



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	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/ep4ce22e22c6n

© 2016 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



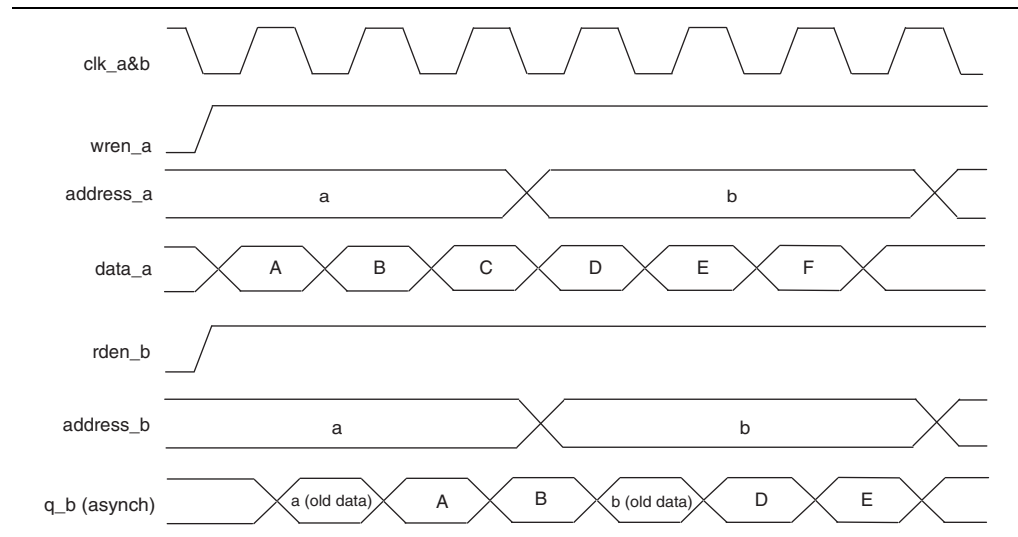
In this mode, you also have two output choices: **Old Data** mode or **Don't Care** mode. In **Old Data** mode, a read-during-write operation to different ports causes the RAM outputs to reflect the old data at that address location. In **Don't Care** mode, the same operation results in a “Don't Care” or unknown value on the RAM outputs.



For more information about how to implement the desired behavior, refer to the [RAM Megafunction User Guide](#).

Figure 3-16 shows a sample functional waveform of mixed port read-during-write behavior for **Old Data** mode. In **Don't Care** mode, the old data is replaced with “Don't Care”.

Figure 3-16. Mixed Port Read-During-Write: Old Data Mode



For mixed-port read-during-write operation with dual clocks, the relationship between the clocks determines the output behavior of the memory. If you use the same clock for the two clocks, the output is the old data from the address location. However, if you use different clocks, the output is unknown during the mixed-port read-during-write operation. This unknown value may be the old or new data at the address location, depending on whether the read happens before or after the write.

Conflict Resolution

When you are using M9K memory blocks in true dual-port mode, it is possible to attempt two write operations to the same memory location (address). Because there is no conflict resolution circuitry built into M9K memory blocks, this results in unknown data being written to that location. Therefore, you must implement conflict-resolution logic external to the M9K memory block.

Figure 6-16. RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks ⁽¹⁾

Note to Figure 6-16:

(1) R_S and R_P values are pending characterization.

A resistor network is required to attenuate the output voltage swing to meet RSDS, mini-LVDS, and PPDS specifications when using emulated transmitters. You can modify the resistor network values to reduce power or improve the noise margin.

The resistor values chosen must satisfy Equation 6-1.

Equation 6-1. Resistor Network

$$\frac{R_S \times \frac{R_P}{2}}{R_S + \frac{R_P}{2}} = 50 \, \Omega$$

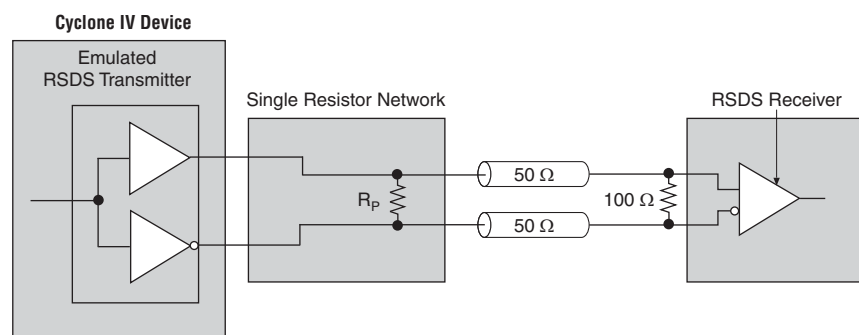


Altera recommends that you perform simulations using Cyclone IV devices IBIS models to validate that custom resistor values meet the RSDS, mini-LVDS, or PPDS requirements.

It is possible to use a single external resistor instead of using three resistors in the resistor network for an RSDS interface, as shown in Figure 6-17. The external single-resistor solution reduces the external resistor count while still achieving the required signaling level for RSDS. However, the performance of the single-resistor solution is lower than the performance with the three-resistor network.

