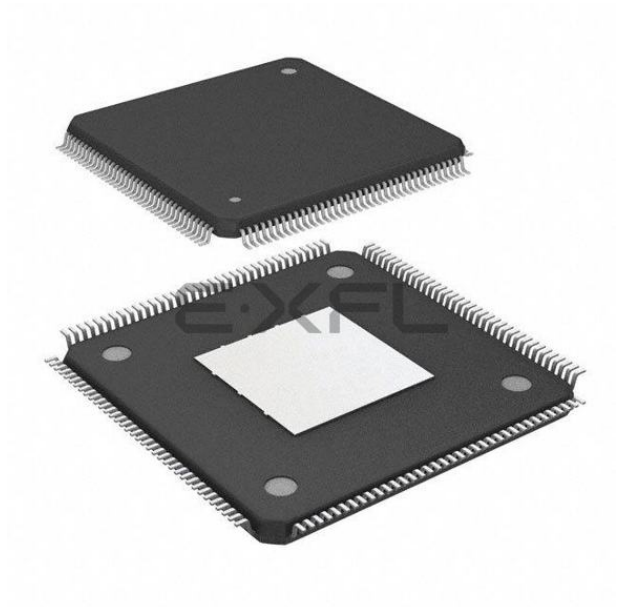


Welcome to E-XFL.COM

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	1395
Number of Logic Elements/Cells	22320
Total RAM Bits	608256
Number of I/O	79
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	144-LQFP Exposed Pad
Supplier Device Package	144-EQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce22e22c8

PLLs in Cyclone IV Devices

Cyclone IV GX devices offer two variations of PLLs: general purpose PLLs and multipurpose PLLs. Cyclone IV E devices only have the general purpose PLLs.

The general purpose PLLs are used for general-purpose applications in the FPGA fabric and periphery such as external memory interfaces. The multipurpose PLLs are used for clocking the transceiver blocks. When the multipurpose PLLs are not used for transceiver clocking, they can be used for general-purpose clocking.

For more details about the multipurpose PLLs used for transceiver clocking, refer to the *Cyclone IV Transceivers* chapter.

Cyclone IV GX devices contain up to eight general purpose PLLs and multipurpose PLLs while Cyclone IV E devices have up to four general purpose PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces.

For more information about the number of general purpose PLLs and multipurpose PLLs in each device density, refer to the *Cyclone IV Device Family Overview* chapter.

The general I/O pins cannot drive the PLL clock input pins.

Table 5–5 lists the features available in Cyclone IV GX PLLs.

Table 5–5. Cyclone IV GX PLL Features (Part 1 of 2)

Features	Availability									
	General Purpose PLLs				Multipurpose PLLs					
	PLL_1 (1), (10)	PLL_2 (1), (10)	PLL_3 (2)	PLL_4 (3)	PLL_1 (4)	PLL_2 (4)	PLL_5 (1), (10)	PLL_6 (1), (10)	PLL_7 (1)	PLL_8 (1)
C (output counters)	5									
M, N, C counter sizes	1 to 512 (5)									
Dedicated clock outputs	1 single-ended or 1 differential pair									
Clock input pins	12 single-ended or 6 differential pairs (6) and 4 differential pairs (7)									
Spread-spectrum input clock tracking	✓ (8)									
PLL cascading	Through GCLK									
Source-Synchronous Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
No Compensation Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Normal Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Zero Delay Buffer Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Deterministic Latency Compensation Mode	✓	✓	—	—	✓	✓	✓	✓	✓	✓
Phase shift resolution (9)	Down to 96 ps increments									
Programmable duty cycle	✓									
Output counter cascading	✓									

