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	963
Number of Logic Elements/Cells	15408
Total RAM Bits	516096
Number of I/O	343
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
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/ep4ce15f23c8ln

Visual Cue	Meaning
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	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.

Cyclone IV Device Family Architecture

This section describes Cyclone IV device architecture and contains the following topics:

- "FPGA Core Fabric"
- "I/O Features"
- "Clock Management"
- "External Memory Interfaces"
- "Configuration"
- "High-Speed Transceivers (Cyclone IV GX Devices Only)"
- "Hard IP for PCI Express (Cyclone IV GX Devices Only)"

FPGA Core Fabric

Cyclone IV devices leverage the same core fabric as the very successful Cyclone series devices. The fabric consists of LEs, made of 4-input look up tables (LUTs), memory blocks, and multipliers.

Each Cyclone IV device M9K memory block provides 9 Kbits of embedded SRAM memory. You can configure the M9K blocks as single port, simple dual port, or true dual port RAM, as well as FIFO buffers or ROM. They can also be configured to implement any of the data widths in Table 1–7.

Table 1–7. M9K Block Data Widths for Cyclone IV Device Family

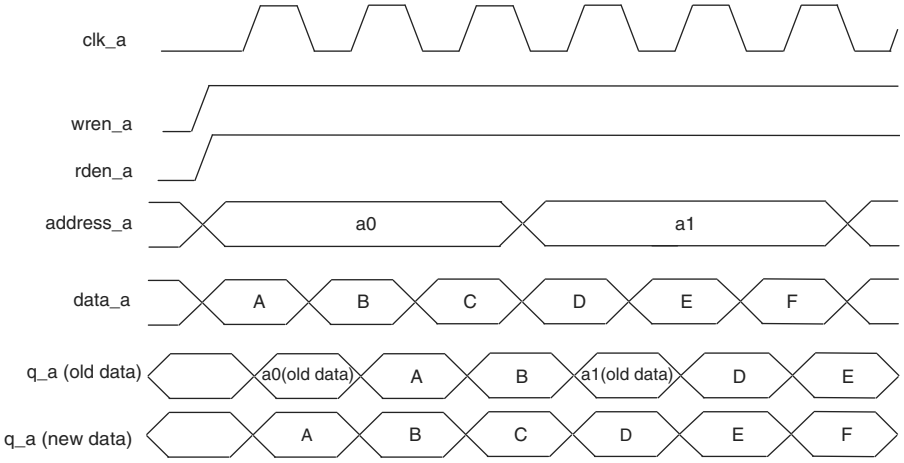
Mode	Data Width Configurations
Single port or simple dual port	×1, ×2, ×4, ×8/9, ×16/18, and ×32/36
True dual port	×1, ×2, ×4, ×8/9, and ×16/18

The multiplier architecture in Cyclone IV devices is the same as in the existing Cyclone series devices. The embedded multiplier blocks can implement an 18 × 18 or two 9 × 9 multipliers in a single block. Altera offers a complete suite of DSP IP including finite impulse response (FIR), fast Fourier transform (FFT), and numerically controlled oscillator (NCO) functions for use with the multiplier blocks. The Quartus® II design software's DSP Builder tool integrates MathWorks Simulink and MATLAB design environments for a streamlined DSP design flow.

For more information, refer to the *Logic Elements and Logic Array Blocks in Cyclone IV Devices*, *Memory Blocks in Cyclone IV Devices*, and *Embedded Multipliers in Cyclone IV Devices* chapters.

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.

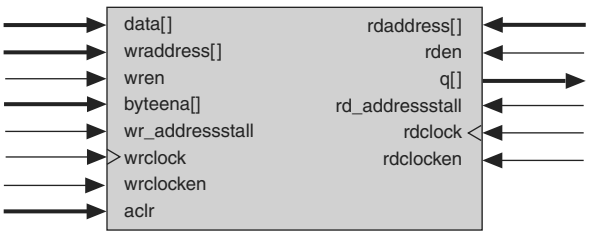
Figure 3–7. Cyclone IV Devices Single-Port Mode Timing Waveform



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.

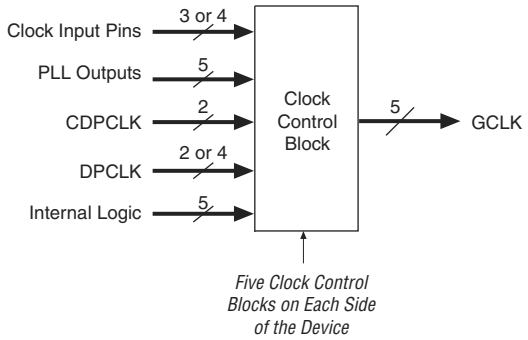
Cyclone IV devices M9K memory blocks support mixed-width configurations, allowing different read and write port widths. Table 3–3 lists mixed-width configurations.

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 5–6 shows a simplified version of the five clock control blocks on each side of the Cyclone IV E device periphery.