Figure 6-17 shows the RSDS interface with a single resistor network on the top and bottom I/O banks.

Figure 6-17. RSDS Interface with Single Resistor Network on the Top and Bottom I/O Banks ⁽¹⁾




Note to Figure 6-17:

(1) R_P value is pending characterization.

LVPECL I/O Support in Cyclone IV Devices

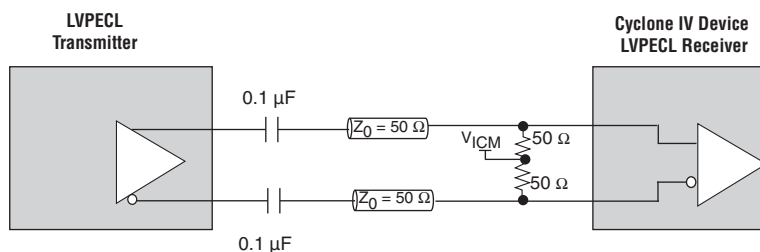
The LVPECL I/O standard is a differential interface standard that requires a 2.5-V V_{CCIO} . This standard is used in applications involving video graphics, telecommunications, data communications, and clock distribution. Cyclone IV devices support the LVPECL input standard at the dedicated clock input pins only. The LVPECL receiver requires an external 100- Ω termination resistor between the two signals at the input buffer.

 For the LVPECL I/O standard electrical specification, refer to the *Cyclone IV Device Datasheet* chapter.

AC coupling is required when the LVPECL common mode voltage of the output buffer is higher than the Cyclone IV devices LVPECL input common mode voltage.

Figure 6-18 shows the AC-coupled termination scheme. The 50- Ω resistors used at the receiver are external to the device. DC-coupled LVPECL is supported if the LVPECL output common mode voltage is in the Cyclone IV devices LVPECL input buffer specification (refer to Figure 6-19).

Figure 6-18. LVPECL AC-Coupled Termination ⁽¹⁾

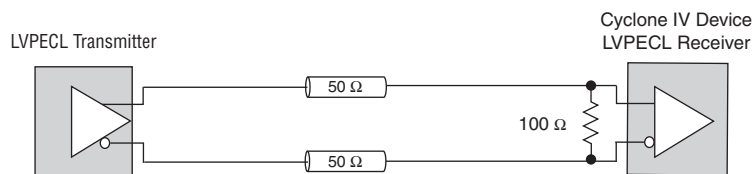


Note to Figure 6-18:

(1) The LVPECL AC-coupled termination is applicable only when an Altera FPGA transmitter is used.

Figure 6-19 shows the LVPECL DC-coupled termination.

Figure 6-19. LVPECL DC-Coupled Termination ⁽¹⁾



Note to Figure 6-19:

(1) The LVPECL DC-coupled termination is applicable only when an Altera FPGA transmitter is used.

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.



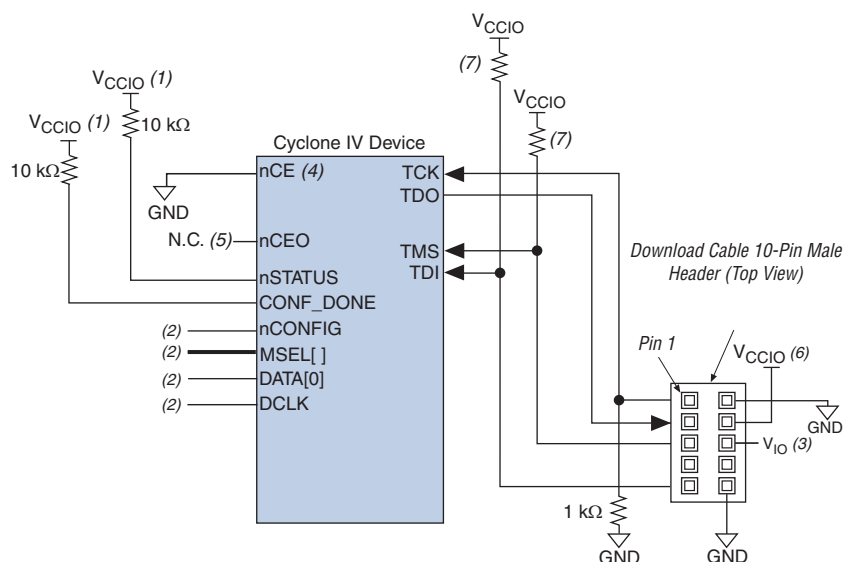
The FPP configuration is not supported in E144 package of Cyclone IV E devices.