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

Programmable Pull-Up Resistor

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

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

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

Programmable Delay

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

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

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

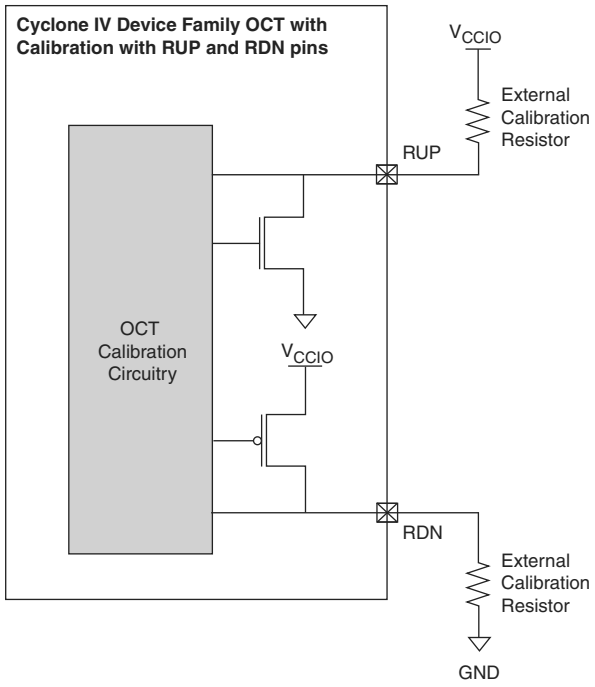
Table 6–1. Cyclone IV Devices Programmable Delay Chain

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

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

Figure 6–3 shows the external calibration resistors setup on the RUP and RDN pins and the associated OCT calibration circuitry.

Figure 6–3. Cyclone IV Devices R_S OCT with Calibration Setup



RUP and RDN pins go to a tri-state condition when calibration is completed or not running. These two pins are dual-purpose I/Os and function as regular I/Os if you do not use the calibration circuit.

On-Chip Series Termination Without Calibration

Cyclone IV devices support driver impedance matching to match the impedance of the transmission line, which is typically 25 or 50 Ω. When used with the output drivers, OCT sets the output driver impedance to 25 or 50 Ω. Cyclone IV devices also support I/O driver series termination (R_S = 50 Ω) for SSTL-2 and SSTL-18.

For more information about Cyclone IV PLL, refer to the *Clock Networks and PLLs in Cyclone IV Devices* chapter.

Document Revision History

Table 7–3 lists the revision history for this chapter.

Table 7–3. Document Revision History

Date	Version	Changes
March 2016	2.6	■ Updated Table 7–1 to remove support for the N148 package. ■ Updated note (1) in Figure 7–2 to remove support for the N148 package. ■ Updated Figure 7–4 to remove support for the N148 package.
May 2013	2.5	Updated Table 7–2 to add new device options and packages.
February 2013	2.4	Updated Table 7–2 to add new device options and packages.
October 2012	2.3	Updated Table 7–1 and Table 7–2.
December 2010	2.2	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Updated Table 7–2. ■ Minor text edits.
November 2010	2.1	Updated “Data and Data Clock/Strobe Pins” section.
February 2010	2.0	■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release. ■ Updated Table 7–1. ■ Added Table 7–2. ■ Added Figure 7–5 and Figure 7–6.
November 2009	1.0	Initial release.

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 **.ttf** 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

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

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

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

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

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

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

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

Programming Serial Configuration Devices In-System with the JTAG Interface

Cyclone IV devices in a single- or multiple-device chain support in-system programming of a serial configuration device with the JTAG interface through the SFL design. The intelligent host or download cable of the board can use the four JTAG pins on the Cyclone IV device to program the serial configuration device in system, even if the host or download cable cannot access the configuration pins (DCLK, DATA, ASDI, and nCS pins).

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

In a multiple device chain, you must only configure the master device that controls the serial configuration device. Slave devices in the multiple device chain that are configured by the serial configuration device do not have to be configured when using this feature. To successfully use this feature, set the MSEL pins of the master device to select the AS configuration scheme (Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9). The serial configuration device in-system programming through the Cyclone IV device JTAG interface has three stages, which are described in the following sections:

- "Loading the SFL Design"
- "ISP of the Configuration Device" on page 8–56
- "Reconfiguration" on page 8–57

Loading the SFL Design

The SFL design is a design inside the Cyclone IV device that bridges the JTAG interface and AS interface with glue logic.

The intelligent host uses the JTAG interface to configure the master device with a SFL design. The SFL design allows the master device to control the access of four serial configuration device pins, also known as the Active Serial Memory Interface (ASMI) pins, through the JTAG interface. The ASMI pins are serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and active-low chip select (nCS) pins.

Table 9–4 defines the registers shown in Figure 9–1.