Figure 5–6. Clock Control Blocks on Each Side of Cyclone IV E Device *(1)*



Note to Figure 5–6:

(1) The left and right sides of the device have two DPCLK pins; the top and bottom of the device have four DPCLK pins.

GCLK Network Power Down

You can disable a Cyclone IV device's GCLK (power down) using both static and dynamic approaches. In the static approach, configuration bits are set in the configuration file generated by the Quartus II software, which automatically disables unused GCLKs. The dynamic clock enable or disable feature allows internal logic to control clock enable or disable the GCLKs in Cyclone IV devices.

When a clock network is disabled, all the logic fed by the clock network is in an off-state, thereby reducing the overall power consumption of the device. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 5–1 on page 5–11.

You can set the input clock sources and the clkena signals for the GCLK multiplexers through the Quartus II software using the ALTCLKCTRL megafunction.

For more information, refer to the *ALTCLKCTRL Megafunction User Guide*.

clkena Signals

Cyclone IV devices support clkena signals at the GCLK network level. This allows you to gate-off the clock even when a PLL is used. Upon re-enabling the output clock, the PLL does not need a resynchronization or re-lock period because the circuit gates off the clock at the clock network level. In addition, the PLL can remain locked independent of the clkena signals because the loop-related counters are not affected.

Programmable Bandwidth

The PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. PLLs of Cyclone IV devices provide advanced control of the PLL bandwidth using the programmable characteristics of the PLL loop, including loop filter and charge pump. The closed-loop gain 3-dB frequency in the PLL determines the PLL bandwidth. The bandwidth is approximately the unity gain point for open loop PLL response.

Phase Shift Implementation

Phase shift is used to implement a robust solution for clock delays in Cyclone IV devices. Phase shift is implemented with a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time are the most accurate methods of inserting delays, because they are based only on counter settings that are independent of process, voltage, and temperature.

You can phase shift the output clocks from the PLLs of Cyclone IV devices in one of two ways:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

Fine resolution phase shifts are implemented by allowing any of the output counters (`C[4..0]`) or the M counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution.

Equation 5–1 shows the minimum delay time that you can insert using this method.

Equation 5–1. Fine Resolution Phase Shift

$$\Phi_{fine} = \frac{T_{VCO}}{8} = \frac{1}{8f_{VCO}} = \frac{N}{8Mf_{REF}}$$

in which f_{REF} is the input reference clock frequency.

For example, if f_{REF} is 100 MHz, N = 1, and M = 8, then f_{VCO} = 800 MHz, and Φ_{fine} = 156.25 ps. The PLL operating frequency defines this phase shift, a value that depends on reference clock frequency and counter settings.

Coarse resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks. Equation 5–2 shows the coarse phase shift.

Equation 5–2. Coarse Resolution Phase Shift

$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

C is the count value set for the counter delay time (this is the initial setting in the PLL usage section of the compilation report in the Quartus II software). If the initial value is 1, *C* – 1 = 0° phase shift.

You can use the Quartus II software with the APU and the appropriate configuration device programming adapter to program serial configuration devices. All serial configuration devices are offered in an 8- or 16-pin small outline integrated circuit (SOIC) package.

In production environments, serial configuration devices are programmed using multiple methods. Altera programming hardware or other third-party programming hardware is used to program blank serial configuration devices before they are mounted onto PCBs. Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system by porting the reference C-based SRunner software driver provided by Altera.

A serial configuration device is programmed in-system by an external microprocessor with the SRunner software driver. The SRunner software driver is a software driver developed for embedded serial configuration device programming, which is easily customized to fit in different embedded systems. The SRunner software driver is able to read a Raw Programming Data (**.rpd**) file and write to serial configuration devices. The serial configuration device programming time, using the SRunner software driver, is comparable to the programming time with the Quartus II software.

For more information about the SRunner software driver, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code at the Altera website.

AP Configuration (Supported Flash Memories)

The AP configuration scheme is only supported in Cyclone IV E devices. In the AP configuration scheme, Cyclone IV E devices are configured using commodity 16-bit parallel flash memory. These external non-volatile configuration devices are industry standard microprocessor flash memories. The flash memories provide a fast interface to access configuration data. The speed up in configuration time is mainly due to the 16-bit wide parallel data bus, which is used to retrieve data from the flash memory.

Some of the smaller Cyclone IV E devices or package options do not support the AP configuration scheme. Table 8–9 lists the supported AP configuration scheme for each Cyclone IV E devices.

