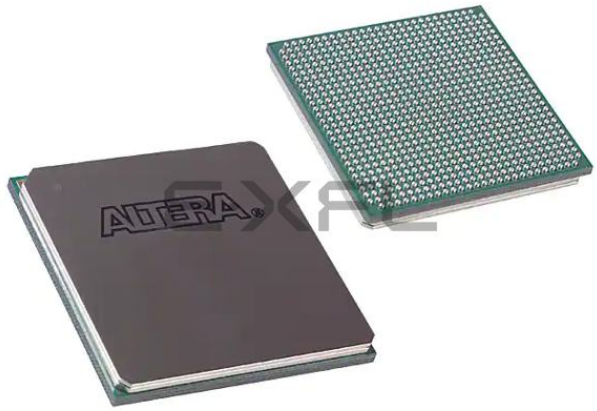


Welcome to E-XFL.COM

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	2475
Number of Logic Elements/Cells	39600
Total RAM Bits	1161216
Number of I/O	532
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	780-BGA
Supplier Device Package	780-FBGA (29x29)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce40f29i8ln

Chapter 9. SEU Mitigation in Cyclone IV Devices

Chapter 10. JTAG Boundary-Scan Testing for Cyclone IV Devices

In addition to the three general routing outputs, LEs in an LAB have register chain outputs, which allows registers in the same LAB to cascade together. The register chain output allows the LUTs to be used for combinational functions and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources.

LE Operating Modes

Cyclone IV LEs operate in the following modes:

- Normal mode
- Arithmetic mode

The Quartus® II software automatically chooses the appropriate mode for common functions, such as counters, adders, subtractors, and arithmetic functions, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions. You can also create special-purpose functions that specify which LE operating mode to use for optimal performance, if required.

Normal Mode

Normal mode is suitable for general logic applications and combinational functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (Figure 2–2). The Quartus II Compiler automatically selects the carry-in (`cin`) or the `data3` signal as one of the inputs to the LUT. LEs in normal mode support packed registers and register feedback.

Figure 2–2 shows LEs in normal mode.

Figure 2–2. Cyclone IV Device LEs in Normal Mode

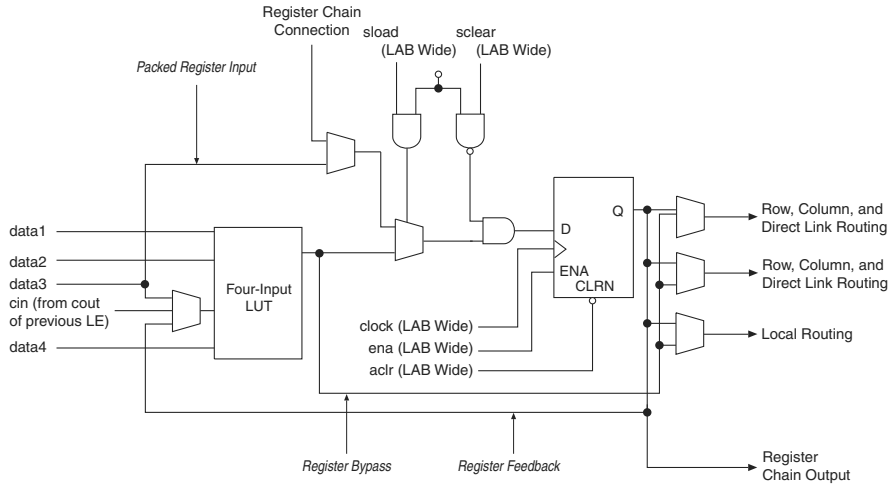
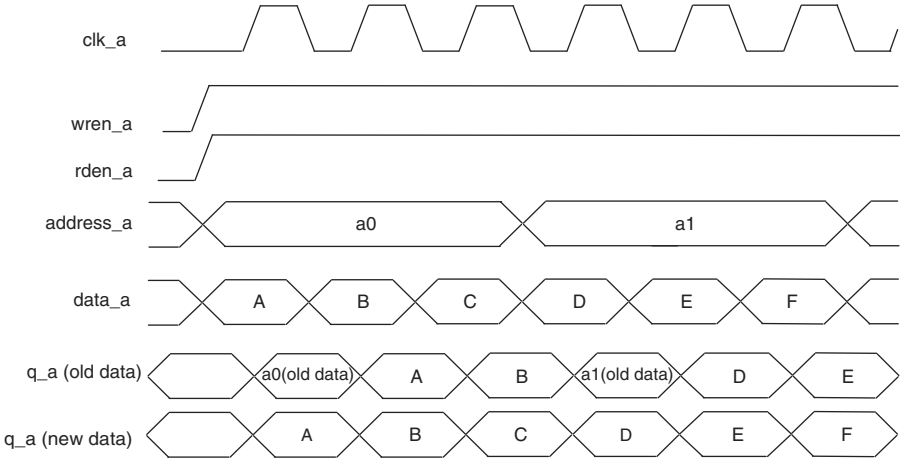


Figure 3–7 shows a timing waveform for read and write operations in single-port mode with unregistered outputs. Registering the outputs of the RAM simply delays the q output by one clock cycle.