Table 9–4. Error Detection Registers

Register	Function
32-bit signature register	This register contains the CRC signature. The signature register contains the result of the user mode calculated CRC value compared against the pre-calculated CRC value. If no errors are detected, the signature register is all zeros. A non-zero signature register indicates an error in the configuration CRAM contents. The CRC_ERROR signal is derived from the contents of this register.
32-bit storage register	This register is loaded with the 32-bit pre-computed CRC signature at the end of the configuration stage. The signature is then loaded into the 32-bit CRC circuit (called the Compute and Compare CRC block, as shown in Figure 9–1) during user mode to calculate the CRC error. This register forms a 32-bit scan chain during execution of the CHANGE_EDREG JTAG instruction. The CHANGE_EDREG JTAG instruction can change the content of the storage register. Therefore, the functionality of the error detection CRC circuitry is checked in-system by executing the instruction to inject an error during the operation. The operation of the device is not halted when issuing the CHANGE_EDREG instruction.

Error Detection Timing

When the error detection CRC feature is enabled through the Quartus II software, the device automatically activates the CRC process upon entering user mode after configuration and initialization is complete.

The CRC_ERROR pin is driven low until the error detection circuitry detects a corrupted bit in the previous CRC calculation. After the pin goes high, it remains high during the next CRC calculation. This pin does not log the previous CRC calculation. If the new CRC calculation does not contain any corrupted bits, the CRC_ERROR pin is driven low. The error detection runs until the device is reset.

The error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency.

Table 9–5 lists the minimum and maximum error detection frequencies.

Table 9–5. Minimum and Maximum Error Detection Frequencies for Cyclone IV Devices

Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (2^n)
80 MHz/2^n	80 MHz	312.5 kHz	0, 1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (for more information, refer to "Software Support"). The divisor is a power of two (2), where *n* is between 0 and 8. The divisor ranges from one through 256. Refer to Equation 9–1.

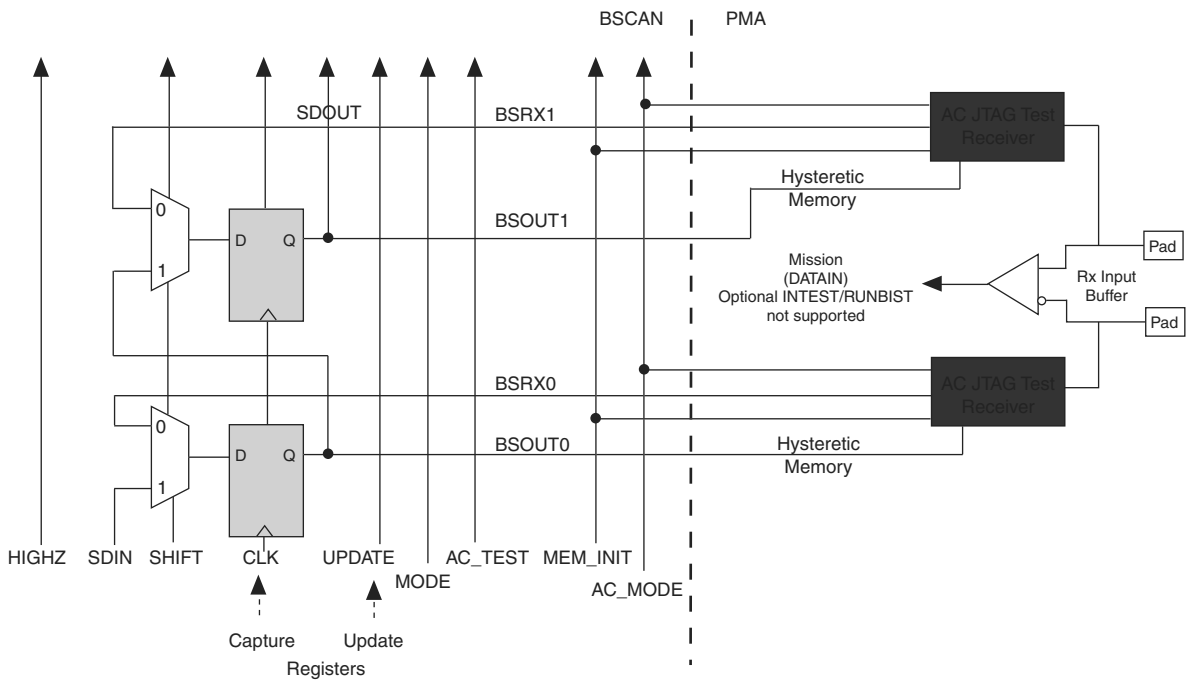
Equation 9–1.

$$\text{error detection frequency} = \frac{80\ \text{MH}}{2^n}$$

CRC calculation time depends on the device and the error detection clock frequency.