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

Figure 8-24. JTAG Configuration of a Single Device Using a Download Cable (1.5-V or 1.8-V V_{CCIO} Powering the JTAG Pins)**Notes to Figure 8-24:**

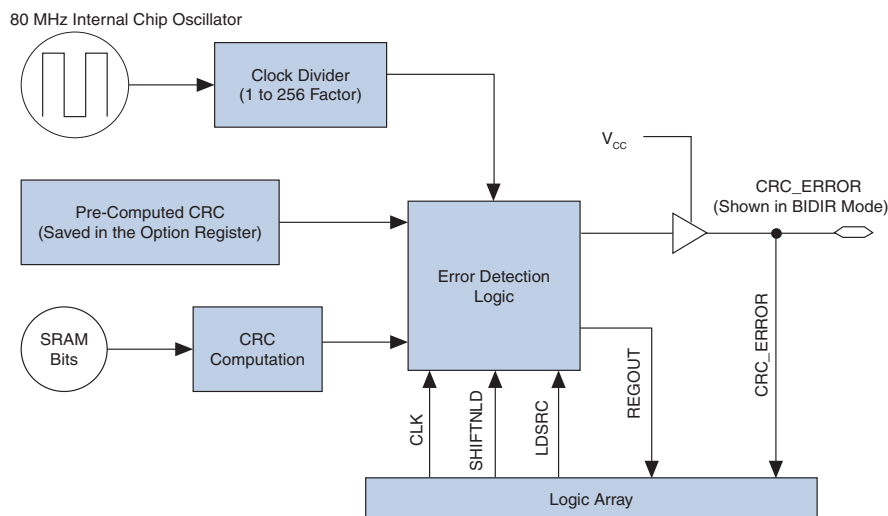
- (1) Connect these pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.
- (2) Connect the $nCONFIG$ and $MSEL$ pins to support a non-JTAG configuration scheme. If you only use JTAG configuration, connect the $nCONFIG$ pin to logic-high and the $MSEL$ pins to GND. In addition, pull $DCLK$ and $DATA[0]$ to either high or low, whichever is convenient on your board.
- (3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for AS programming; otherwise it is a no connect.
- (4) The nCE must be connected to GND or driven low for successful JTAG configuration.
- (5) 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.
- (6) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II or USB-Blaster cable with supply from V_{CCIO} . The Ethernet-Blaster, ByteBlaster II, and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide*, the *USB-Blaster Download Cable User Guide*, and the *EthernetBlaster Communications Cable User Guide*.
- (7) Resistor value can vary from 1 kΩ to 10 kΩ.

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the TDI pin to the TDO pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the TDO pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration after completion. At the end of configuration, the software checks the state of $CONF_DONE$ through the JTAG port. When Quartus II generates a **.jam** for a multi-device chain, it contains instructions so that all the devices in the chain are initialized at the same time. If $CONF_DONE$ is not high, the Quartus II software indicates that configuration has failed. If $CONF_DONE$ is high, the software indicates that configuration was successful. After the configuration bitstream is serially sent using the JTAG TDI port, the TCK port clocks an additional clock cycles to perform device initialization.

Figure 9-3 shows the error detection block diagram in FPGA devices and shows the interface that the WYSIWYG atom enables in your design.

Figure 9-3. Error Detection Block Diagram



The user logic is affected by the soft error failure, so reading out the 32-bit CRC signature through the `regout` should not be relied upon to detect a soft error. You should rely on the `CRC_ERROR` output signal itself, because this `CRC_ERROR` output signal cannot be affected by a soft error.

To enable the `cycloneiv_crcblock` WYSIWYG atom, you must name the atom for each Cyclone IV device accordingly.

Example 9-1 shows an example of how to define the input and output ports of a WYSIWYG atom in a Cyclone IV device.

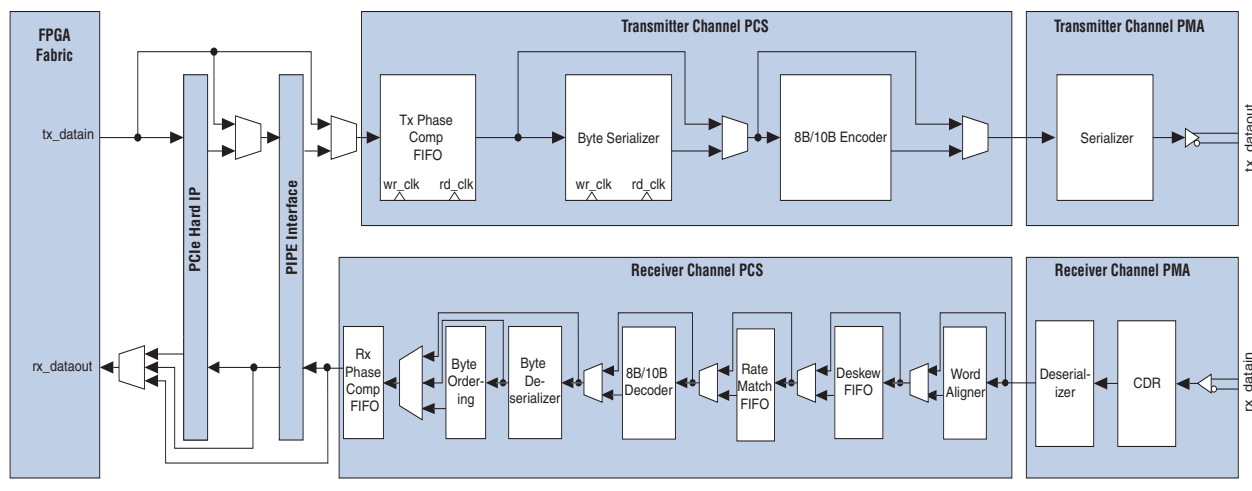
Example 9-1. Error Detection Block Diagram

```
cycloneiv_crcblock<crclblock_name>
(
  .clk(<clock source>),
  .shiftnld(<shiftnld source>),
  .ldsrc(<ldsrc source>),
  .crcerror(<crcerror out destination>),
  .regout(<output destination>),
);
```