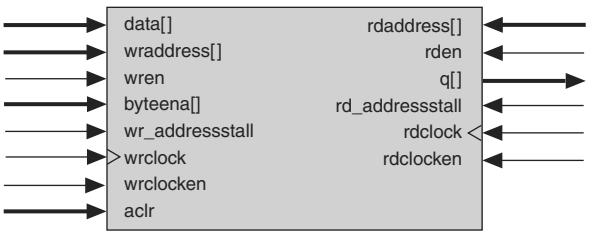
Figure 3–7. Cyclone IV Devices Single-Port Mode Timing Waveform



Simple Dual-Port Mode

Simple dual-port mode supports simultaneous read and write operations to different locations. Figure 3–8 shows the simple dual-port memory configuration.

Figure 3–8. Cyclone IV Devices Simple Dual-Port Memory *(1)*



Note to Figure 3–8:

(1) Simple dual-port RAM supports input or output clock mode in addition to the read or write clock mode shown.

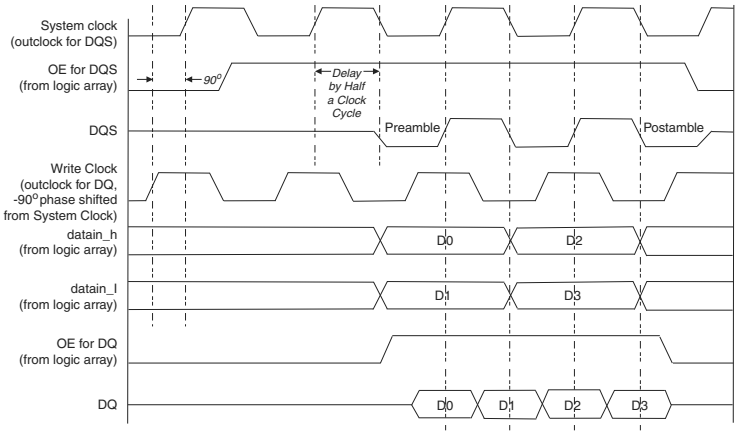
Cyclone IV devices M9K memory blocks support mixed-width configurations, allowing different read and write port widths. Table 3–3 lists mixed-width configurations.

Table 3–3. Cyclone IV Devices M9K Block Mixed-Width Configurations (Simple Dual-Port Mode) ***(Part 1 of 2)***

Read Port	Write Port								
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	256 × 32	1024 × 9	512 × 18	256 × 36
8192 × 1	✓	✓	✓	✓	✓	✓	—	—	—
4096 × 2	✓	✓	✓	✓	✓	✓	—	—	—
2048 × 4	✓	✓	✓	✓	✓	✓	—	—	—
1024 × 8	✓	✓	✓	✓	✓	✓	—	—	—

Figure 7–9 illustrates how the second output enable register extends the DQS high-impedance state by half a clock cycle during a write operation.

Figure 7–9. Extending the OE Disable by Half a Clock Cycle for a Write Transaction *(1)*



Note to Figure 7–9:

(1) The waveform reflects the software simulation result. The OE signal is an active low on the device. However, the Quartus II software implements the signal as an active high and automatically adds an inverter before the A_{OE} register D input.

OCT with Calibration

Cyclone IV devices support calibrated on-chip series termination (R_S OCT) in both vertical and horizontal I/O banks. To use the calibrated OCT, you must use the RUP and RDN pins for each R_S OCT control block (one for each side). You can use each OCT calibration block to calibrate one type of termination with the same V_{CCIO} for that given side.

For more information about the Cyclone IV devices OCT calibration block, refer to the *Cyclone IV Device I/O Features* chapter.

PLL

When interfacing with external memory, the PLL is used to generate the memory system clock, the write clock, the capture clock and the logic-core clock. The system clock generates the DQS write signals, commands, and addresses. The write-clock is shifted by -90° from the system clock and generates the DQ signals during writes. You can use the PLL reconfiguration feature to calibrate the read-capture phase shift to balance the setup and hold margins.

The PLL is instantiated in the ALTMEMPHY megafunction. All outputs of the PLL are used when the ALTMEMPHY megafunction is instantiated to interface with external memories. PLL reconfiguration is used in the ALTMEMPHY megafunction to calibrate and track the read-capture phase to maintain the optimum margin.

For more information about usage of PLL outputs by the ALTMEMPHY megafunction, refer to the *External Memory Interface Handbook*.