Figure 10–2 shows the Cyclone IV GX HSSI receiver BSC.

Figure 10–2. HSSI Receiver BSC with IEEE Std. 1149.6 BST Circuitry for the Cyclone IV GX Devices



For more information about Cyclone IV devices user I/O boundary-scan cells, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

BST Operation Control

Table 10–1 lists the boundary-scan register length for Cyclone IV devices.

Table 10–1. Boundary-Scan Register Length for Cyclone IV Devices (Part 1 of 2)

Device	Boundary-Scan Register Length
EP4CE6	603
EP4CE10	603
EP4CE15	1080
EP4CE22	732
EP4CE30	1632
EP4CE40	1632
EP4CE55	1164
EP4CE75	1314
EP4CE115	1620
EP4CGX15	260
EP4CGX22	494
EP4CGX30 (1)	494
EP4CGX50	1006

For Transmitter and Receiver operation in bonded channel configuration, the receiver PCS supports configuration with rate match FIFO, and configuration without rate match FIFO. Figure 1–39 shows the datapath clocking in Transmitter and Receiver operation with rate match FIFO in ×2 and ×4 bonded channel configurations. For Transmitter and Receiver operation in bonded channel configuration without rate match FIFO, the datapath clocking is identical to Figure 1–38 for the bonded transmitter channels, and Figure 1–34 on page 1–35 for the receiver channels.

In configuration with rate match FIFO, the transmitter datapath clocking is identical to Transmitter Only operation as shown in Figure 1–38. In each bonded receiver channel, the CDR unit recovers the clock from serial received data and generates the high- and low-speed recovered clock for each bonded channel. The high-speed recovered clock feeds the channel's deserializer, and low-speed recovered clock is forwarded to receiver PCS. The individual low-speed recovered clock feeds to the following blocks in the receiver PCS:

- word aligner
- write clock of rate match FIFO

The common bonded low-speed clock that is used in all bonded transmitter PCS datapaths feeds the following blocks in each bonded receiver PCS:

- read clock of rate match FIFO
- 8B/10B decoder
- write clock of byte deserializer
- byte ordering
- write clock of RX phase compensation FIFO

When the byte deserializer is enabled, the common bonded low-speed clock frequency is halved before feeding to the write clock of RX phase compensation FIFO. The common bonded low-speed clock is available in FPGA fabric as `coreclkout` port, which can be used in FPGA fabric to send transmitter data and control signals, and capture receiver data and status signals from the bonded channels.

FPGA Fabric-Transceiver Interface Clocking

The FPGA fabric-transceiver interface clocks consists of clock signals from the FPGA fabric to the transceiver blocks, and from the transceiver blocks to the FPGA fabric. These clock resources use the global clock networks (GCLK) in the FPGA core.

For information about the GCLK resources in the Cyclone IV GX devices, refer to *Clock Networks and PLLs in Cyclone IV Devices* chapter.

Table 1–11 lists the FPGA fabric-transceiver interface clocks.

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

Clock Name	Clock Description	Interface Direction
tx_clkout	Phase compensation FIFO clock	Transceiver to FPGA fabric
rx_clkout	Phase compensation FIFO clock	Transceiver to FPGA fabric
coreclkout	Phase compensation FIFO clock	Transceiver to FPGA fabric
fixed_clk	125MHz receiver detect clock in PIPE mode	FPGA fabric to transceiver
reconfig_clk (1), (2)	Transceiver dynamic reconfiguration and offset cancellation clock	FPGA fabric to transceiver

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.

