



Welcome to [E-XFL.COM](https://www.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.

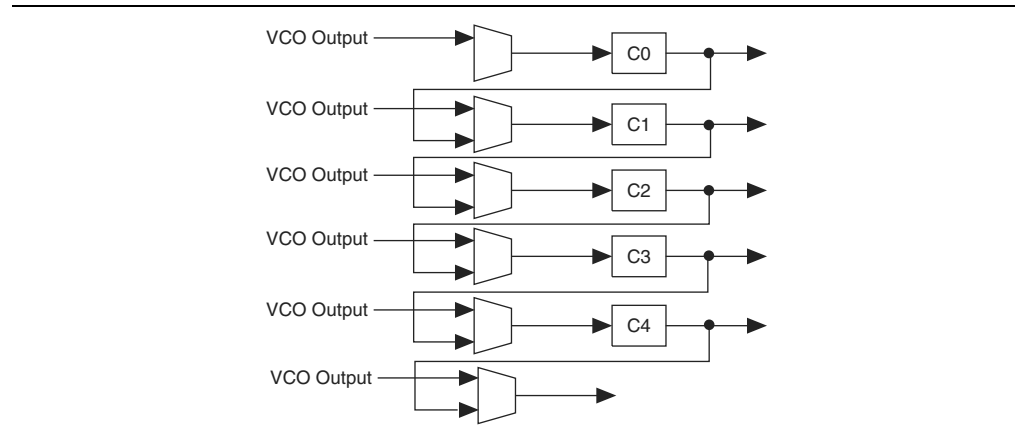
Details

Product Status	Active
Number of LABs/CLBs	4713
Number of Logic Elements/Cells	75408
Total RAM Bits	2810880
Number of I/O	426
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
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/ep4ce75f29i7n

Post-Scale Counter Cascading

PLLs of Cyclone IV devices support post-scale counter cascading to create counters larger than 512. This is implemented by feeding the output of one C counter into the input of the next C counter, as shown in Figure 5-16.

Figure 5-16. Counter Cascading



When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings.

For example, if $C0 = 4$ and $C1 = 2$, the cascaded value is $C0 \times C1 = 8$.



Post-scale counter cascading is automatically set by the Quartus II software in the configuration file. Post-scale counter cascading cannot be performed using the PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. You can achieve the duty cycle setting by a low and high time count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. For example, if the C0 counter is 10, steps of 5% are possible for duty cycle choices between 5 to 90%.

Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

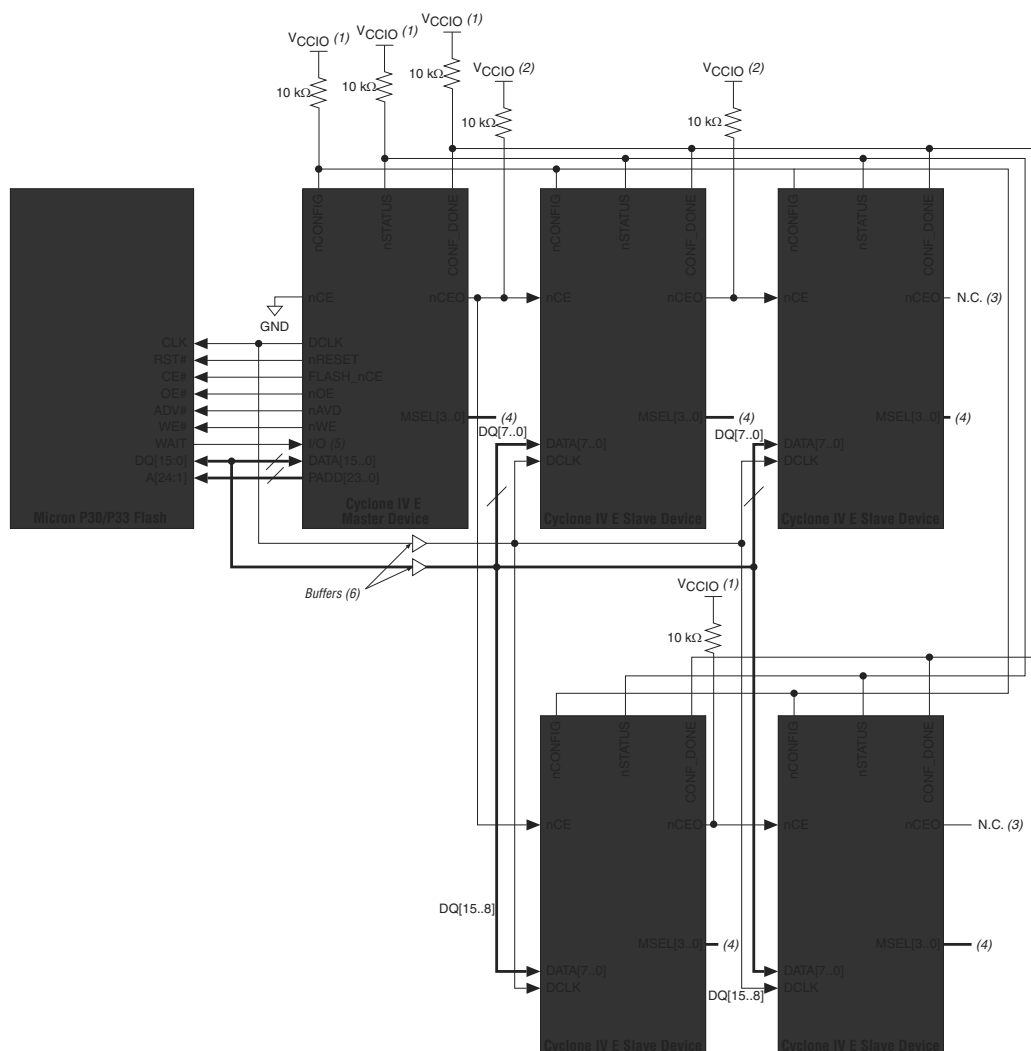
PLL Control Signals

You can use the `pfdena`, `areset`, and `locked` signals to observe and control the PLL operation and resynchronization.



For more information about the PLL control signals, refer to the *ALTPLL Megafunction User Guide*.

Figure 8–9. Word-Wide Multi-Device AP Configuration



- (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 pin 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.



In a multi-device AP configuration, the board trace length between the parallel flash and the master device must follow the recommendations listed in Table 8-11.

The `nSTATUS` and `CONF_DONE` pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27. These pins are open-drain bidirectional pins on the devices. When the first device asserts `nCEO` (after receiving all its configuration data), it releases its `CONF_DONE` pin. However, the subsequent devices in the chain keep this shared `CONF_DONE` line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release `CONF_DONE`, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

Guidelines for Connecting Parallel Flash to Cyclone IV E Devices for an AP Interface

For single- and multi-device AP configuration, the board trace length and loading between the supported parallel flash and Cyclone IV E devices must follow the recommendations listed in Table 8–11. These recommendations also apply to an AP configuration with multiple bus masters.

Table 8–11. Maximum Trace Length and Loading for AP Configuration

Cyclone IV E AP Pins	Maximum Board Trace Length from Cyclone IV E Device to Flash Device (inches)	Maximum Board Load (pF)
DCLK	6	15
DATA[15..0]	6	30
PADD[23..0]	6	30
nRESET	6	30
Flash_nCE	6	30
nOE	6	30
nAVD	6	30
nWE	6	30
I/O ⁽¹⁾	6	30

Note to Table 8–11:

- (1) The AP configuration ignores the `WAIT` signal from the flash 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.

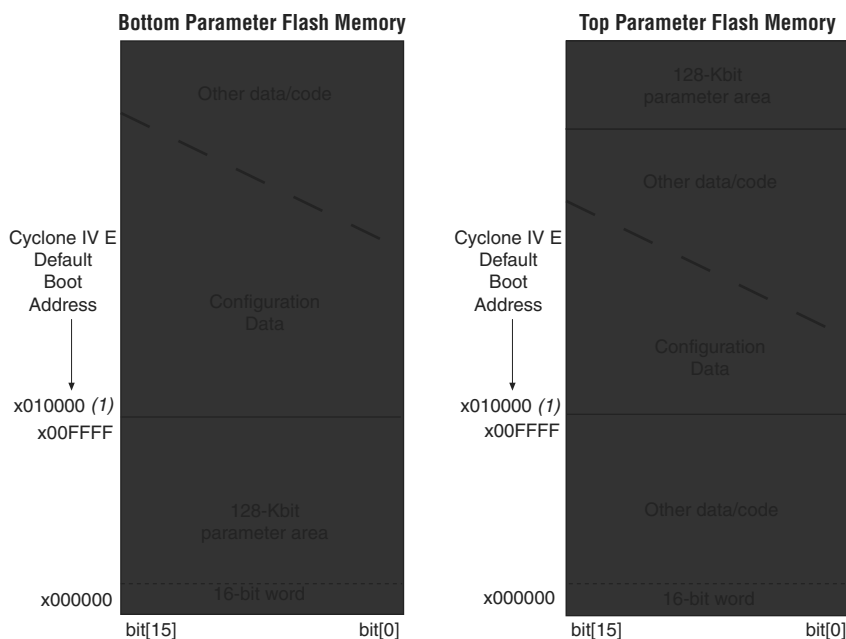
Configuring With Multiple Bus Masters

Similar to the AS configuration scheme, the AP configuration scheme supports multiple bus masters for the parallel flash. For another master to take control of the AP configuration bus, the master must assert `nCONFIG` low for at least 500 ns to reset the master Cyclone IV E device and override the weak 10-k Ω pull-down resistor on the `nCE` pin. This resets the master Cyclone IV E device and causes it to tri-state its AP configuration bus. The other master device then takes control of the AP configuration bus. After the other master device is done, it releases the AP configuration bus, then releases the `nCE` pin, and finally pulses `nCONFIG` low to restart the configuration.