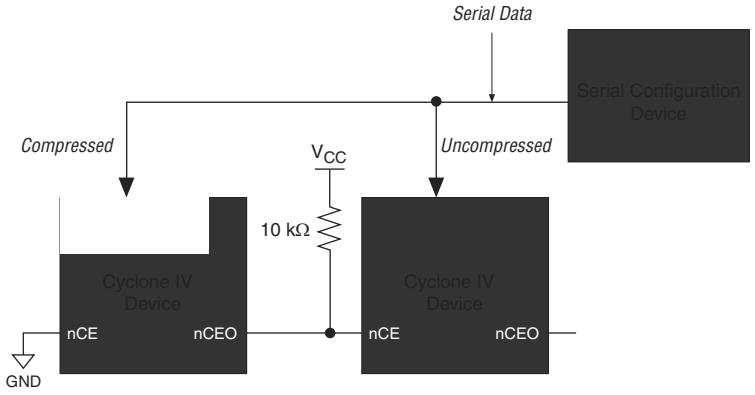
3. Click the **Configuration** tab.
4. Turn on **Generate compressed bitstreams**.
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

You can enable compression when creating programming files from the **Convert Programming Files** dialog box. To enable compression, perform the following steps:

1. On the File menu, click **Convert Programming Files**.
2. Under **Output programming file**, select your desired file type from the **Programming file type** list.
3. If you select **Programmer Object File (.pof)**, you must specify the configuration device in the **Configuration device** list.
4. Under **Input files to convert**, select **SOF Data**.
5. Click **Add File** to browse to the Cyclone IV device SRAM object files (**.sof)**.
6. In the **Convert Programming Files** dialog box, select the **.pof** you added to **SOF Data** and click **Properties**.
7. In the **SOF File Properties** dialog box, turn on the **Compression** option.

When multiple Cyclone IV devices are cascaded, you can selectively enable the compression feature for each device in the chain. Figure 8–1 shows a chain of two Cyclone IV devices. The first device has compression enabled and receives compressed bitstream from the configuration device. The second device has the compression feature disabled and receives uncompressed data. You can generate programming files for this setup in the **Convert Programming Files** dialog box.

Figure 8–1. Compressed and Uncompressed Configuration Data in the Same Configuration File



Configuration Requirement

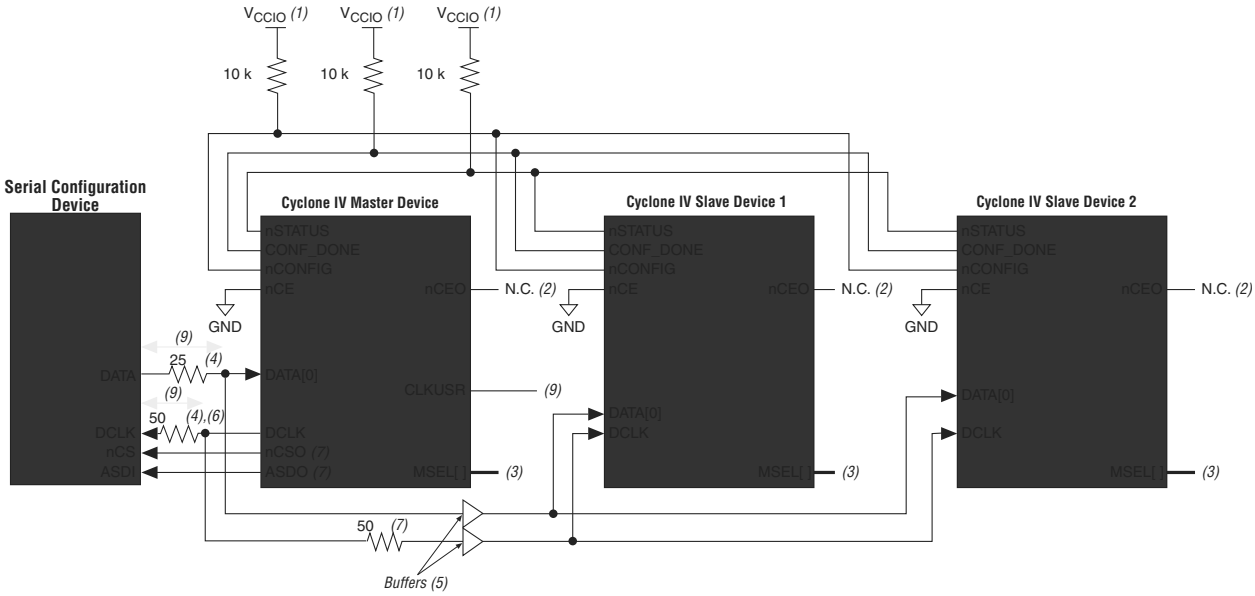
This section describes Cyclone IV device configuration requirement and includes the following topics:

- "Power-On Reset (POR) Circuit" on page 8–4
- "Configuration File Size" on page 8–4
- "Power Up" on page 8–6

Single SRAM Object File

The second method configures both the master device and slave devices with the same **.sof**. The serial configuration device stores one copy of the **.sof**. You must set up one or more slave devices in the chain. All the slave devices must be set up in the same way (Figure 8–5).