Table 1–26. Transmitter Ports in ALTGX Megafunction for Cyclone IV GX

Block	Port Name	Input/ Output	Clock Domain	Description
TX PCS	tx_datain	Input	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	Parallel data input from the FPGA fabric to the transmitter. ■ Bus width depends on channel width multiplied by number of channels per instance.
	tx_clkout	Output	Clock signal	FPGA fabric-transmitter interface clock in non-bonded modes ■ Each channel has a `tx_clkout` signal that can be used to clock data (`tx_datain`) from the FPGA fabric into the transmitter.
	tx_coreclk	Input	Clock signal	Optional write clock port for the TX phase compensation FIFO.
	tx_phase_comp_fifo_error	Output	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	TX phase compensation FIFO full or empty indicator. ■ A high level indicates FIFO is either full or empty.
	tx_ctrlenable	Input	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	8B/10B encoder control or data identifier. This signal passes through the TX Phase Compensation FIFO. ■ A high level to encode data as a /Kx.y/ control code group. ■ A low level to encode data as a /Dx.y/ data code group.
	tx_forcedisp	Input	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	8B/10B encoder forcing disparity control. This signal passes through the TX Phase Compensation FIFO. ■ A high level to force encoding to positive or negative disparity depending on the `tx_dispval` signal level. ■ A low level to allow default encoding according to the 8B/10B running disparity rules.
	tx_dispval	Input	Synchronous to `tx_clkout` (non-bonded modes) or `coreclkout` (bonded modes)	8B/10B encoder forcing disparity value. This signal passes through the TX Phase Compensation FIFO. ■ A high level to force encoding with a negative disparity code group when `tx_forcedisp` port is asserted high. ■ A low level to force encoding with a positive disparity code group when `tx_forcedisp` port is asserted high.
	tx_invpolarity	Input	Asynchronous signal. Minimum pulse width is two parallel clock cycles.	Transmitter polarity inversion control. ■ A high level to invert the polarity of every bit of the 8- or 10-bit input data to the serializer.
	tx_bitslipboundaryselect	Input	Asynchronous signal.	Control the number of bits to slip before serializer. ■ Valid values from 0 to 9
TX PMA	tx_dataout	Output	—	Transmitter serial data output signal.
	tx_forceelecidle	Input	Asynchronous signal.	Force the transmitter buffer to PIPE electrical idle signal levels. For equivalent signal defined in PIPE 2.00 specification, refer to Table 1–15 on page 1–54.

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

Term	Description
Memory Initialization File, also known as **.mif**	A file with the **.mif** extension will be generated for **.mif**-based reconfiguration mode. It can be either in Channel Reconfiguration mode or PLL Reconfiguration mode. ■ Channel Reconfiguration mode—this file contains information about the various ALTGX MegaWizard Plug-In Manager options that you set. Each word in the **.mif** is 16 bits wide. The dynamic reconfiguration controller writes information from the **.mif** into the transceiver channel. ■ PLL Reconfiguration mode—this file contains information about the various PLL parameters and settings that you use to configure the transceiver PLL to different output frequency. The **.mif** file is 144 × 1-bit size. During PLL reconfiguration mode, the PLL reconfiguration controller shifts these 144-bit serially into the transceiver PLL.
PMA controls	Represents **analog controls** (**Voltage Output Differential** [V_{OD}], **Pre-emphasis**, **DC Gain**, and **Manual Equalization**) as displayed in both the ALTGX and ALTGX_RECONFIG MegaWizard Plug-In Managers.
Transceiver channel	Refers to a transmitter channel, a receiver channel, or a duplex channel that has both PMA and PCS blocks.

Dynamic Reconfiguration Controller Architecture

The dynamic reconfiguration controller is a soft intellectual property (IP) that utilizes FPGA-fabric resources. You can use only one controller per transceiver block. You cannot use the dynamic reconfiguration controller to control multiple Cyclone IV devices or any off-chip interfaces.

Control and Status Signals for Channel Reconfiguration

The various control and status signals involved in the Channel Reconfiguration mode are as follows. Refer to "Dynamic Reconfiguration Controller Port List" on page 3–4 for the descriptions of the control and status signals.

The following are the input control signals:

- `logical_channel_address[n..0]`
- `reset_reconfig_address`
- `reconfig_reset`
- `reconfig_mode_sel[2..0]`
- `write_all`