Table 8–9. Supported AP Configuration Scheme for Cyclone IV E Devices

Device	Package Options								
	E144	M164	M256	U256	F256	F324	U484	F484	F780
EP4CE6	—	—	—	—	—	—	—	—	—
EP4CE10	—	—	—	—	—	—	—	—	—
EP4CE15	—	—	—	—	—	—	—	✓	—
EP4CE22	—	—	—	—	—	—	—	—	—
EP4CE30	—	—	—	—	—	✓	—	✓	✓
EP4CE40	—	—	—	—	—	✓	✓	✓	✓
EP4CE55	—	—	—	—	—	—	✓	✓	✓
EP4CE75	—	—	—	—	—	—	✓	✓	✓
EP4CE115	—	—	—	—	—	—	—	✓	✓

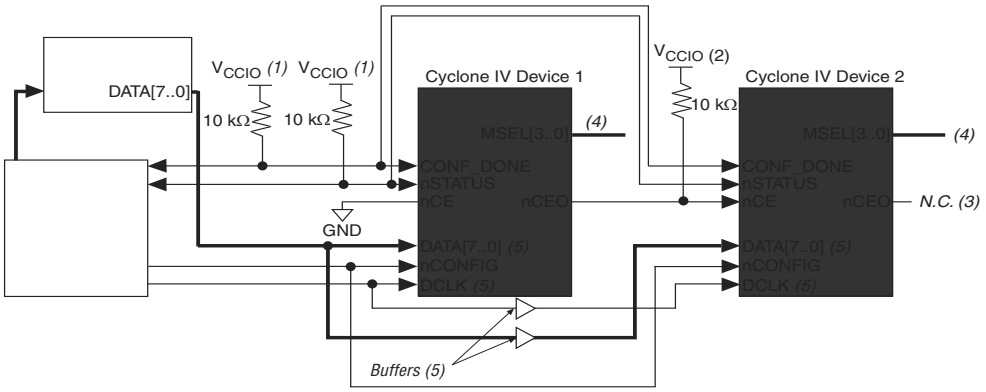
To ensure that DCLK and DATA[0] are not left floating at the end of the configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you choose the FPP scheme in the Quartus II software, 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,

JTAG instructions have precedence over any other configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration in Cyclone IV devices during PS configuration, PS configuration terminates and JTAG configuration begins. If the MSEL pins are set to AS mode, the Cyclone IV device does not output a DCLK signal when JTAG configuration takes place.

The four required pins for a device operating in JTAG mode are TDI, TDO, TMS, and TCK. All the JTAG input pins are powered by the V_{CCIO} pin and support the LVTTL I/O standard only. All user I/O pins are tri-stated during JTAG configuration. Table 8–14 explains the function of each JTAG pin.

Table 8–14. Dedicated JTAG Pins

Pin Name	Pin Type	Description
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data shifts in on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC}. TDI pin has weak internal pull-up resistors (typically 25 kΩ).
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data shifts out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by leaving this pin unconnected.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions in the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC}. TMS pin has weak internal pull-up resistors (typically 25 kΩ).
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to GND. The TCK pin has an internal weak pull-down resistor.

You can download data to the device through the USB-Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV download cable, or the EthernetBlaster communications cable during JTAG configuration. Configuring devices with a cable is similar to programming devices in-system. Figure 8–23 and Figure 8–24 show the JTAG configuration of a single Cyclone IV device.

Remote System Upgrade Mode

In remote update mode, Cyclone IV devices load the factory configuration image after power up. The user-defined factory configuration determines the application configuration to be loaded and triggers a reconfiguration cycle. The factory configuration can also contain application logic.

When used with configuration memory, the remote update mode allows an application configuration to start at any flash sector boundary. Additionally, the remote update mode features a user watchdog timer that can detect functional errors in an application configuration.

Remote Update Mode

In AS configuration scheme, when a Cyclone IV device is first powered up in remote update, it loads the factory configuration located at address `boot_address[23:0] = 24b'0`. Altera recommends storing the factory configuration image for your system at boot address `24b'0`, which corresponds to the start address location `0×000000` in the serial configuration device. A factory configuration image is a bitstream for the Cyclone IV device in your system that is programmed during production and is the fall-back image when an error occurs. This image is stored in non-volatile memory and is never updated or modified using remote access.

When you use the AP configuration in Cyclone IV E devices, the Cyclone IV E device loads the default factory configuration located at the following address after device power-up in remote update mode:
`boot_address[23:0] = 24'h010000 = 24'b1 0000 0000 0000 0000`.

You can change the default factory configuration address to any desired address using the `APFC_BOOT_ADDR` JTAG instruction. The factory configuration image is stored in non-volatile memory and is never updated or modified using remote access. This corresponds to the default start address location `0×010000` represented in 16-bit word addressing (or the updated address if the default address is changed) in the supported parallel flash memory. For more information about the application of the `APFC_BOOT_ADDR` JTAG instruction in AP configuration scheme, refer to the "JTAG Instructions" on page 8–57.

The factory configuration image is user-designed and contains soft logic (Nios II processor or state machine and the remote communication interface) to:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations and store the new configuration data in the local non-volatile memory device
- Determine the application configuration to be loaded into the Cyclone IV device
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

- External configuration reset (nCONFIG) assertion
- User watchdog timer time out

Table 8–24 lists the contents of the current state logic in the status register, when the remote system upgrade master state machine is in factory configuration or application configuration accessing the factory information or application information, respectively. The status register bit in Table 8–24 lists the bit positions in a 32-bit logic.