Architectural Overview

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

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




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

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

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

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

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

Actual lock time depends on the transition density of the incoming data and the ppm difference between the receiver input reference clock and the upstream transmitter reference clock.

Transition from the LTD state to the LTR state occurs when either of the following conditions is met:

- Signal detection circuitry indicates the absence of valid signal levels at the receiver input buffer. This condition is valid for PCI Express (PIPE) mode only. CDR transitions are not dependent on signal detection circuitry in other modes.
- The recovered clock is not within the configured ppm frequency threshold setting with respect to CDR clocks from multipurpose PLLs.

In automatic lock mode, the switch from LTR to LTD states is indicated by the assertion of the `rx_freqlocked` signal and the switch from LTD to LTR states indicated by the de-assertion of the `rx_freqlocked` signal.

Manual Lock Mode

State transitions are controlled manually by using `rx_locktorefclk` and `rx_locktodata` ports. The LTR/LTD controller sets the CDR state depending on the logic level on the `rx_locktorefclk` and `rx_locktodata` ports. This mode provides the flexibility to control the CDR for a reduced lock time compared to the automatic lock mode. In automatic lock mode, the LTR/LTD controller relies on the ppm detector and the phase relationship detector to set the CDR in LTR or LTD mode. The ppm detector and phase relationship detector reaction times can be too long for some applications that require faster CDR lock time.

In manual lock mode, the `rx_freqlocked` signal is asserted when the CDR is in LTD state and de-asserted when CDR is in LTR state. For descriptions of `rx_locktorefclk` and `rx_locktodata` port controls, refer to [Table 1-27 on page 1-87](#).



If you do not enable the optional `rx_locktorefclk` and `rx_locktodata` ports, the Quartus II software automatically configures the LTR/LTD controller in automatic lock mode.



The recommended transceiver reset sequence varies depending on the CDR lock mode. For more information about the reset sequence recommendations, refer to the [Reset Control and Power Down for Cyclone IV GX Devices](#) chapter.

Deserializer

The deserializer converts received serial data from the receiver input buffer to parallel 8- or 10-bit data. Serial data is assumed to be received from the LSB to the MSB. The deserializer operates with the high-speed recovered clock from the CDR with the frequency at half of the serial data rate.

Figure 1–35 shows the datapath clocking in the transmitter and receiver operation mode with the rate match FIFO. The receiver datapath clocking in configuration without the rate match FIFO is identical to Figure 1–34.

In configuration with the rate match FIFO, the CDR unit in the receiver channel recovers the clock from received serial data and generates the high-speed recovered clock for the deserializer, and low-speed recovered clock for forwarding to the receiver PCS. The low-speed recovered clock feeds to the following blocks in the receiver PCS:

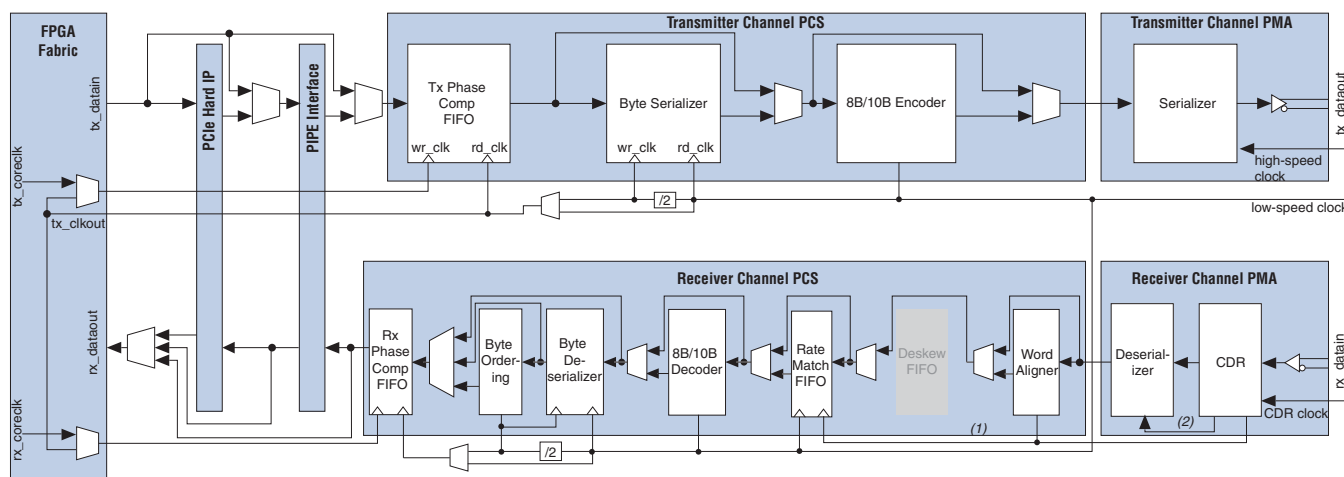
- word aligner
- write clock of rate match FIFO