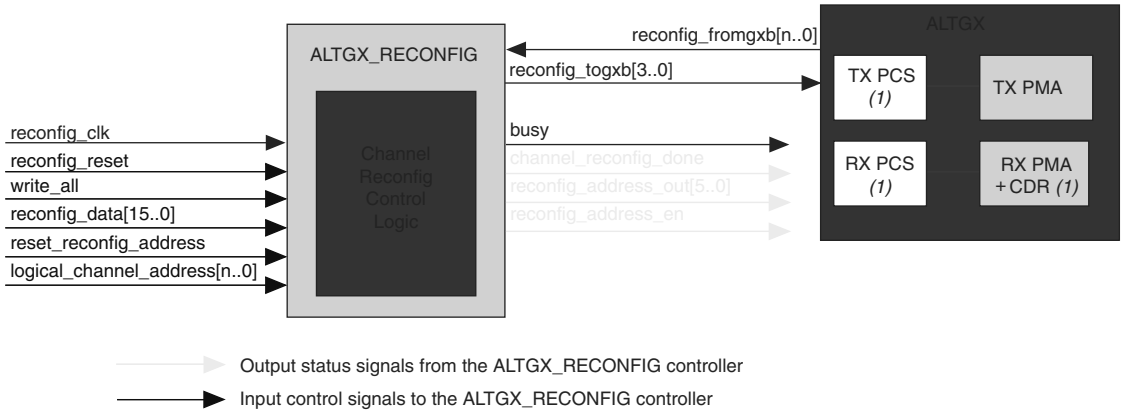
The following are output status signals:

- `reconfig_address_en`
- `reconfig_address_out[5..0]`
- `channel_reconfig_done`
- `busy`

The ALTGX_RECONFIG connection to the ALTGX instances when set in channel reconfiguration mode are as follows. For the port information, refer to "Dynamic Reconfiguration Controller Port List" on page 3–4.

Figure 3–10 shows the connection for channel reconfiguration mode.

Figure 3–10. ALTGX and ALTGX_RECONFIG Connection for Channel Reconfiguration Mode



Note to Figure 3–10:

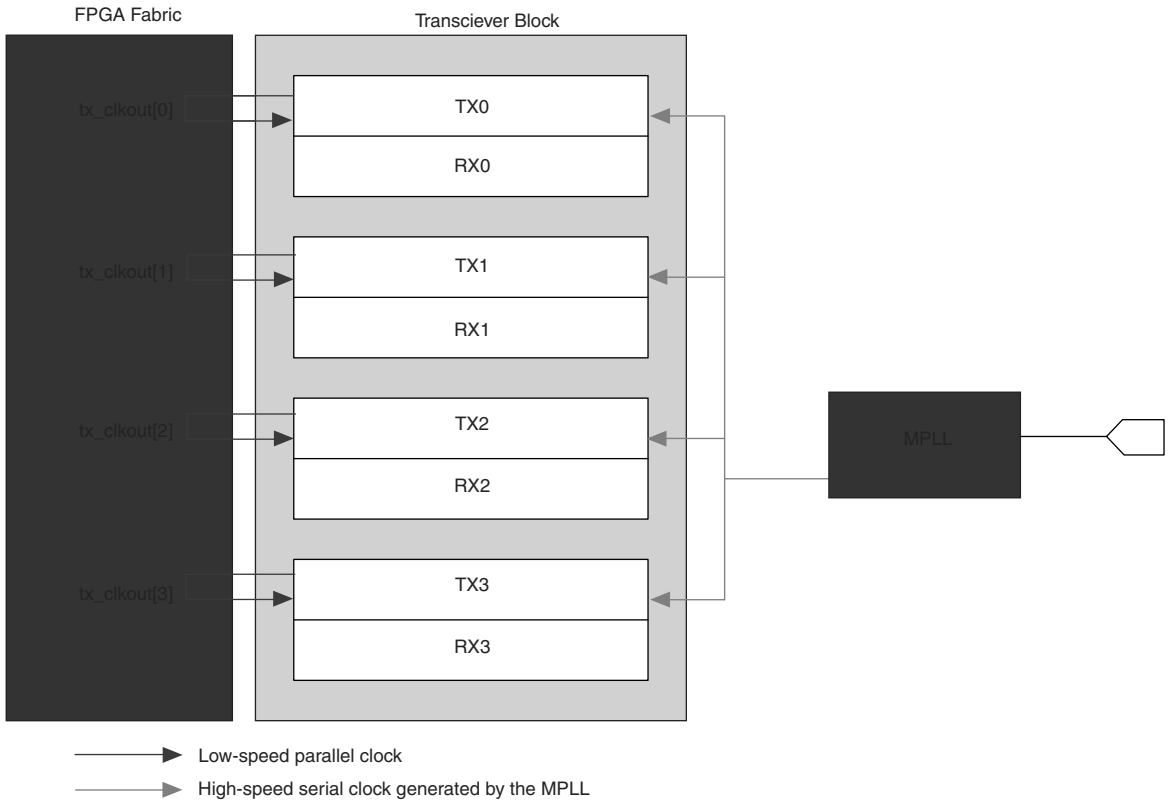
(1) This block can be reconfigured in channel reconfiguration mode.