Table 8–24. Remote System Upgrade Current State Logic Contents In Status Register

Remote System Upgrade Master State Machine	Status Register Bit	Definition	Description
Factory information (1)	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29:24	Reserved bits	Padding bits that are set to all 0's
	23:0	Boot address	The current 24-bit boot address that was used by the configuration scheme as the start address to load the current configuration.
Application information 1 (2)	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29	User watchdog timer enable bit	The current state of the user watchdog enable, which is active high
	28:0	User watchdog timer time-out value	The current entire 29-bit watchdog time-out value.
Application information 2 (2)	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29:24	Reserved bits	Padding bits that are set to all 0's
	23:0	Boot address	The current 24-bit boot address that was used as the start address to load the current configuration

Notes to Table 8–24:

(1) The remote system upgrade master state machine is in factory configuration.

(2) The remote system upgrade master state machine is in application configuration.

The previous two application configurations are available in the previous state registers (previous state register 1 and previous state register 2), but only for debugging purposes.

The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

Quartus II Software Support

Implementation in your design requires a remote system upgrade interface between the Cyclone IV device logic array and remote system upgrade circuitry. You must also generate configuration files for production and remote programming of the system configuration memory. The Quartus II software provides these features.

The two implementation options, ALTREMOTE_UPDATE megafunction and remote system upgrade atom, are for the interface between the remote system upgrade circuitry and the device logic array interface. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

For more information about the ALTREMOTE_UPDATE megafunction, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Document Revision History

Table 8–28 lists the revision history for this chapter.

Table 8–28. Document Revision History (Part 1 of 2)

Date	Version	Changes
May 2013	1.7	■ Added Table 8–6. ■ Updated Table 8–9 to add new device options and packages. ■ Updated Figure 8–16 and Figure 8–22 to include user mode. ■ Updated the “Dedicated” column for DATA[0] and DCLK in Table 8–19. ■ Updated the “User Mode” and “Pin Type” columns for DCLK in Table 8–20.
February 2013	1.6	Updated Table 8–9 to add new device options and packages.
October 2012	1.5	■ Updated “AP Configuration Supported Flash Memories”, “Configuration Data Decompression”, and “Overriding the Internal Oscillator” sections. ■ Updated Figure 8–3, Figure 8–4, Figure 8–5, Figure 8–7, Figure 8–8, Figure 8–9, Figure 8–10, and Figure 8–11. ■ Updated Table 8–2, Table 8–8, Table 8–12, Table 8–13, Table 8–18, and Table 8–19.
November 2011	1.4	■ Added information about how to gain control of EPCS pins. ■ Updated “Reset”, “Single-Device AS Configuration”, “Single-Device AP Configuration”, and “Overriding the Internal Oscillator” sections. ■ Added Table 8–7. ■ Updated Table 8–6 and Table 8–19. ■ Updated Figure 8–3, Figure 8–4, and Figure 8–5.
December 2010	1.3	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Updated Table 8–7, Table 8–10, and Table 8–11. ■ Minor text edits.

In a DC-coupled link, the transmitter DC common mode voltage is seen unblocked at the receiver input buffer as shown in Figure 1–13. The link common mode voltage depends on the transmitter common mode voltage and the receiver common mode voltage. When using the receiver OCT and on-chip biasing circuitry in a DC coupled link, you must ensure the transmitter common mode voltage is compatible with the receiver common mode requirements. If you disable the OCT, you must terminate and bias the receiver externally and ensure compatibility between the transmitter and the receiver common mode voltage.

Figure 1–13. DC-Coupled Link with OCT

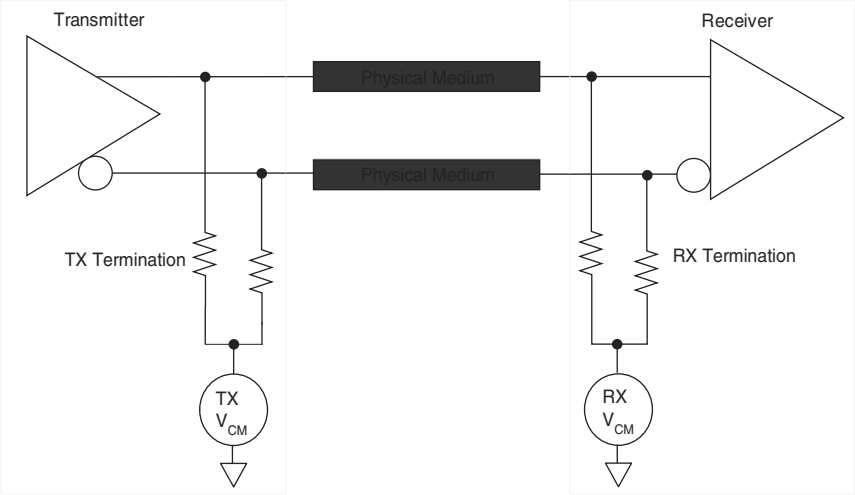
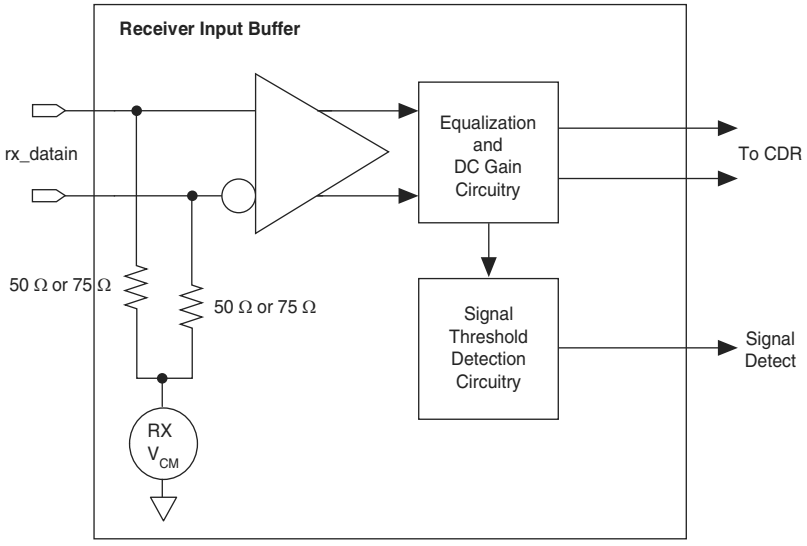