In the AP configuration scheme, multiple masters share the parallel flash. Similar to the AS configuration scheme, the bus control is negotiated by the `nCE` pin.

The default configuration boot address allows the system to use special parameter blocks in the flash memory map. Parameter blocks are at the top or bottom of the memory map. Figure 8–12 shows the configuration boot address in the AP configuration scheme. You can change the default configuration default boot address 0x010000 to any desired address using the APFC_BOOT_ADDR JTAG instruction. For more information about the APFC_BOOT_ADDR JTAG instruction, refer to “JTAG Instructions” on page 8–57.

Figure 8–12. Configuration Boot Address in AP Flash Memory Map



Note to Figure 8–12:

(1) The default configuration boot address is x010000 when represented in 16-bit word addressing.

PS Configuration

You can perform PS configuration on Cyclone IV devices with an external intelligent host, such as a MAX[®] II device, microprocessor with flash memory, or a download cable. In the PS scheme, an external host controls the configuration. Configuration data is clocked into the target Cyclone IV device through DATA[0] at each rising edge of DCLK.

If your system already contains a common flash interface (CFI) flash memory, you can use it for Cyclone IV device configuration storage as well. The MAX II PFL feature provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control the configuration from the flash memory device to the Cyclone IV device.



For more information about the PFL, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.



Cyclone IV devices do not support enhanced configuration devices for PS configuration.

FPP Configuration

The FPP configuration in Cyclone IV devices is designed to meet the increasing demand for faster configuration time. Cyclone IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Cyclone IV devices with an intelligent host, such as a MAX II device or microprocessor with flash memory. If your system already contains a CFI flash memory, you can use it for the Cyclone IV device configuration storage as well. The MAX II PFL feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control configuration from the flash memory device to the Cyclone IV device.



For more information about the PFL, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.



FPP configuration is supported in EP4CGX30 (only for F484 package), EP4CGX50, EP4CGX75, EP4CGX110, EP4CGX150, and all Cyclone IV E devices.



The FPP configuration is not supported in E144 package of Cyclone IV E devices.



Cyclone IV devices do not support enhanced configuration devices for FPP configuration.

FPP Configuration Using an External Host

FPP configuration using an external host provides a fast method to configure Cyclone IV devices. In the FPP configuration scheme, you can use an external host device to control the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store configuration data in an **.rbf**, **.hex**, or **.tff** format. When using the external host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to

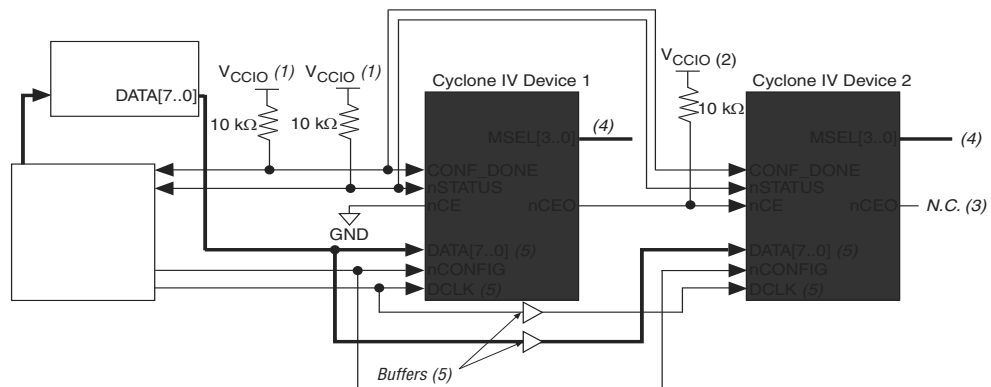
To ensure that DCLK and DATA[0] are not left floating at the end of the configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you choose the FPP scheme in the Quartus II software, 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 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 the CONF_DONE and INIT_DONE pins 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 8-20 shows how to configure multiple devices with a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Cyclone IV devices are cascaded for multi-device configuration.