Figure 8–5. Multi-Device AS Configuration in Which Devices Receive the Same Data with a Single .sof



Notes to Figure 8–5:

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

(2) 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.

(3) 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.

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

(5) Connect the repeater buffers between the master and slave devices 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.

(6) 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.

(7) 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.

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

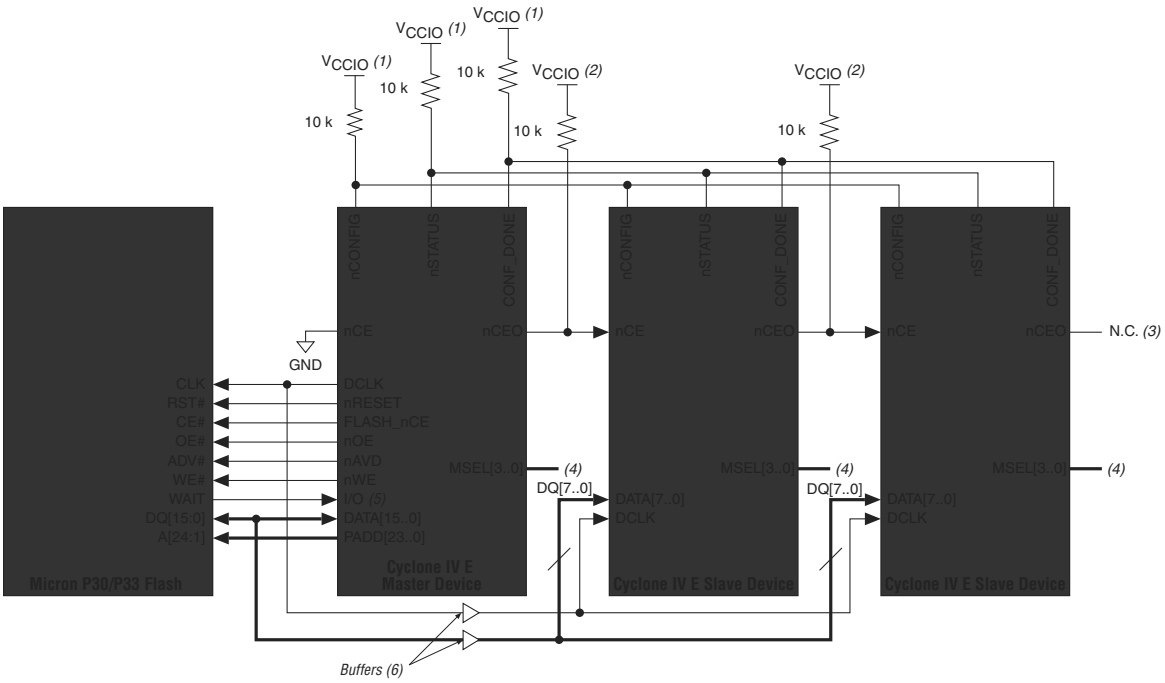
(9) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

In this setup, all the Cyclone IV devices in the chain are connected for concurrent configuration. This reduces the AS configuration time because all the Cyclone IV devices are configured in one configuration cycle. Connect the nCE input pins of all the Cyclone IV devices to GND. You can either leave the nCEO output pins on all the Cyclone IV devices unconnected or use the nCEO output pins as normal user I/O pins. The DATA and DCLK pins are connected in parallel to all the Cyclone IV devices.

Byte-Wide Multi-Device AP Configuration

The simpler method for multi-device AP configuration is the byte-wide multi-device AP configuration. In the byte-wide multi-device AP configuration, the LSB of the DATA[7..0] pin from the flash and master device (set to the AP configuration scheme) is connected to the slave devices set to the FPP configuration scheme, as shown in Figure 8–8.

Figure 8–8. Byte-Wide Multi-Device AP Configuration



Notes to Figure 8–8:

(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 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. You must set the master device in AP mode and the slave devices in FPP mode. To connect MSEL[3..0] for the master device in AP mode and the slave devices in FPP mode, refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the Cyclone IV E master device and slave devices for DATA[15..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.

Word-Wide Multi-Device AP Configuration

The more efficient setup is one in which some of the slave devices are connected to the LSB of the DATA[7..0] and the remaining slave devices are connected to the MSB of the DATA[15..8]. In the word-wide multi-device AP configuration, the nCEO pin of the master device enables two separate daisy chains of slave devices, allowing both chains to be programmed concurrently, as shown in Figure 8–9.