Figure 1–14 shows the receiver input buffer block diagram.

Figure 1–14. Receiver Input Buffer Block Diagram



The receiver input buffers support the following features:

Table 1–11. FPGA Fabric-Transceiver Interface Clocks (Part 2 of 2)

Clock Name	Clock Description	Interface Direction
cal_blk_clk (2)	Transceiver calibration block clock	FPGA fabric to transceiver

Notes to Table 1–11:

(1) Offset cancellation process that is executed after power cycle requires `reconfig_clk` clock. The `reconfig_clk` must be driven with a free-running clock and not derived from the transceiver blocks.

(2) For the supported clock frequency range, refer to the *Cyclone IV Device Data Sheet*.

In the transmitter datapath, TX phase compensation FIFO forms the FPGA fabric-transmitter interface. Data and control signals for the transmitter are clocked with the FIFO write clock. The FIFO write clock supports automatic clock selection by the Quartus II software (depending on channel configuration), or user-specified clock from `tx_coreclk` port. Table 1–12 details the automatic TX phase compensation FIFO write clock selection by the Quartus II software.

The Quartus II software assumes automatic clock selection for TX phase compensation FIFO write clock if you do not enable the `tx_coreclk` port.

Table 1–12. Automatic TX Phase Compensation FIFO Write Clock Selection

Channel Configuration	Quartus II Selection
Non-bonded	`tx_clkout` clock feeds the FIFO write clock. `tx_clkout` is forwarded through the transmitter channel from low-speed clock, which also feeds the FIFO read clock.
Bonded	`coreclkout` clock feeds the FIFO write clock for the bonded channels. `coreclkout` clock is the common bonded low-speed clock, which also feeds the FIFO read clock in the bonded channels.

When using user-specified clock option, ensure that the clock feeding `tx_coreclk` port has 0 ppm difference with the TX phase compensation FIFO read clock.

In the receiver datapath, RX phase compensation FIFO forms the receiver-FPGA fabric interface. Data and status signals from the receiver are clocked with the FIFO read clock. The FIFO read clock supports automatic clock selection by the Quartus II software (depending on channel configuration), or user-specified clock from `rx_coreclk` port. Table 1–13 details the automatic RX phase compensation FIFO read clock selection by the Quartus II software.

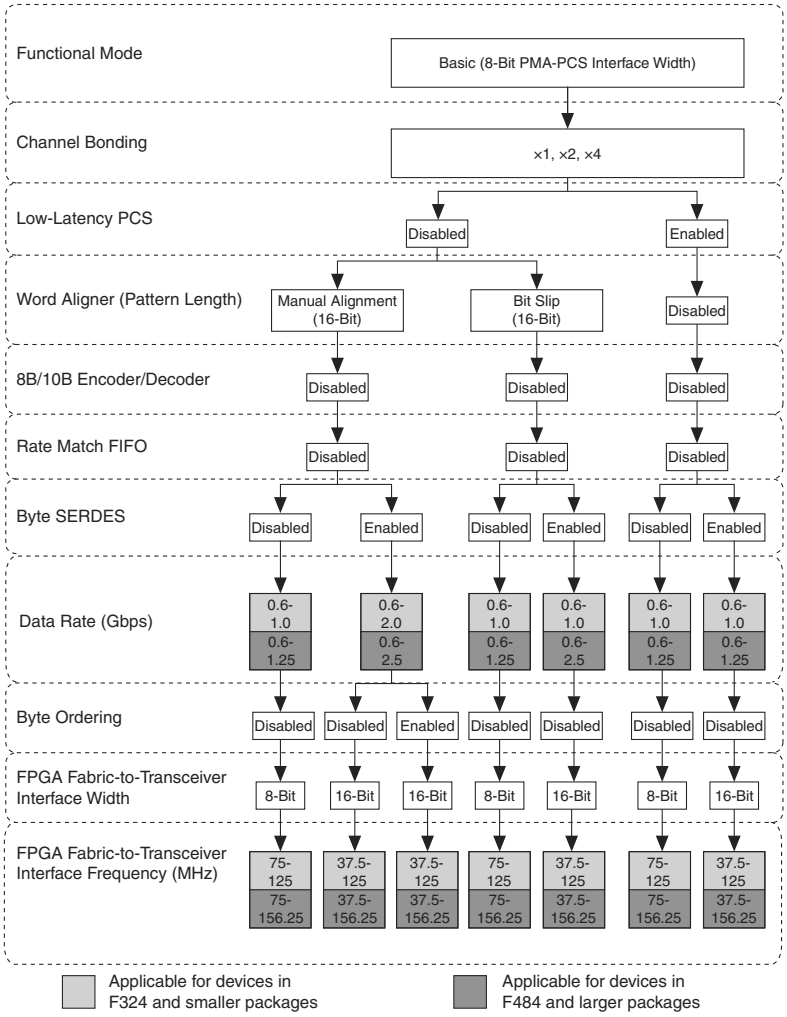
The Quartus II software assumes automatic clock selection for RX phase compensation FIFO read clock if you do not enable the `rx_coreclk` port.