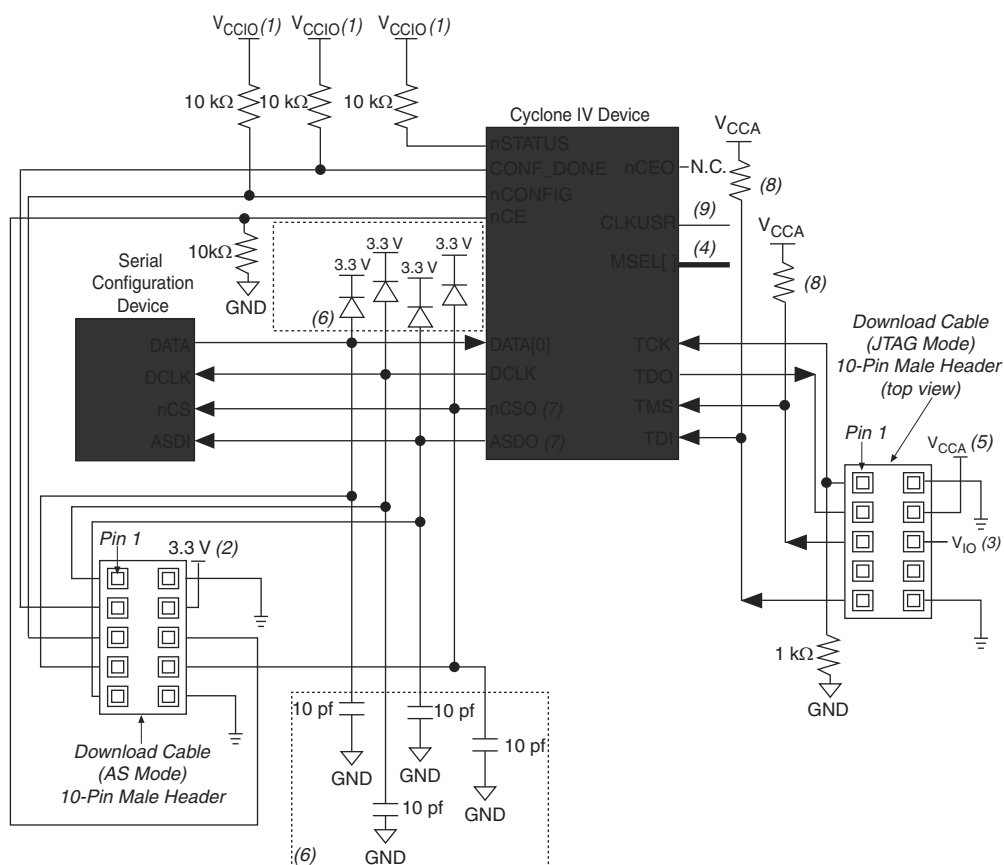
Figure 8-20. Multi-Device FPP Configuration Using an External Host



Notes to Figure 8-20:

- (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 8-4 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[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8-1 on page 8-5.

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. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS,

Figure 8–28. Combining JTAG and AS Configuration Schemes**Notes to Figure 8–28:**

- (1) Connect these pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.
- (2) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, or USB-Blaster cable with the 3.3-V supply.
- (3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. The V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using the USB-Blaster and ByteBlaster II cables, this pin is connected to nCEO when it is used for AS programming, otherwise it is a no connect.
- (4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.
- (5) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V V_{CCA} supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.
- (6) You must place the diodes and capacitors as close as possible to the Cyclone IV device. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.
- (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 $\text{DATA}[1]$ pin in AP and FPP modes.
- (8) Resistor value can vary from 1 k Ω to 10 k Ω .
- (9) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK .

Table 9–7 lists the input and output ports that you must include in the atom.

Table 9–7. CRC Block Input and Output Ports

Port	Input/Output	Definition
<code><crcblock_name></code>	Input	Unique identifier for the CRC block, and represents any identifier name that is legal for the given description language (for example, Verilog HDL, VHDL, and AHDL). This field is required.
<code>.clk (<clock source>)</code>	Input	This signal designates the clock input of this cell. All operations of this cell are with respect to the rising edge of the clock. Whether it is the loading of the data into the cell or data out of the cell, it always occurs on the rising edge. This port is required.
<code>.shiftnld (<shiftnld source>)</code>	Input	This signal is an input into the error detection block. If <code>shiftnld=1</code> , the data is shifted from the internal shift register to the <code>regout</code> at each rising edge of <code>clk</code> . If <code>shiftnld=0</code> , the shift register parallel loads either the pre-calculated CRC value or the update register contents, depending on the <code>ldsrc</code> port input. To do this, the <code>shiftnld</code> must be driven low for at least two clock cycles. This port is required.
<code>.ldsrc (<ldsrc source>)</code>	Input	This signal is an input into the error detection block. If <code>ldsrc=0</code> , the pre-computed CRC register is selected for loading into the 32-bit shift register at the rising edge of <code>clk</code> when <code>shiftnld=0</code> . If <code>ldsrc=1</code> , the signature register (result of the CRC calculation) is selected for loading into the shift register at the rising edge of <code>clk</code> when <code>shiftnld=0</code> . This port is ignored when <code>shiftnld=1</code> . This port is required.
<code>.crcerror (<crcerror indicator output>)</code>	Output	This signal is the output of the cell that is synchronized to the internal oscillator of the device (80-MHz internal oscillator) and not to the <code>clk</code> port. It asserts high if the error block detects that a SRAM bit has flipped and the internal CRC computation has shown a difference with respect to the pre-computed value. You must connect this signal either to an output pin or a bidirectional pin. If it is connected to an output pin, you can only monitor the <code>CRC_ERROR</code> pin (the core cannot access this output). If the <code>CRC_ERROR</code> signal is used by core logic to read error detection logic, you must connect this signal to a <code>BIDIR</code> pin. The signal is fed to the core indirectly by feeding a <code>BIDIR</code> pin that has its output enable port connected to V_{CC} (see Figure 9–3 on page 9–8).
<code>.regout (<registered output>)</code>	Output	This signal is the output of the error detection shift register synchronized to the <code>clk</code> port to be read by core logic. It shifts one bit at each cycle, so you should clock the <code>clk</code> signal 31 cycles to read out the 32 bits of the shift register.

