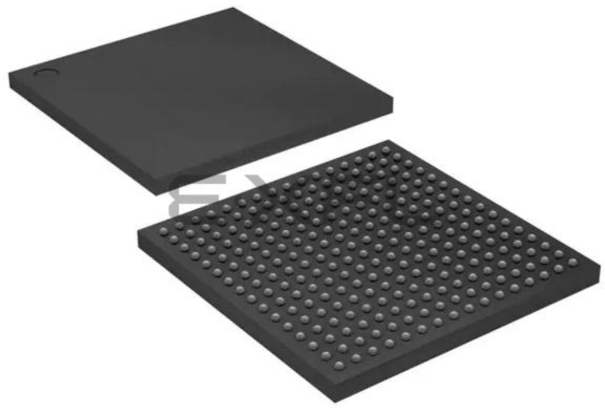


Welcome to E-XFL.COM

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.

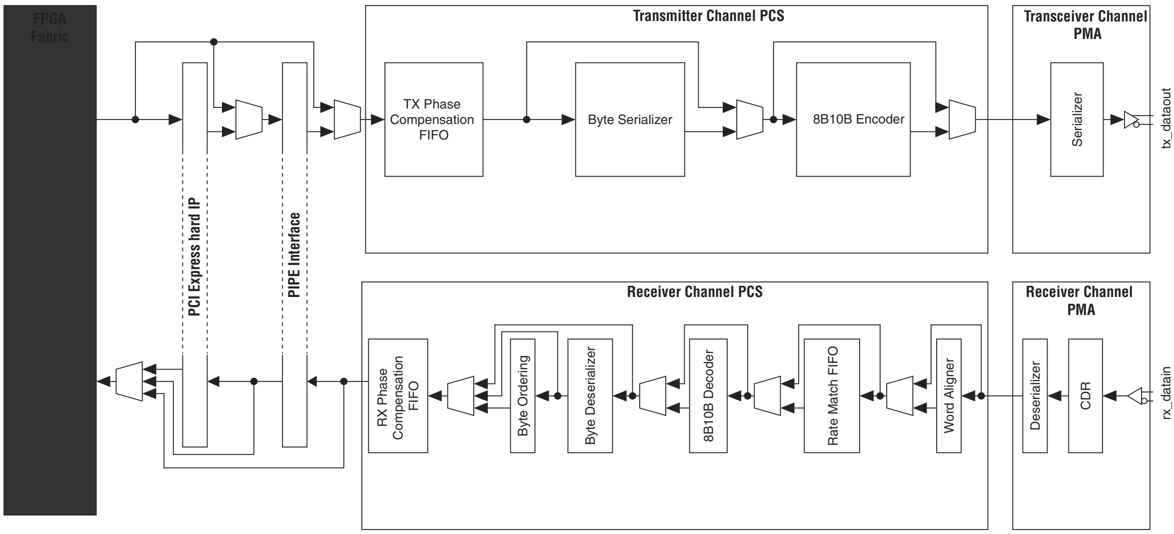
Applications of Embedded - FPGAs

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	165
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce15f17c8ln

Figure 1–1 shows the structure of the Cyclone IV GX transceiver.

Figure 1–1. Transceiver Channel for the Cyclone IV GX Device



For more information, refer to the *Cyclone IV Transceivers Architecture* chapter.