Table 1–13. Automatic RX Phase Compensation FIFO Read Clock Selection (Part 1 of 2)

Channel Configuration		Quartus II Selection
Non-bonded	With rate match FIFO (1)	`tx_clkout` clock feeds the FIFO read clock. `tx_clkout` is forwarded through the receiver channel from low-speed clock, which also feeds the FIFO write clock and transmitter PCS.
	Without rate match FIFO	`rx_clkout` clock feeds the FIFO read clock. `rx_clkout` is forwarded through the receiver channel from low-speed recovered clock, which also feeds the FIFO write clock.

Figure 1–45 and Figure 1–46 show the supported transceiver configurations in Basic mode with the 8-bit and 10-bit PMA-PCS interface width respectively.

Figure 1–45. Supported Transceiver Configurations in Basic Mode with the 8-bit PMA-PCS Interface Width



Clock Frequency Compensation

In GIGE mode, the rate match FIFO compensates up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The GIGE protocol requires the transmitter to send idle ordered sets /I1/ (/K28.5/D5.6/) and /I2/ (/K28.5/D16.2/) during inter-packet gaps, adhering to the rules listed in the IEEE 802.3 specification.

The rate match operation begins after the synchronization state machine in the word aligner indicates synchronization has been acquired by driving the `rx_syncstatus` signal high. The rate match FIFO deletes or inserts both symbols of the /I2/ ordered sets (/K28.5/ and /D16.2/) to prevent the rate match FIFO from overflowing or underflowing. It can insert or delete as many /I2/ ordered sets as necessary to perform the rate match operation.

If you have the auto-negotiation state machine in the FPGA, note that the rate match FIFO is capable of inserting or deleting the first two bytes (/K28.5//D2.2/) of /C2/ ordered sets during auto-negotiation. However, the insertion or deletion of the first two bytes of /C2/ ordered sets can cause the auto-negotiation link to fail. For more information, refer to the Altera Knowledge Base Support Solution.

The status flags `rx_rmfifodatadeleted` and `rx_rmfifodatainserted` to indicate rate match FIFO deletion and insertion events, respectively, are forwarded to the FPGA fabric. These two flags are asserted for two clock cycles for each deleted and inserted /I2/ ordered set.

Figure 1–58 shows an example of rate match FIFO deletion where three symbols must be deleted. Because the rate match FIFO can only delete /I2/ ordered sets, it deletes two /I2/ ordered sets (four symbols deleted).