Recovering from CRC Errors

The system that the Altera FPGA resides in must control device reconfiguration. After detecting an error on the `CRC_ERROR` pin, strobing the `nCONFIG` low directs the system to perform the reconfiguration at a time when it is safe for the system to reconfigure the FPGA.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera devices, certain high-reliability applications might require a design to account for these errors.

Document Revision History	2–24
---------------------------------	------

Chapter 3. Cyclone IV Dynamic Reconfiguration

Glossary of Terms	3–1
Dynamic Reconfiguration Controller Architecture	3–2
Dynamic Reconfiguration Controller Port List	3–4
Offset Cancellation Feature	3–10
Functional Simulation of the Offset Cancellation Process	3–12
Dynamic Reconfiguration Modes	3–12
PMA Controls Reconfiguration Mode	3–13
Method 1: Using logical_channel_address to Reconfigure Specific Transceiver Channels	3–14
Method 2: Writing the Same Control Signals to Control All the Transceiver Channels	3–16
Method 3: Writing Different Control Signals for all the Transceiver Channels at the Same Time ..	3–19
Transceiver Channel Reconfiguration Mode	3–21
Channel Interface Reconfiguration Mode	3–22
Data Rate Reconfiguration Mode Using RX Local Divider	3–26
Control and Status Signals for Channel Reconfiguration	3–27
PLL Reconfiguration Mode	3–33
Error Indication During Dynamic Reconfiguration	3–36
Functional Simulation of the Dynamic Reconfiguration Process	3–37
Document Revision History	3–37

- Programmable equalization—boosts the high-frequency gain of the incoming signal up to 7 dB. This compensates for the low-pass filter effects of the transmission media. The amount of high-frequency gain required depends on the loss characteristics of the physical medium.
- Programmable DC gain—provides equal boost to incoming signal across the frequency spectrum with DC gain settings up to 6 dB.
- Programmable differential OCT—provides calibrated OCT at 100 Ω or 150 Ω with on-chip receiver common mode voltage at 0.82 V. The common mode voltage is tri-stated when you disable the OCT to use external termination.
- Offset cancellation—corrects the analog offset voltages that might exist from process variations between the positive and negative differential signals in the equalizer stage and CDR circuit.
- Signal detection—detects if the signal level present at the receiver input buffer is higher than the threshold with a built-in signal threshold detection circuitry. The circuitry has a hysteresis response that filters out any high-frequency ringing caused by ISI effects or high-frequency losses in the transmission medium. Detection is indicated by the assertion of the `rx_signaldetect` signal. Signal detection is only supported when 8B/10B encoder/decoder block is enabled. When not supported, the `rx_signaldetect` signal is forced high, bypassing the signal detection function.



Disable OCT to use external termination if the link requires a 85 Ω termination, such as when you are interfacing with certain PCIe Gen1 or Gen2 capable devices.



For specifications on programmable equalization and DC gain settings, refer to the *Cyclone IV Device Data Sheet*.

Clock Frequency Compensation

In Serial RapidIO mode, the rate match FIFO compensates up to ± 100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock.

Rate matcher is an optional block available for selection in Serial RapidIO mode. However, this block is not fully compliant to the SRIO specification. When enabled in the ALTGX MegaWizard Plug-In Manager, the default settings are:

- control pattern 1 = K28.5 with positive disparity
- skip pattern 1 = K29.7 with positive disparity
- control pattern 2 = K28.5 with negative disparity
- skip pattern 2 = K29.7 with negative disparity

When enabled, the rate match FIFO operation begins after the link is synchronized (indicated by assertion of `rx_syncstatus` from the word aligner). When the rate matcher receives either of the two 10-bit control patterns followed by the respective 10-bit skip pattern, it inserts or deletes the 10-bit skip pattern as necessary to avoid the rate match FIFO from overflowing or under-running. The rate match FIFO can delete/insert a maximum of one skip pattern from a cluster.



The rate match FIFO may perform multiple insertion or deletion if the ppm difference is more than the allowable 200 ppm range. Ensure that the ppm difference in your system is less than 200 ppm.

