices *(1), (2)*

Parameter	Paths Affected	Number of Setting	Min Offset	Max Offset					Unit
				Fast Corner		Slow Corner			
				C8L	I8L	C8L	C9L	I8L	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	2.057	1.921	3.389	4.146	3.412	ns
Input delay from pin to input register	Pad to I/O input register	8	0	2.059	1.919	3.420	4.374	3.441	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.670	0.623	1.160	1.420	1.168	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.960	0.919	1.656	2.258	1.656	ns

Notes to Table 1–41:

(1) The incremental values for the settings are generally linear. For the exact values for each setting, use the latest version of the Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software.