Figure 1–58. Example of Rate Match FIFO Deletion in GIGE Mode

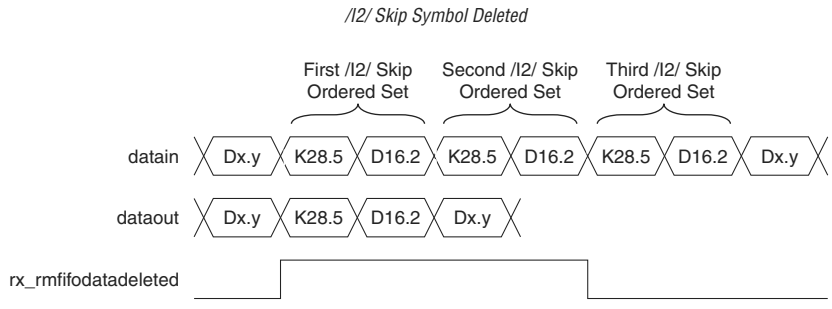
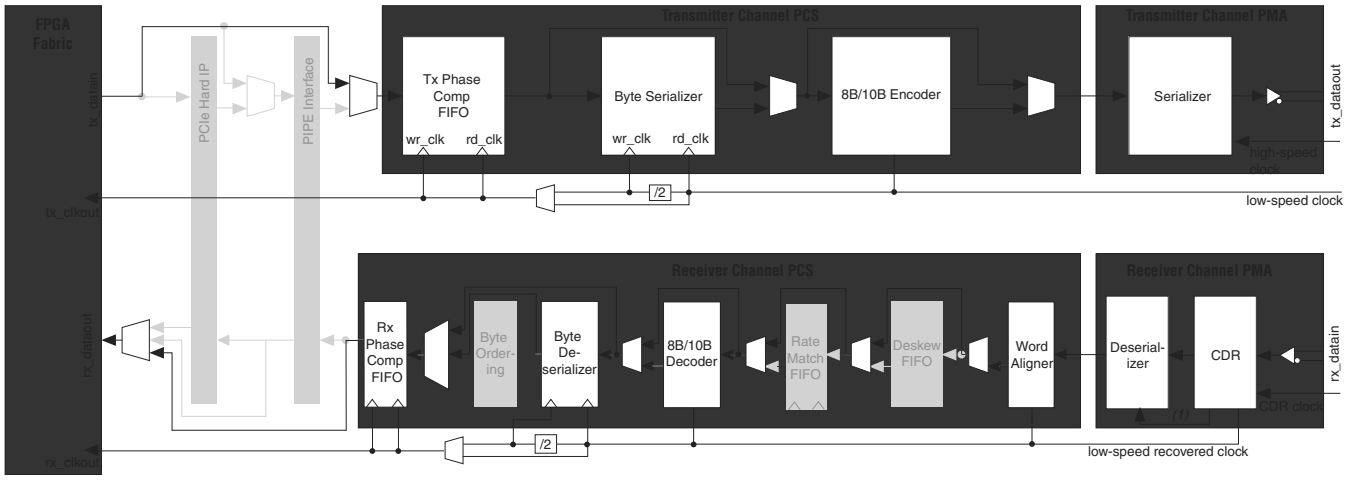


Figure 1–66 shows the transceiver channel datapath and clocking when configured in deterministic latency mode.

Figure 1–66. Transceiver Channel Datapath and Clocking when Configured in Deterministic Latency Mode

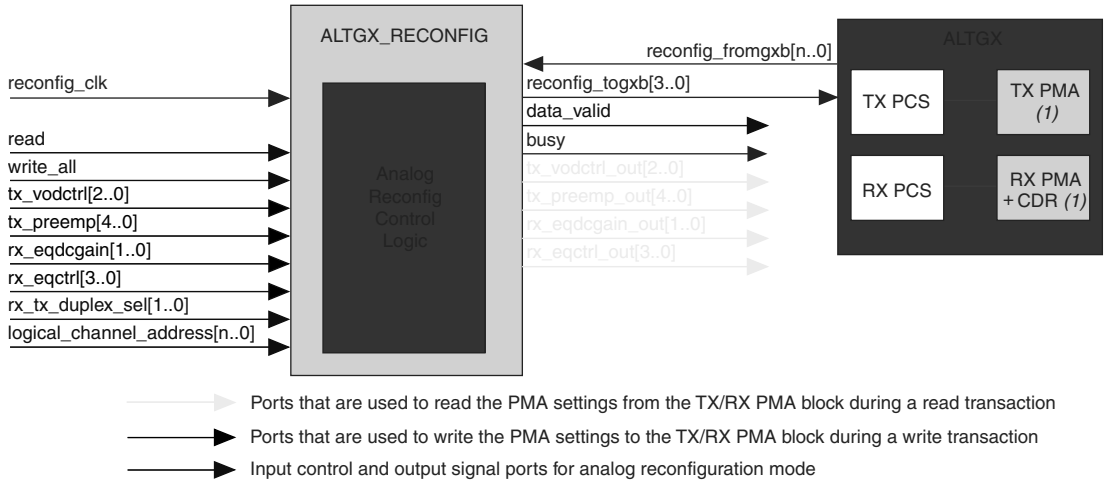


Note to Figure 1–66:

(1) High-speed recovered clock.

Figure 3–9 shows the connection for PMA reconfiguration mode.

Figure 3–9. ALTGX and ALTGX_RECONFIG Connection for PMA Reconfiguration Mode



Note to Figure 3–9:

(1) This block can be reconfigured in PMA reconfiguration mode.

Transceiver Channel Reconfiguration Mode

You can dynamically reconfigure the transceiver channel from an existing functional mode to a different functional mode by selecting the **Channel Reconfiguration** option in ALTGX and ALTGX_RECONFIG MegaWizards. The blocks that are reconfigured by channel reconfiguration mode are the PCS and RX PMA blocks of a transceiver channel.

For more information about reconfiguring the RX PMA blocks of the transceiver channel using channel reconfiguration mode, you can refer to "Data Rate Reconfiguration Mode Using RX Local Divider" on page 3–26.