The low-speed clock that is used in the transmitter PCS datapath feeds the following blocks in the 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 low-speed clock frequency is halved before feeding into the write clock of RX phase compensation FIFO. The low-speed clock is available in the FPGA fabric as tx_clkout port, which can be used in the FPGA fabric to send transmitter data and control signals, and capture receiver data and status signals.

Figure 1–35. Transmitter and Receiver Datapath Clocking with Rate Match FIFO in Non-Bonded Channel Configuration

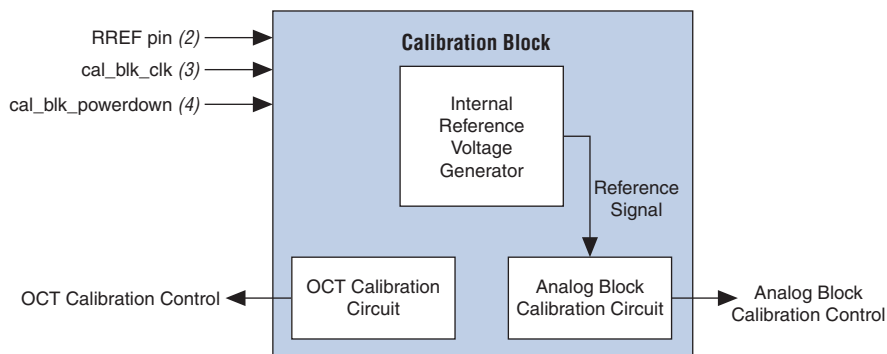


Notes to Figure 1–35:

- (1) Low-speed recovered clock.
- (2) High-speed recovered clock.

The calibration block internally generates a constant internal reference voltage, independent of PVT variations and uses this voltage and the external reference resistor on the RREF pin to generate constant reference currents. The OCT calibration circuit calibrates the OCT resistors present in the transceiver channels. Figure 1-41 shows the calibration block diagram.

Figure 1-41. Input Signals to the Calibration Blocks (1)



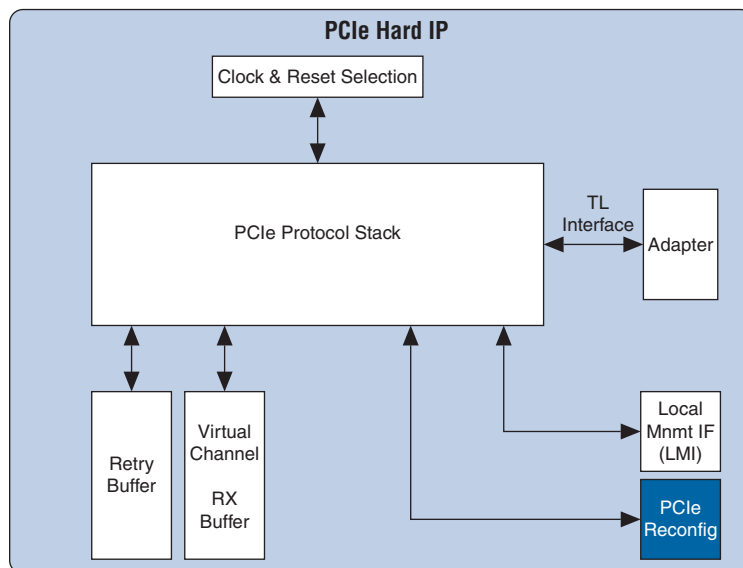
Notes to Figure 1-41:

- (1) All transceiver channels use the same calibration block clock and power down signals.
- (2) Connect a 2 k Ω (tolerance max $\pm 1\%$) external resistor to the RREF pin to ground. The RREF resistor connection in the board must be free from any external noise.
- (3) Supports up to 125 MHz clock frequency. Use either dedicated global clock or divide-down logic from the FPGA fabric to generate a slow clock on the local clock routing.
- (4) The calibration block restarts the calibration process following deassertion of the cal_blk_powerdown signal.

PCI-Express Hard IP Block

Figure 1-42 shows the block diagram of the PCIe hard IP block implementing the PHY MAC, Data Link Layer, and Transaction Layer for PCIe interfaces. The PIPE interface is used as the interface between the transceiver and the hard IP block.