XAUI Mode

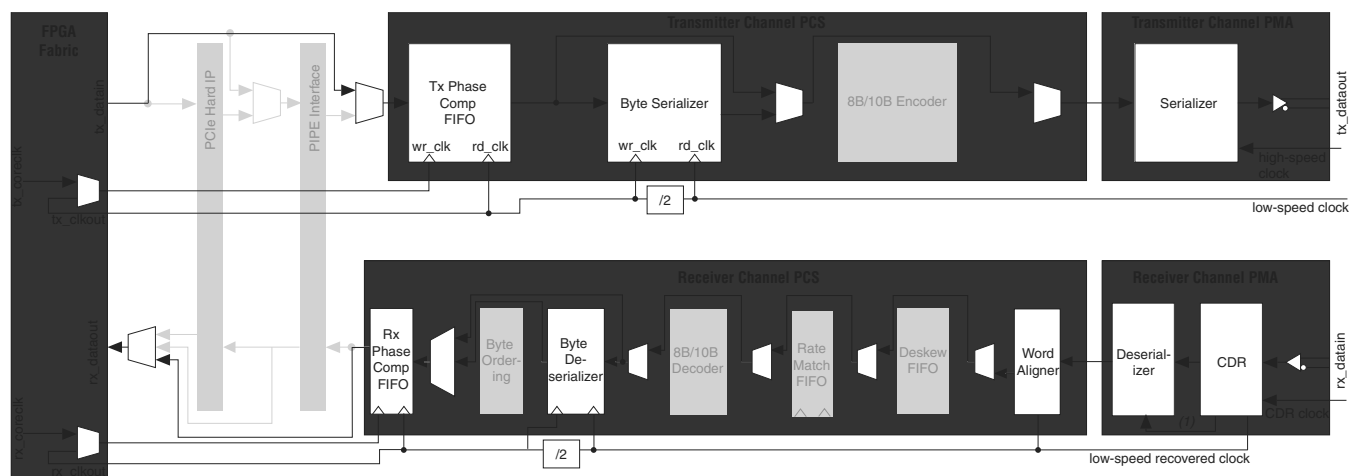
XAUI mode provides the bonded ($\times 4$) transceiver channel datapath configuration for XAUI protocol implementation. The Cyclone IV GX transceivers configured in XAUI mode provides the following functions:

- XGMII-to-PCS code conversion at transmitter datapath
- PCS-to-XGMII code conversion at receiver datapath
- channel deskewing of four lanes
- 8B/10B encoding and decoding
- IEEE P802.3ae-compliant synchronization state machine
- clock rate compensation

The XAUI is a self-managed interface to transparently extend the physical reach of the XGMII between the reconciliation sublayer and the PHY layer in the 10 Gbps LAN as shown in Figure 1-62. The XAUI interface consists of four lanes, each running at 3.125 Gbps with 8B/10B encoded data for a total of actual 10 Gbps data throughput. At the transmit side of the XAUI interface, the data and control characters are

Figure 1-68 shows the transceiver channel datapath and clocking when configured in SDI mode.

Figure 1-68. Transceiver Channel Datapath and Clocking when Configured in SDI Mode



Note to Figure 1-68:

- (1) High-speed recovered clock.

Table 1–28. PIPE Interface Ports in ALTGX Megafunction for Cyclone IV GX⁽¹⁾ (Part 1 of 2)

Port Name	Input/ Output	Clock Domain	Description
fixedclk	Input	Clock signal	125-MHz clock for receiver detect and offset cancellation only in PIPE mode.
tx_detectrxloop	Input	Asynchronous signal	Receiver detect or reverse parallel loopback control. <ul style="list-style-type: none"> ■ A high level in the P1 power state and <code>tx_forceelecidle</code> signal asserted begins the receiver detection operation to determine if there is a valid receiver downstream. This signal must be deasserted when the <code>pipephydonestatus</code> signal indicates receiver detect completion. ■ A high level in the P0 power state with the <code>tx_forceelecidle</code> signal deasserted dynamically configures the channel to support reverse parallel loopback mode.
tx_forcedisp compliance	Input	Asynchronous signal	Force the 8B/10B encoder to encode with negative running disparity. <ul style="list-style-type: none"> ■ Assert only when transmitting the first byte of the PIPE-compliance pattern to force the 8B/10B encoder with a negative running disparity.
pipe8b10binvpolarity	Input	Asynchronous signal	Invert the polarity of every bit of the 10-bit input to the 8B/10B decoder
powerdn	Input	Asynchronous signal	PIPE power state control. <ul style="list-style-type: none"> ■ Signal is 2 bits wide and is encoded as follows: <ul style="list-style-type: none"> ■ 2'b00: P0 (Normal operation) ■ 2'b01: P0s (Low recovery time latency, low power state) ■ 2'b10: P1 (Longer recovery time latency, lower power state) ■ 2'b11: P2 (Lowest power state)
pipedatavalid	Output	N/A	Valid data and control on the <code>rx_dataout</code> and <code>rx_ctrlldetect</code> ports indicator.
pipephydone status	Output	Asynchronous signal	PHY function completion indicator. <ul style="list-style-type: none"> ■ Asserted for one clock cycle to communicate completion of several PHY functions, such as power state transition and receiver detection.
pipeelecidle	Output	Asynchronous signal	Electrical idle detected or inferred at the receiver indicator. <ul style="list-style-type: none"> ■ When electrical idle inference is used, this signal is driven high when it infers an electrical idle condition ■ When electrical idle inference is not used, the <code>rx_signaldetect</code> signal is inverted and driven on this port.