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

Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV (1)	Cyclone IV E (2)	Cyclone IV (1)	Cyclone IV E (2)	
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		—		µs
t_{DH}	Data hold time after rising edge on DCLK	0		—		ns
t_{CD2UM}	CONF_DONE high to user mode (5)	300		650		µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		—		—
t_{CD2UMC}	CONF_DONE high to user mode with **CLKUSR** option on	t_{CD2CU} + (3,192 × CLKUSR period)		—		—
t_{DSU}	Data setup time before rising edge on DCLK	5	8	—	—	ns
t_{CH}	DCLK high time	3.2	6.4	—	—	ns
t_{CL}	DCLK low time	3.2	6.4	—	—	ns
t_{CLK}	DCLK period	7.5	15	—	—	ns
f_{MAX}	DCLK frequency (6)	—	—	133	66	MHz

Notes to Table 8–13:

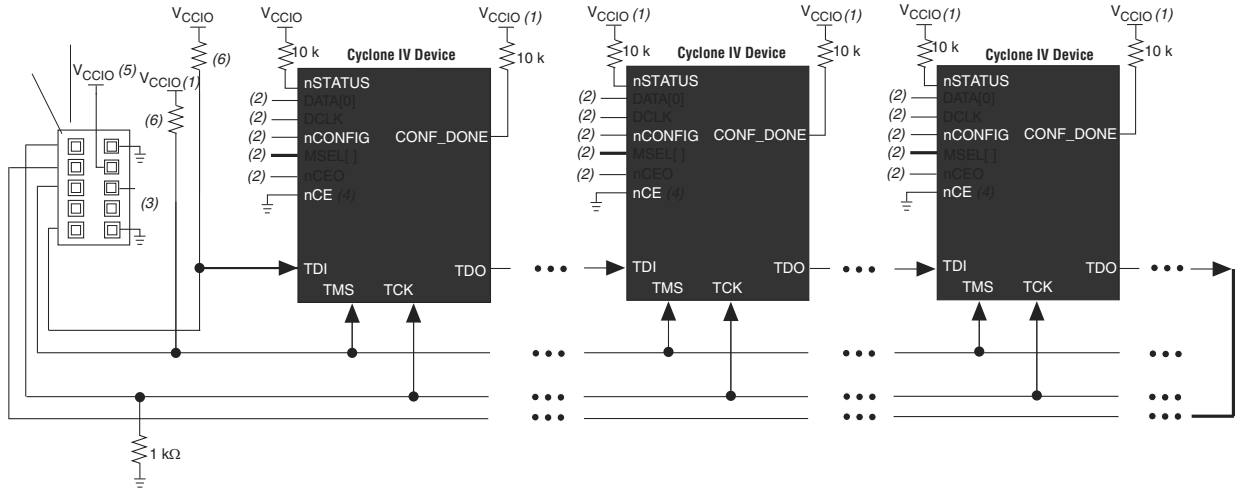
(1) Applicable for Cyclone IV GX and Cyclone IV E with 1.2-V core voltage.
(2) Applicable for Cyclone IV E with 1.0-V core voltage.
(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.
(5) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.
(6) Cyclone IV E devices with 1.0-V core voltage have slower F_{MAX} when compared with Cyclone IV GX devices with 1.2-V core voltage.

JTAG Configuration

JTAG has developed a specification for boundary-scan testing (BST). The BST architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is normally operating. You can also use the JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates **.sof** for JTAG configuration with a download cable in the Quartus II software Programmer.

For more information about the JTAG boundary-scan testing, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

Figure 8–26. JTAG Configuration of Multiple Devices Using a Download Cable (1.2, 1.5, and 1.8-V V_{CCIO} Powering the JTAG Pins)



Notes to Figure 8–26:

(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) In the USB-Blaster and ByteBlaster II cable, 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 or USB-Blaster cable with supply from V_{CCIO}. The ByteBlaster II and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide* and the *USB-Blaster Download Cable User Guide*.

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

If a non-Cyclone IV device is cascaded in the JTAG-chain, TDO of the non-Cyclone IV device driving into TDI of the Cyclone IV device must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

The CONF_DONE and nSTATUS signals are shared in multi-device AS, AP, PS, and FPP configuration chains to ensure that the devices enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 8–25 or Figure 8–26, in which each of the CONF_DONE and nSTATUS signals are isolated so that each device can enter user mode individually.

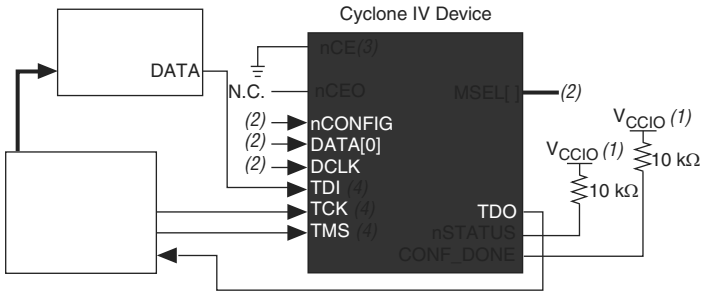
After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device drives the nCE pin of the next device low when it has successfully been JTAG configured. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

JTAG configuration allows an unlimited number of Cyclone IV devices to be cascaded in a JTAG chain.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

Figure 8–27 shows JTAG configuration with a Cyclone IV device and a microprocessor.