Figure 1-42. PCI Express Hard IP High-Level Block Diagram



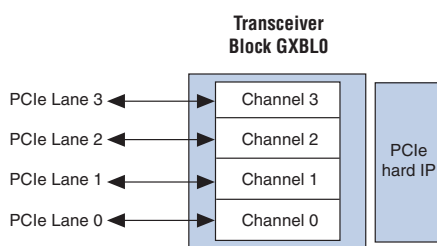
The hard IP block supports 1, 2, or 4 initial lane configurations with a maximum payload of 256 bytes at Gen1 frequency. The application interface is 64 bits with a data width of 16 bits per channel running at up to 125 MHz. As a hard macro and a verified block, it uses very few FPGA resources, while significantly reducing design risk and the time required to achieve timing closure. It is compliant with the PCI Express Base Specification 1.1. You do not have to pay a licensing fee to use this module. Configuring the hard IP block requires using the PCI Express Compiler.



For more information about the hard IP block, refer to the *PCI Express Compiler User Guide*.

Figure 1-43 shows the lane placement requirements when implementing PCIe with hard IP block.

Figure 1-43. PCIe with Hard IP Block Lane Placement Requirements ⁽¹⁾



Note to Figure 1-43:

(1) Applicable for PCIe $\times 1$, $\times 2$, and $\times 4$ implementations with hard IP blocks only.

Transceiver Functional Modes

The Cyclone IV GX transceiver supports the functional modes as listed in Table 1-14 for protocol implementation.

Table 1-14. Transceiver Functional Modes for Protocol Implementation (Part 1 of 2)

Functional Mode	Protocol	Key Feature	Reference
Basic	Proprietary, SATA, V-by-One, Display Port	Low latency PCS, transmitter in electrical idle, signal detect at receiver, wider spread asynchronous SSC	"Basic Mode" on page 1-48
PCI Express (PIPE)	PCIe Gen1 with PIPE Interface	PIPE ports, receiver detect, transmitter in electrical idle, electrical idle inference, signal detect at receiver, fast recovery, protocol-compliant word aligner and rate match FIFO, synchronous SSC	"PCI Express (PIPE) Mode" on page 1-52
GIGE	GbE	Running disparity preservation, protocol-compliant word aligner, recovered clock port for applications such as Synchronous Ethernet	"GIGE Mode" on page 1-59
Serial RapidIO	SRIO	Protocol-compliant word aligner	"Serial RapidIO Mode" on page 1-64
XAUI	XAUI	Deskew FIFO, protocol-compliant word aligner and rate match FIFO	"XAUI Mode" on page 1-67

- transmitter in electrical idle
- receiver signal detect
- receiver spread spectrum clocking

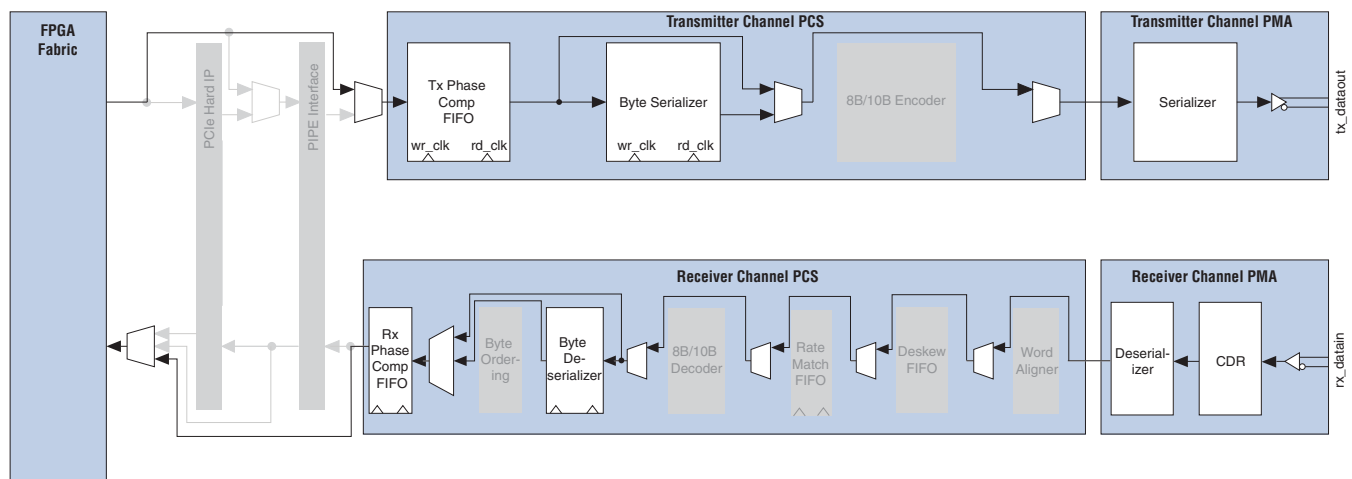
Low-Latency PCS Operation

When configured in low-latency PCS operation, the following blocks in the transceiver PCS are bypassed, resulting in a lower latency PCS datapath:

- 8B/10B encoder and decoder
- word aligner
- rate match FIFO
- byte ordering

Figure 1-47 shows the transceiver channel datapath in Basic mode with low-latency PCS operation.