Option 2: Use the Respective Channel Transmitter Core Clocks

- Enable this option if you want the individual transmitter channel `tx_clkout` signals to provide the write clock to their respective Transmit Phase Compensation FIFOs.
- This option is typically enabled when each transceiver channel is reconfigured to a different functional mode using channel reconfiguration.

Figure 3–12 shows how each transmitter channel's `tx_clkout` signal provides a clock to the Transmit Phase Compensation FIFOs of the respective transceiver channels.

Figure 3–12. Option 2 for Transmitter Core Clocking (Channel Reconfiguration Mode)



Receiver core clocking refers to the clock that is used to read the parallel data from the Receiver Phase Compensation FIFO into the FPGA fabric. You can use one of the following clocks to read from the Receive Phase Compensation FIFO:

- `rx_coreclk`—you can use a clock of the same frequency as `rx_clkout` from the FPGA fabric to provide the read clock to the Receive Phase Compensation FIFO. If you use `rx_coreclk`, it overrides the `rx_clkout` options in the ALTGX MegaWizard Plug-In Manager.
- `rx_clkout`—the Quartus II software automatically routes `rx_clkout` to the FPGA fabric and back into the Receive Phase Compensation FIFO.

Table 1–4. Recommended Operating Conditions for Cyclone IV GX Devices (Part 2 of 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CCA_GXB}	Transceiver PMA and auxiliary power supply	—	2.375	2.5	2.625	V
V_{CCL_GXB}	Transceiver PMA and auxiliary power supply	—	1.16	1.2	1.24	V
V_I	DC input voltage	—	–0.5	—	3.6	V
V_O	DC output voltage	—	0	—	V_{CCIO}	V
T_J	Operating junction temperature	For commercial use	0	—	85	°C
		For industrial use	–40	—	100	°C
t_{RAMP}	Power supply ramp time	Standard power-on reset (POR) *(7)*	50 µs	—	50 ms	—
		Fast POR *(8)*	50 µs	—	3 ms	—
I_{Diode}	Magnitude of DC current across PCI-clamp diode when enabled	—	—	—	10	mA

Notes to Table 1–4:

(1) All VCCA pins must be powered to 2.5 V (even when PLLs are not used) and must be powered up and powered down at the same time.

(2) You must connect V_{CCD_PLL} to V_{CCINT} through a decoupling capacitor and ferrite bead.

(3) Power supplies must rise monotonically.

(4) V_{CCIO} for all I/O banks must be powered up during device operation. Configurations pins are powered up by V_{CCIO} of I/O Banks 3, 8, and 9 where I/O Banks 3 and 9 only support V_{CCIO} of 1.5, 1.8, 2.5, 3.0, and 3.3 V. For fast passive parallel (FPP) configuration mode, the V_{CCIO} level of I/O Bank 8 must be powered up to 1.5, 1.8, 2.5, 3.0, and 3.3 V.

(5) You must set V_{CC_CLKIN} to 2.5 V if you use CLKIN as a high-speed serial interface (HSSI) refclk or as a DIFFCLK input.

(6) The CLKIN pins in I/O Banks 3B and 8B can support single-ended I/O standard when the pins are used to clock left PLLs in non-transceiver applications.

(7) The POR time for Standard POR ranges between 50 and 200 ms. V_{CCINT}, V_{CCA}, and V_{CCIO} of I/O Banks 3, 8, and 9 must reach the recommended operating range within 50 ms.

(8) The POR time for Fast POR ranges between 3 and 9 ms. V_{CCINT}, V_{CCA}, and V_{CCIO} of I/O Banks 3, 8, and 9 must reach the recommended operating range within 3 ms.

ESD Performance

This section lists the electrostatic discharge (ESD) voltages using the human body model (HBM) and charged device model (CDM) for Cyclone IV devices general purpose I/Os (GPIOs) and high-speed serial interface (HSSI) I/Os. Table 1–5 lists the ESD for Cyclone IV devices GPIOs and HSSI I/Os.