In channel reconfiguration, only a write transaction can occur; no read transactions are allowed. You can optionally choose to trigger `write_all` once by selecting the continuous write operation in the ALTGX_RECONFIG MegaWizard Plug-In Manager. The Quartus II software then continuously writes all the words required for reconfiguration.

For channel reconfiguration, **.mif** files are required to dynamically reconfigure the transceivers channels in channel reconfiguration modes. The **.mif** carries the reconfiguration information that will be used to reconfigure the transceivers channel dynamically on-the-fly. The **.mif** contents is generated automatically when you select the **Generate GXB Reconfig MIF** option in the Quartus II software setting. For different **.mif** settings, you need to later reconfigure and recompile the ALTGX MegaWizard to generate the **.mif** based on the required reconfiguration settings.

The dynamic reconfiguration controller can optionally perform a continuos write operation or a regular write operation of the **.mif** contents in terms of word size (16-bit data) to the transceivers channel that is selected for reconfiguration.

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

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
20-bit FPGA fabric-Transceiver Channel Interface with PCS-PMA set to 10 bits	Two 10-bit Data (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[25:16] - rx_dataout (MSByte)
	wo Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[27]: 8B/10B disparity error indicator (rx_disperr)
	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
	rx_dataoutfull[15] and rx_dataoutfull[31]: 8B/10B running disparity indicator (rx_runningdisp)

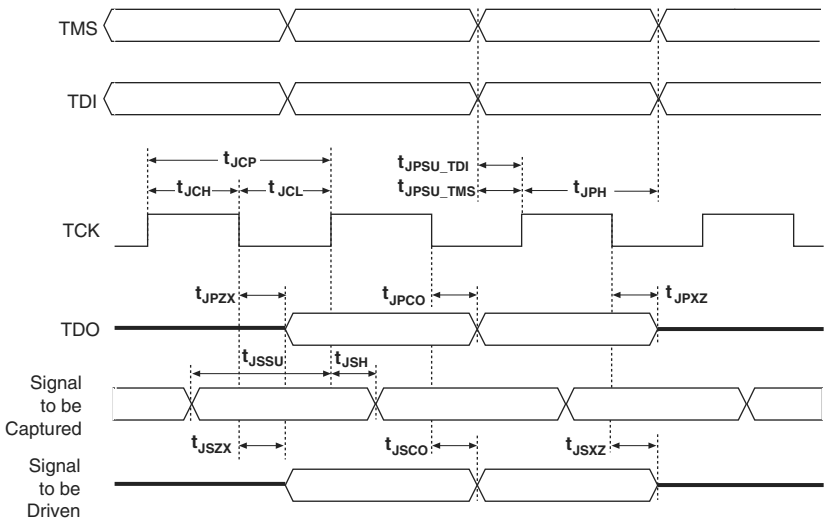
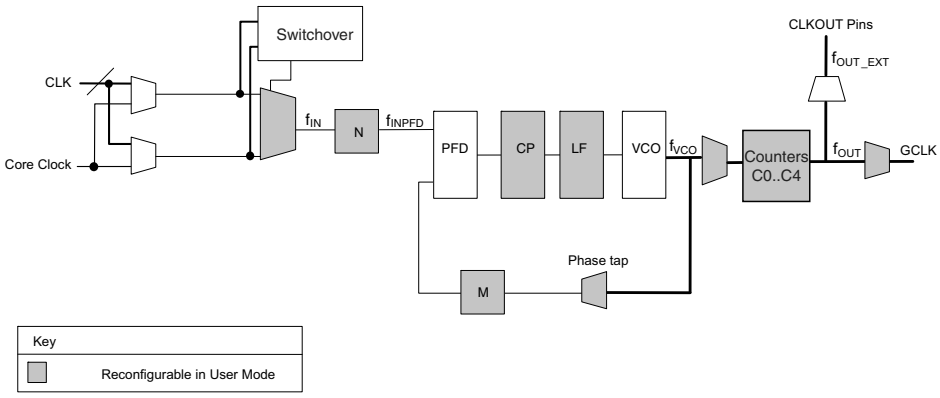
Data Rate Reconfiguration Mode Using RX Local Divider

The RX local divider resides in the RX PMA block for every channels. This is a hardware feature where a /2 divider is available in each of the receiver channel for the supported device. You can use this RX local divider to reconfigure the data rate at the receiver channel. This can be used for protocols such as SDI that has data rates in divisions of 2.

By using this RX local divider, you can support two different data rates without using additional transceiver PLLs. This dynamic reconfiguration mode is available only for the receiver and not applicable to the transmitter. This reconfiguration mode using the RX local divider (/2) is only supported and available in EP4CGX30 (F484 package), EP4CGX50, and EP4CGX75 devices.

For more information about this RX local divider, refer to the *Cyclone IV GX Transceiver Architecture* chapter.

Table 1–46. Glossary (Part 2 of 5)

Letter	Term	Definitions
J	JTAG Waveform	
K	—	—
L	—	—
M	—	—
N	—	—
O	—	—
P	PLL Block	The following highlights the PLL specification parameters: 
Q	—	—