Figure 1-47. Transceiver Channel Datapath in Basic Mode with Low-Latency PCS Operation



Transmitter in Electrical Idle

The transmitter buffer supports electrical idle state, where when enabled, the differential output buffer driver is tri-stated. During electrical idle, the output buffer assumes the common mode output voltage levels. For details about the electrical idle features, refer to “PCI Express (PIPE) Mode” on page 1-52.



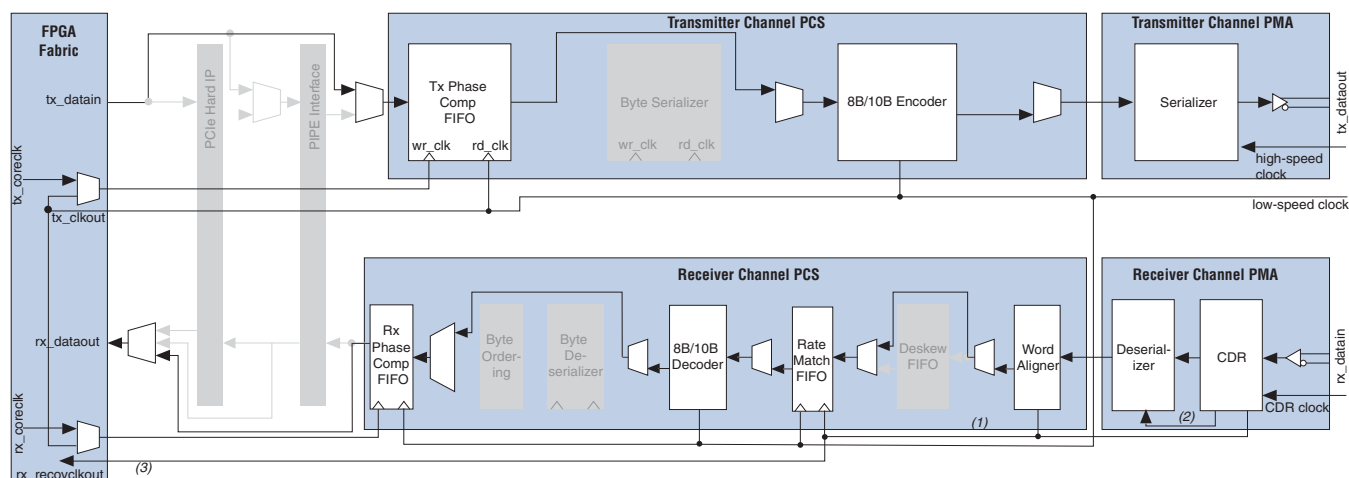
The transmitter in electrical idle feature is required for compliance to the version 2.00 of PHY Interface for the PCI Express (PIPE) Architecture specification for PCIe protocol implementation.

Signal Detect at Receiver

Signal detect at receiver is only supported when 8B/10B encoder/decoder block is enabled.

Figure 1-55 shows the transceiver channel datapath and clocking when configured in GIGE mode.

Figure 1-55. Transceiver Channel Datapath and Clocking when Configured in GIGE Mode



Notes to Figure 1-55:

- (1) Low-speed recovered clock.
- (2) High-speed recovered clock.
- (3) Optional `rx_recovclkout` port from CDR low-speed recovered clock is available for applications such as Synchronous Ethernet.

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

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

Clocking/Interface Options

The following describes the **Clocking/Interface** options available in Cyclone IV GX devices. The core clocking setup describes the transceiver core clocks that are the write and read clocks of the Transmit Phase Compensation FIFO and the Receive Phase Compensation FIFO, respectively. Core clocking is classified as transmitter core clocking and receiver core clocking.

Table 3–6 lists the supported clocking interface settings for channel reconfiguration mode in Cyclone IV GX devices.

Table 3–6. Dynamic Reconfiguration Clocking Interface Settings in Channel Reconfiguration Mode

ALTGX Setting	Description
Dynamic Reconfiguration Channel Internal and Interface Settings	
How should the receivers be clocked?	Select one of the available options: <ul style="list-style-type: none"> ■ Share a single transmitter core clock between receivers ■ Use the respective channel transmitter core clocks ■ Use the respective channel receiver core clocks
How should the transmitters be clocked?	Select one of the available options: <ul style="list-style-type: none"> ■ Share a single transmitter core clock between transmitters ■ Use the respective channel transmitter core clocks

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

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

Example 1–1 shows how to calculate the change of 50-Ω I/O impedance from 25°C at 3.0 V to 85°C at 3.15 V.

Example 1–1. Impedance Change

$$\Delta R_V = (3.15 - 3) \times 1000 \times -0.026 = -3.83$$

$$\Delta R_T = (85 - 25) \times 0.262 = 15.72$$

Because ΔR_V is negative,

$$MF_V = 1 / (3.83/100 + 1) = 0.963$$

Because ΔR_T is positive,

$$MF_T = 15.72/100 + 1 = 1.157$$

$$MF = 0.963 \times 1.157 = 1.114$$

$$R_{\text{final}} = 50 \times 1.114 = 55.71 \, \Omega$$

Pin Capacitance

Table 1–11 lists the pin capacitance for Cyclone IV devices.