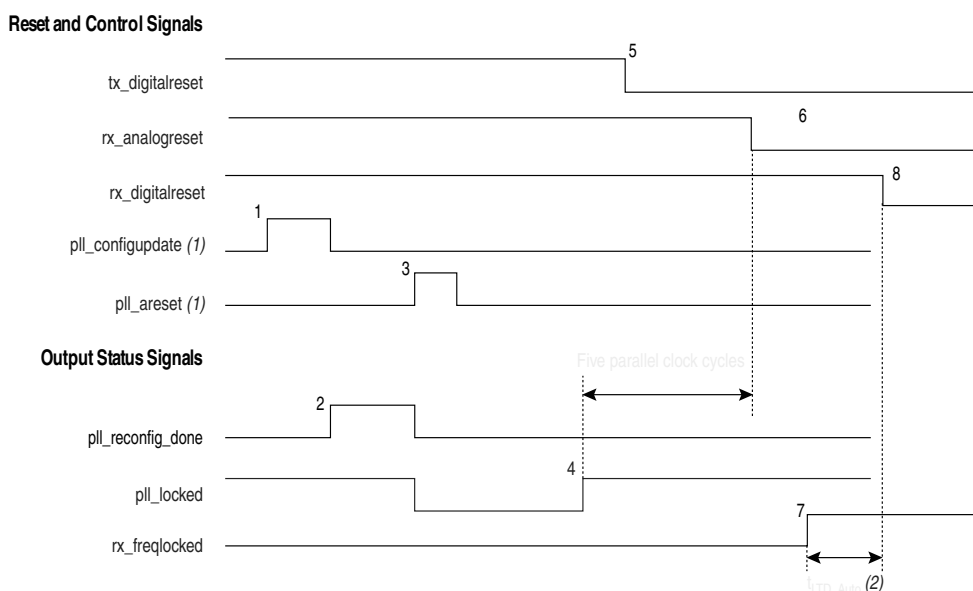
Dynamic Reconfiguration Reset Sequences

When using dynamic reconfiguration in data rate divisions in PLL reconfiguration or channel reconfiguration mode, use the following reset sequences.

Reset Sequence in PLL Reconfiguration Mode

Use the example reset sequence shown in Figure 2-11 when you use the PLL dynamic reconfiguration controller to change the data rate of the transceiver channel. In this example, PLL dynamic reconfiguration is used to dynamically reconfigure the data rate of the transceiver channel configured in Basic $\times 1$ mode with the receiver CDR in automatic lock mode.

Figure 2-11. Reset Sequence When Using the PLL Dynamic Reconfiguration Controller to Change the Data Rate of the Transceiver Channel



Notes to Figure 2-11:

- (1) The `pll_configupdate` and `pll_areset` signals are driven by the `ALTPLL_RECONFIG` megafunction. For more information, refer to *AN 609: Implementing Dynamic Reconfiguration in Cyclone IV GX Devices* and the *Cyclone IV Dynamic Reconfiguration* chapter.
- (2) For t_{LTD_Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.

As shown in Figure 2-11, perform the following reset procedure when using the PLL dynamic reconfiguration controller to change the configuration of the PLLs in the transmitter channel:

1. Assert the `tx_digitalreset`, `rx_analogreset`, and `rx_digitalreset` signals. The `pll_configupdate` signal is asserted (marker 1) by the `ALTPLL_RECONFIG` megafunction after the final data bit is sent out. The `pll_reconfig_done` signal is asserted (marker 2) to inform the `ALTPLL_RECONFIG` megafunction that the scan chain process is completed. The `ALTPLL_RECONFIG` megafunction then asserts the `pll_areset` signal (marker 3) to reset the transceiver PLL.

Table 3–5 describes the rx_dataoutfull[31..0] FPGA fabric-Transceiver channel interface signals.