Figure 8–27. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 8–27:

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain.

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

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. Signals driving into `TDI`, `TMS`, and `TCK` must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

Configuring Cyclone IV Devices with Jam STAPL

Jam™ STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard. The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam Player, visit the Altera website (www.altera.com).

Configuring Cyclone IV Devices with the JRunner Software Driver

The JRunner software driver allows you to configure Cyclone IV devices through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The supported programming input file is in **.rbf** format. The JRunner software driver also requires a Chain Description File (**.cdf**) generated by the Quartus II software. The JRunner software driver is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS). You can customize the code to make it run on your embedded platform.

EN_ACTIVE_CLK

The EN_ACTIVE_CLK instruction causes the CLKUSR pin signal to replace the internal oscillator as the clock source. When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. After this instruction is issued, other JTAG instructions can be issued while the CLKUSR pin signal remains as the clock source. The clock source is only reverted back to the internal oscillator by issuing the DIS_ACTIVE_CLK instruction or a POR.

DIS_ACTIVE_CLK

The DIS_ACTIVE_CLK instruction breaks the CLKUSR enable latch set by the EN_ACTIVE_CLK instruction and causes the clock source to revert back to the internal oscillator. After the DIS_ACTIVE_CLK instruction is issued, you must continue to clock the CLKUSR pin for 10 clock cycles.

Changing the Start Boot Address of the AP Flash

In the AP configuration scheme (for Cyclone IV E devices only), you can change the default configuration boot address of the parallel flash memory to any desired address using the APFC_BOOT_ADDR JTAG instruction.

APFC_BOOT_ADDR

The APFC_BOOT_ADDR instruction is for Cyclone IV E devices only and allows you to define a start boot address for the parallel flash memory in the AP configuration scheme.

This instruction shifts in a start boot address for the AP flash. When this instruction becomes the active instruction, the TDI and TDO pins are connected through a 22-bit active boot address shift register. The shifted-in boot address bits get loaded into the 22-bit AP boot address update register, which feeds into the AP controller. The content of the AP boot address update register can be captured and shifted-out of the active boot address shift register from TDO.

The boot address in the boot address shift register and update register are shifted to the right (in the LSB direction) by two bits versus the intended boot address. The reason for this is that the two LSB of the address are not accessible. When this boot address is fed into the AP controller, two 0s are attached in the end as LSB, thereby pushing the shifted-in boot address to the left by two bits, which become the actual AP boot address the AP controller gets.

If you have enabled the remote update feature, the APFC_BOOT_ADDR instruction sets the boot address for the factory configuration only.

The APFC_BOOT_ADDR instruction is retained after reconfiguration while the system board is still powered on. However, you must reprogram the instruction whenever you restart the system board.

Cyclone IV Device Handbook,

Volume 2

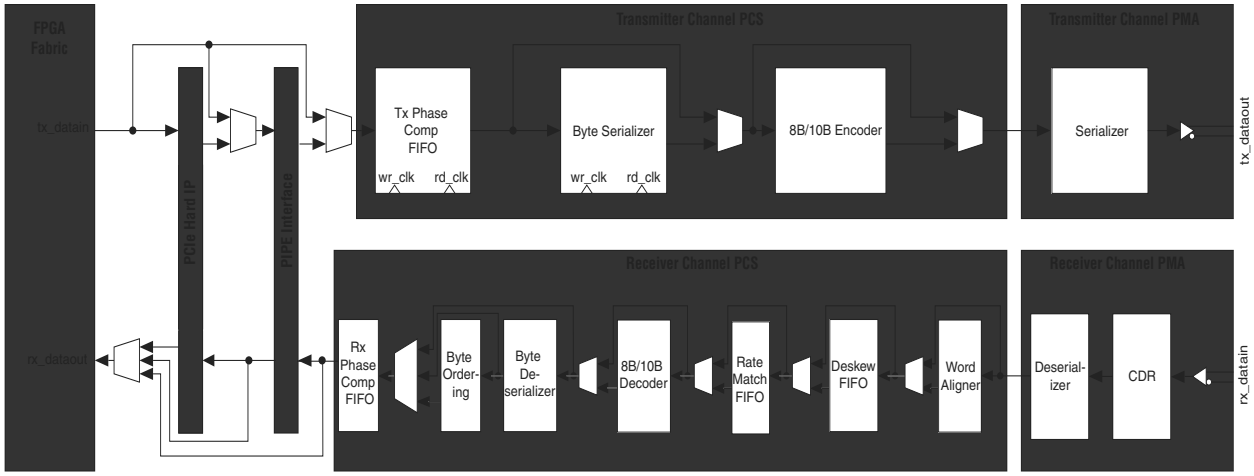
101 Innovation Drive
San Jose, CA 95134
www.altera.com

CYIV-5V2-1.9

Architectural Overview

Figure 1–3 shows the Cyclone IV GX transceiver channel datapath.