Table 1–11. Pin Capacitance for Cyclone IV Devices ⁽¹⁾

Symbol	Parameter	Typical – Quad Flat Pack (QFP)	Typical – Quad Flat No Leads (QFN)	Typical – Ball-Grid Array (BGA)	Unit
C _{IOTB}	Input capacitance on top and bottom I/O pins	7	7	6	pF
C _{IOLR}	Input capacitance on right I/O pins	7	7	5	pF
C _{LVDSLR}	Input capacitance on right I/O pins with dedicated LVDS output	8	8	7	pF
C _{VREFLR} (2)	Input capacitance on right dual-purpose V _{REF} pin when used as V _{REF} or user I/O pin	21	21	21	pF
C _{VREFTB} (2)	Input capacitance on top and bottom dual-purpose V _{REF} pin when used as V _{REF} or user I/O pin	23 (3)	23	23	pF
C _{CLKTB}	Input capacitance on top and bottom dedicated clock input pins	7	7	6	pF
C _{CLKLR}	Input capacitance on right dedicated clock input pins	6	6	5	pF

Notes to Table 1–11:

- (1) The pin capacitance applies to FBGA, UBGA, and MBGA packages.
- (2) When you use the V_{REF} pin as a regular input or output, you can expect a reduced performance of toggle rate and t_{CO} because of higher pin capacitance.
- (3) C_{VREFTB} for the EP4CE22 device is 30 pF.

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.

Table 1–34. True LVDS Transmitter Timing Specifications for Cyclone IV Devices ^{(1), (3)}

Symbol	Modes	C6		C7, I7		C8, A7		C8L, I8L		C9L		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
f _{HCLK} (input clock frequency)	×10	5	420	5	370	5	320	5	320	5	250	MHz
	×8	5	420	5	370	5	320	5	320	5	250	MHz
	×7	5	420	5	370	5	320	5	320	5	250	MHz
	×4	5	420	5	370	5	320	5	320	5	250	MHz
	×2	5	420	5	370	5	320	5	320	5	250	MHz
	×1	5	420	5	402.5	5	402.5	5	362	5	265	MHz
HSIODR	×10	100	840	100	740	100	640	100	640	100	500	Mbps
	×8	80	840	80	740	80	640	80	640	80	500	Mbps
	×7	70	840	70	740	70	640	70	640	70	500	Mbps
	×4	40	840	40	740	40	640	40	640	40	500	Mbps
	×2	20	840	20	740	20	640	20	640	20	500	Mbps
	×1	10	420	10	402.5	10	402.5	10	362	10	265	Mbps
t _{DUTY}	—	45	55	45	55	45	55	45	55	45	55	%
TCCS	—	—	200	—	200	—	200	—	200	—	200	ps
Output jitter (peak to peak)	—	—	500	—	500	—	550	—	600	—	700	ps
t _{LOCK} ⁽²⁾	—	—	1	—	1	—	1	—	1	—	1	ms

Notes to Table 1–34:

- (1) Cyclone IV E—true LVDS transmitter is only supported at the output pin of Row I/O Banks 1, 2, 5, and 6.
Cyclone IV GX—true LVDS transmitter is only supported at the output pin of Row I/O Banks 5 and 6.
- (2) t_{LOCK} is the time required for the PLL to lock from the end-of-device configuration.
- (3) Cyclone IV E 1.0 V core voltage devices only support C8L, C9L, and I8L speed grades. Cyclone IV E 1.2 V core voltage devices only support C6, C7, C8, I7, and A7 speed grades. Cyclone IV GX devices only support C6, C7, C8, and I7 speed grades.

Table 1–35. Emulated LVDS Transmitter Timing Specifications for Cyclone IV Devices ^{(1), (3)} (Part 1 of 2)

Symbol	Modes	C6		C7, I7		C8, A7		C8L, I8L		C9L		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
f _{HCLK} (input clock frequency)	×10	5	320	5	320	5	275	5	275	5	250	MHz
	×8	5	320	5	320	5	275	5	275	5	250	MHz
	×7	5	320	5	320	5	275	5	275	5	250	MHz
	×4	5	320	5	320	5	275	5	275	5	250	MHz
	×2	5	320	5	320	5	275	5	275	5	250	MHz
	×1	5	402.5	5	402.5	5	402.5	5	362	5	265	MHz
HSIODR	×10	100	640	100	640	100	550	100	550	100	500	Mbps
	×8	80	640	80	640	80	550	80	550	80	500	Mbps
	×7	70	640	70	640	70	550	70	550	70	500	Mbps
	×4	40	640	40	640	40	550	40	550	40	500	Mbps
	×2	20	640	20	640	20	550	20	550	20	500	Mbps
	×1	10	402.5	10	402.5	10	402.5	10	362	10	265	Mbps