Table 1–5. ESD for Cyclone IV Devices GPIOs and HSSI I/Os

Symbol	Parameter	Passing Voltage	Unit
V_{ESDHBM}	ESD voltage using the HBM (GPIOs) *(1)*	± 2000	V
	ESD using the HBM (HSSI I/Os) *(2)*	± 1000	V
V_{ESDCDM}	ESD using the CDM (GPIOs)	± 500	V
	ESD using the CDM (HSSI I/Os) *(2)*	± 250	V

Notes to Table 1–5:

(1) The passing voltage for EP4CGX15 and EP4CGX30 row I/Os is ±1000V.

(2) This value is applicable only to Cyclone IV GX devices.

Power Consumption

Use the following methods to estimate power for a design:

- the Excel-based EPE
- the Quartus® II PowerPlay power analyzer feature

The interactive Excel-based EPE is used prior to designing the device to get a magnitude estimate of the device power. The Quartus II PowerPlay power analyzer provides better quality estimates based on the specifics of the design after place-and-route is complete. The PowerPlay power analyzer can apply a combination of user-entered, simulation-derived, and estimated signal activities that, combined with detailed circuit models, can yield very accurate power estimates.

For more information about power estimation tools, refer to the *Early Power Estimator User Guide* and the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

Switching Characteristics

This section provides performance characteristics of Cyclone IV core and periphery blocks for commercial grade devices.

These characteristics can be designated as Preliminary or Final.

- Preliminary characteristics are created using simulation results, process data, and other known parameters. The upper-right hand corner of these tables show the designation as “Preliminary”.
- Final numbers are based on actual silicon characterization and testing. The numbers reflect the actual performance of the device under worst-case silicon process, voltage, and junction temperature conditions. There are no designations on finalized tables.

Table 1–23 lists the Cyclone IV GX transceiver block AC specifications.

Table 1–23. Transceiver Block AC Specification for Cyclone IV GX Devices *(1), (2)*

Symbol/ Description	Conditions	C6			C7, I7			C8			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
PCIe Transmit Jitter Generation *(3)*											
Total jitter at 2.5 Gbps (Gen1)	Compliance pattern	—	—	0.25	—	—	0.25	—	—	0.25	UI
PCIe Receiver Jitter Tolerance *(3)*											
Total jitter at 2.5 Gbps (Gen1)	Compliance pattern	> 0.6			> 0.6			> 0.6			UI
GIGE Transmit Jitter Generation *(4)*											
Deterministic jitter (peak-to-peak)	Pattern = CRPAT	—	—	0.14	—	—	0.14	—	—	0.14	UI
Total jitter (peak-to-peak)	Pattern = CRPAT	—	—	0.279	—	—	0.279	—	—	0.279	UI
GIGE Receiver Jitter Tolerance *(4)*											
Deterministic jitter tolerance (peak-to-peak)	Pattern = CJPAT	> 0.4			> 0.4			> 0.4			UI
Combined deterministic and random jitter tolerance (peak-to-peak)	Pattern = CJPAT	> 0.66			> 0.66			> 0.66			UI

Notes to Table 1–23:

(1) Dedicated `refclk` pins were used to drive the input reference clocks.
(2) The jitter numbers specified are valid for the stated conditions only.
(3) The jitter numbers for PIPE are compliant to the PCIe Base Specification 2.0.
(4) The jitter numbers for GIGE are compliant to the IEEE802.3-2002 Specification.

Core Performance Specifications

The following sections describe the clock tree specifications, PLLs, embedded multiplier, memory block, and configuration specifications for Cyclone IV Devices.

Clock Tree Specifications

Table 1–24 lists the clock tree specifications for Cyclone IV devices.

Table 1–24. Clock Tree Performance for Cyclone IV Devices ***(Part 1 of 2)***

Device	Performance								Unit
	C6	C7	C8	C8L *(1)*	C9L *(1)*	I7	I8L *(1)*	A7	
EP4CE6	500	437.5	402	362	265	437.5	362	402	MHz
EP4CE10	500	437.5	402	362	265	437.5	362	402	MHz
EP4CE15	500	437.5	402	362	265	437.5	362	402	MHz
EP4CE22	500	437.5	402	362	265	437.5	362	402	MHz
EP4CE30	500	437.5	402	362	265	437.5	362	402	MHz
EP4CE40	500	437.5	402	362	265	437.5	362	402	MHz