Figure 1–3. Transceiver Channel Datapath for Cyclone IV GX Devices



Each transceiver channel consists of a transmitter and a receiver datapath. Each datapath is further structured into the following:

- Physical media attachment (PMA)—includes analog circuitry for I/O buffers, clock data recovery (CDR), serializer/deserializer (SERDES), and programmable pre-emphasis and equalization to optimize serial data channel performance.
- Physical coding sublayer (PCS)—includes hard logic implementation of digital functionality within the transceiver that is compliant with supported protocols.

Outbound parallel data from the FPGA fabric flows through the transmitter PCS and PMA, is transmitted as serial data. Received inbound serial data flows through the receiver PMA and PCS into the FPGA fabric. The transceiver supports the following interface widths:

- FPGA fabric-transceiver PCS—8, 10, 16, or 20 bits
- PMA-PCS—8 or 10 bits

The transceiver channel interfaces through the PIPE when configured for PCIe protocol implementation. The PIPE is compliant with version 2.00 of the *PHY Interface for the PCI Express Architecture* specification.

Transceiver Clocking Architecture

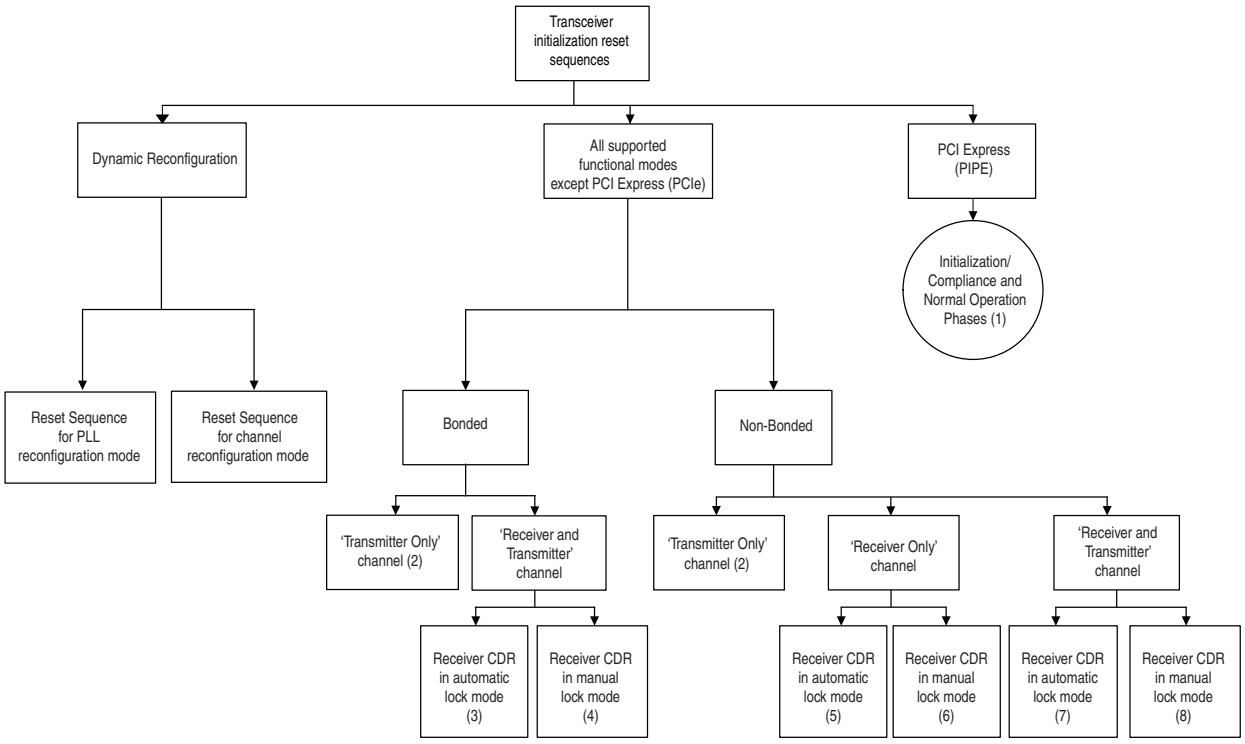
The multipurpose PLLs and general-purpose PLLs located on the left side of the device generate the clocks required for the transceiver operation. The following sections describe the Cyclone IV GX transceiver clocking architecture:

The busy signal remains low for the first reconfig_clk clock cycle. It then gets asserted from the second reconfig_clk clock cycle. Subsequent deassertion of the busy signal indicates the completion of the offset cancellation process. This busy signal is required in transceiver reset sequences except for transmitter only channel configurations. Refer to the reset sequences shown in Figure 2–2 and the associated references listed in the notes for the figure.