Table 3–5. rx_dataoutfull[31..0] FPGA Fabric-Transceiver Channel Interface Signal Descriptions (Part 1 of 3)

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
8-bit FPGA fabric-Transceiver Channel Interface	The following signals are used in 8-bit 8B/10B modes:
	rx_dataoutfull[7:0]: 8-bit decoded data (rx_dataout)
	rx_dataoutfull[8]: Control bit (rx_ctrlldetect)
	rx_dataoutfull[9]: Code violation status signal (rx_errdetect)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Disparity error status signal (rx_disperr)
	rx_dataoutfull[12]: Pattern detect status signal (rx_patterndetect)
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfifoatadeleted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfifoatinserted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14:13]: PCI Express (PIPE) functional mode (rx_pipestatus)
	rx_dataoutfull[15]: 8B/10B running disparity indicator (rx_runningdisp)
10-bit FPGA fabric-Transceiver Channel Interface	rx_dataoutfull[9:0]: 10-bit un-encoded data (rx_dataout)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: 8B/10B disparity error indicator (rx_disperr)
	rx_dataoutfull[12]: rx_patterndetect
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfifoatadeleted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfifoatinserted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[15]: 8B/10B running disparity indicator (rx_runningdisp)









Visual Cue	Meaning
Courier type	<p>Indicates signal, port, register, bit, block, and primitive names. For example, <code>data1</code>, <code>tdi</code>, and <code>input</code>. The suffix <code>n</code> denotes an active-low signal. For example, <code>resetn</code>.</p> <p>Indicates command line commands and anything that must be typed exactly as it appears. For example, <code>c:\qdesigns\tutorial\chiptrip.gdf</code>.</p> <p>Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword <code>SUBDESIGN</code>), and logic function names (for example, <code>TRI</code>).</p>
	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.

Table 1–20. Differential I/O Standard Specifications for Cyclone IV Devices ⁽¹⁾ (Part 2 of 2)

I/O Standard	V _{CCIO} (V)			V _{ID} (mV)		V _{ICM} (V) ⁽²⁾			V _{OD} (mV) ⁽³⁾			V _{OS} (V) ⁽³⁾		
	Min	Typ	Max	Min	Max	Min	Condition	Max	Min	Typ	Max	Min	Typ	Max
LVDS (Column I/Os)	2.375	2.5	2.625	100	—	0.05	$D_{MAX} \leq 500 \text{ Mbps}$	1.80	247	—	600	1.125	1.25	1.375
						0.55	$500 \text{ Mbps} \leq D_{MAX} \leq 700 \text{ Mbps}$	1.80						
						1.05	$D_{MAX} > 700 \text{ Mbps}$	1.55						
BLVDS (Row I/Os) ⁽⁴⁾	2.375	2.5	2.625	100	—	—	—	—	—	—	—	—	—	—
BLVDS (Column I/Os) ⁽⁴⁾	2.375	2.5	2.625	100	—	—	—	—	—	—	—	—	—	—
mini-LVDS (Row I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	300	—	600	1.0	1.2	1.4
mini-LVDS (Column I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	300	—	600	1.0	1.2	1.4
RSDS [®] (Row I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.5
RSDS (Column I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.5
PPDS (Row I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.4
PPDS (Column I/Os) ⁽⁵⁾	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.4

Notes to Table 1–20:

- (1) For an explanation of terms used in Table 1–20, refer to “Glossary” on page 1–37.
- (2) V_{IN} range: $0 \text{ V} \leq V_{IN} \leq 1.85 \text{ V}$.
- (3) R_L range: $90 \leq R_L \leq 110 \Omega$.
- (4) There are no fixed V_{IN}, V_{OD}, and V_{OS} specifications for BLVDS. They depend on the system topology.
- (5) The Mini-LVDS, RSDS, and PPDS standards are only supported at the output pins.
- (6) The LVPECL I/O standard is only supported on dedicated clock input pins. This I/O standard is not supported for output pins.

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.

I/O Timing

Use the following methods to determine I/O timing:

- the Excel-based I/O Timing
- the Quartus II timing analyzer

The Excel-based I/O timing provides pin timing performance for each device density and speed grade. The data is typically used prior to designing the FPGA to get a timing budget estimation as part of the link timing analysis. The Quartus II timing analyzer provides a more accurate and precise I/O timing data based on the specifics of the design after place-and-route is complete.



The Excel-based I/O Timing spreadsheet is downloadable from Cyclone IV Devices Literature website.

Glossary

Table 1-46 lists the glossary for this chapter.

Table 1-46. Glossary (Part 1 of 5)

Letter	Term	Definitions
A	—	—
B	—	—
C	—	—
D	—	—
E	—	—
F	f_{HSCLK}	High-speed I/O block: High-speed receiver/transmitter input and output clock frequency.
G	GCLK	Input pin directly to Global Clock network.
	GCLK PLL	Input pin to Global Clock network through the PLL.
H	HSIODR	High-speed I/O block: Maximum/minimum LVDS data transfer rate ($\text{HSIODR} = 1/\text{TUI}$).
I	Input Waveforms for the SSTL Differential I/O Standard	