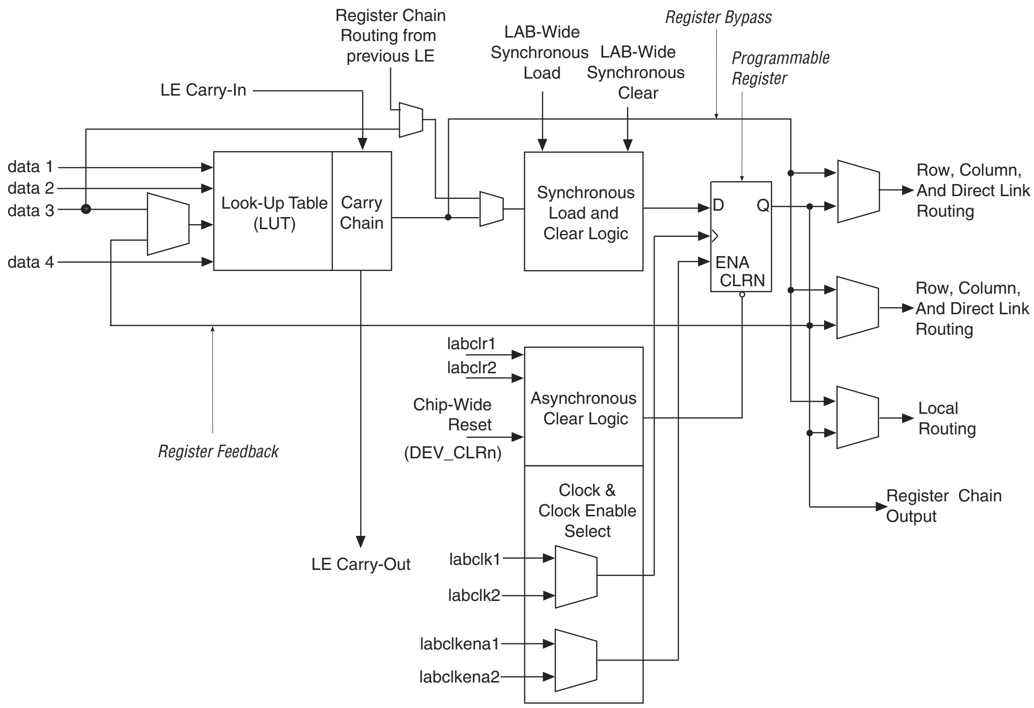
Hard IP for PCI Express (Cyclone IV GX Devices Only)

Cyclone IV GX devices incorporate a single hard IP block for ×1, ×2, or ×4 PCIe (PIPE) in each device. This hard IP block is a complete PCIe (PIPE) protocol solution that implements the PHY-MAC layer, Data Link Layer, and Transaction Layer functionality. The hard IP for the PCIe (PIPE) block supports root-port and end-point configurations. This pre-verified hard IP block reduces risk, design time, timing closure, and verification. You can configure the block with the Quartus II software's PCI Express Compiler, which guides you through the process step by step.

For more information, refer to the *PCI Express Compiler User Guide*.

Figure 2–1 shows the LEs for Cyclone IV devices.

Figure 2–1. Cyclone IV Device LEs



LE Features

You can configure the programmable register of each LE for D, T, JK, or SR flipflop operation. Each register has data, clock, clock enable, and clear inputs. Signals that use the global clock network, general-purpose I/O pins, or any internal logic can drive the clock and clear control signals of the register. Either general-purpose I/O pins or the internal logic can drive the clock enable. For combinational functions, the LUT output bypasses the register and drives directly to the LE outputs.

Each LE has three outputs that drive the local, row, and column routing resources. The LUT or register output independently drives these three outputs. Two LE outputs drive the column or row and direct link routing connections, while one LE drives the local interconnect resources. This allows the LUT to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the LUT for unrelated functions. The LAB-wide synchronous load control signal is not available when using register packing. For more information about the synchronous load control signal, refer to “LAB Control Signals” on page 2–6.

The register feedback mode allows the register output to feed back into the LUT of the same LE to ensure that the register is packed with its own fan-out LUT, providing another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

Same-Port Read-During-Write Mode

This mode applies to a single-port RAM or the same port of a true dual-port RAM. In the same port read-during-write mode, there are two output choices: **New Data** mode (or flow-through) and **Old Data** mode. In **New Data** mode, new data is available on the rising edge of the same clock cycle on which it was written. In **Old Data** mode, the RAM outputs reflect the old data at that address before the write operation proceeds.

When using **New Data** mode together with `byteena`, you can control the output of the RAM. When `byteena` is high, the data written into the memory passes to the output (flow-through). When `byteena` is low, the masked-off data is not written into the memory and the old data in the memory appears on the outputs. Therefore, the output can be a combination of new and old data determined by `byteena`.