Altera strongly recommends adhering to these reset sequences for proper operation of the Cyclone IV GX transceiver.

Figure 2–2 shows the transceiver reset sequences for Cyclone IV GX devices.

Figure 2–2. Transceiver Reset Sequences Chart



Notes to Figure 2–2:

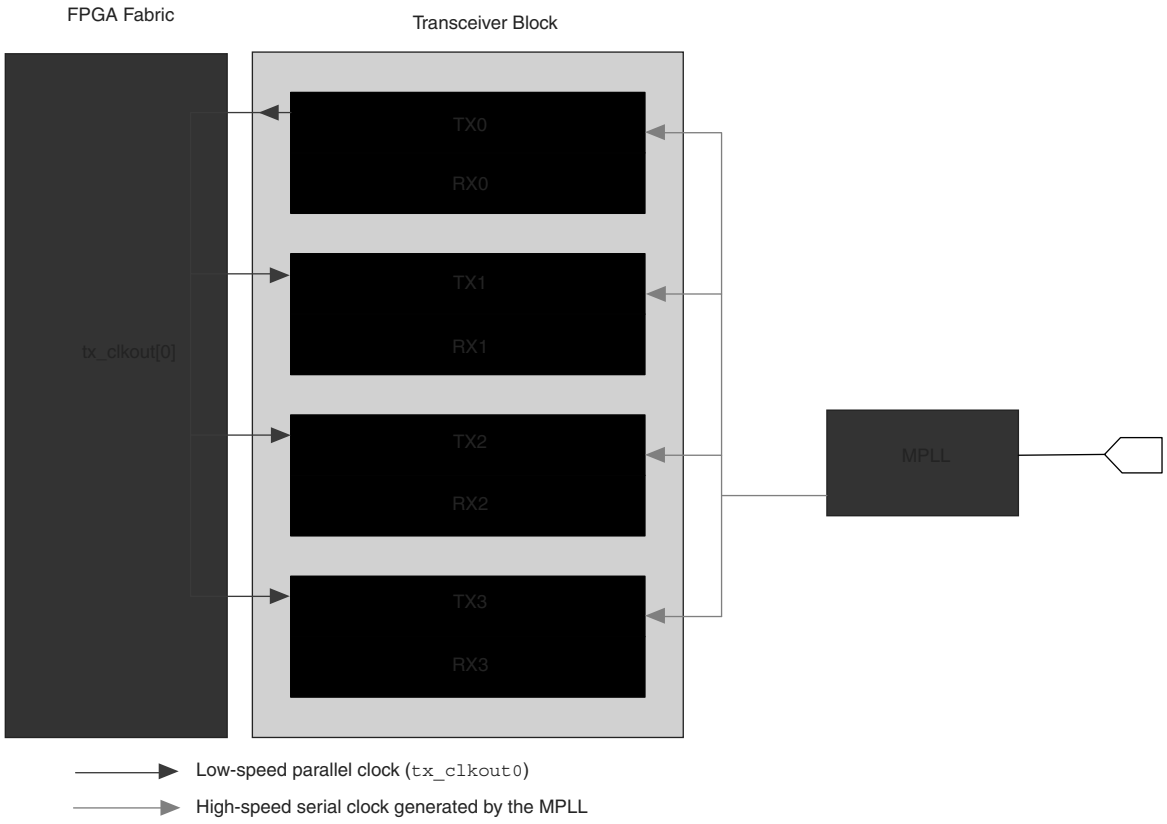
(1) Refer to the Timing Diagram in Figure 2–10.
(2) Refer to the Timing Diagram in Figure 2–3.
(3) Refer to the Timing Diagram in Figure 2–4.
(4) Refer to the Timing Diagram in Figure 2–5.
(5) Refer to the Timing Diagram in Figure 2–6.
(6) Refer to the Timing Diagram in Figure 2–7.
(7) Refer to the Timing Diagram in Figure 2–8.
(8) Refer to the Timing Diagram in Figure 2–9.

Option 1: Share a Single Transmitter Core Clock Between Transmitters

- Enable this option if you want `tx_clkout` of the first channel (channel 0) of the transceiver block to provide the write clock to the Transmitter Phase Compensation FIFOs of the remaining channels in the transceiver block.
- This option is typically enabled when all the channels of a transceiver block have the same functional mode and data rate and are reconfigured to the identical functional mode and data rate.

Figure 3–11 shows the sharing of channel 0's `tx_clkout` between all four regular channels of a transceiver block.

Figure 3–11. Option 1 for Transmitter Core Clocking (Channel Reconfiguration Mode)



Section I. Device Datasheet

This section provides the Cyclone® IV device datasheet. It includes the following chapter:

- Chapter 1, Cyclone IV Device Datasheet

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.

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.