Figure 3–14 and Figure 3–15 show sample functional waveforms of same port read-during-write behavior with both **New Data** and **Old Data** modes, respectively.

Figure 3–14. Same Port Read-During Write: New Data Mode

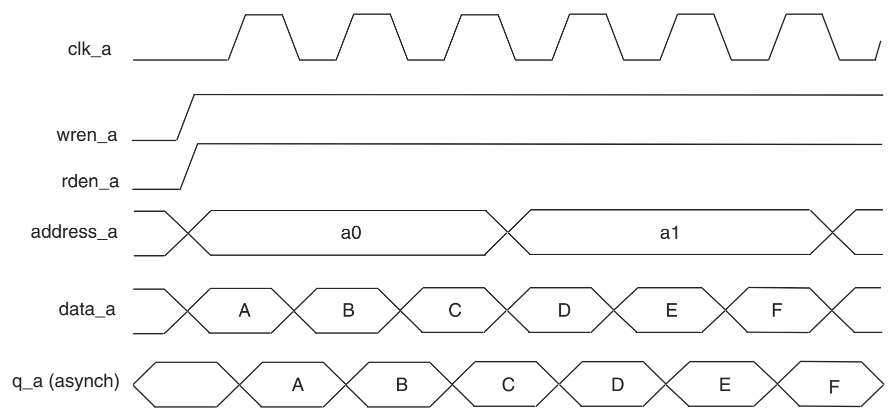
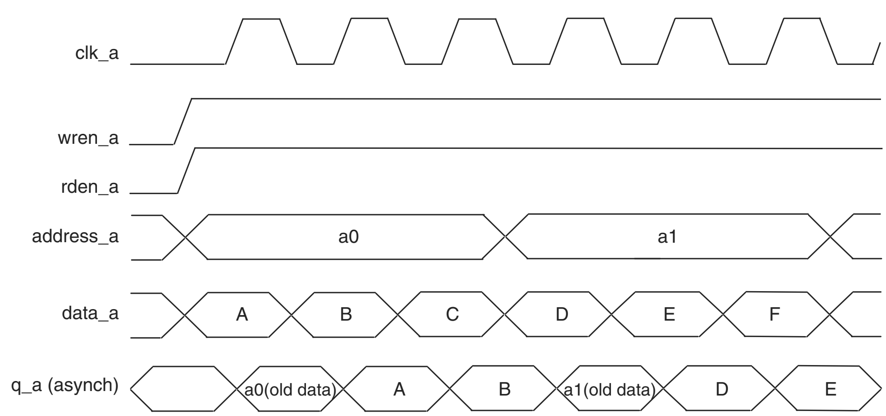


Figure 3–15. Same Port Read-During-Write: Old Data Mode



Mixed-Port Read-During-Write Mode

This mode applies to a RAM in simple or true dual-port mode, which has one port reading and the other port writing to the same address location with the same clock.

Power-Up Conditions and Memory Initialization

The M9K memory block outputs of Cyclone IV devices power up to zero (cleared) regardless of whether the output registers are used or bypassed. All M9K memory blocks support initialization using a **.mif**. You can create **.mif**s in the Quartus II software and specify their use using the RAM MegaWizard Plug-In Manager when instantiating memory in your design. Even if memory is pre-initialized (for example, using a **.mif**), it still powers up with its outputs cleared. Only the subsequent read after power up outputs the pre-initialized values.

For more information about **.mif**s, refer to the *RAM Megafunction User Guide* and the *Quartus II Handbook*.

Power Management

The M9K memory block clock enables of Cyclone IV devices allow you to control clocking of each M9K memory block to reduce AC power consumption. Use the `rden` signal to ensure that read operations only occur when necessary. If your design does not require read-during-write, reduce power consumption by deasserting the `rden` signal during write operations or any period when there are no memory operations. The Quartus II software automatically powers down any unused M9K memory blocks to save static power.

Document Revision History

Table 3–6 shows the revision history for this chapter.

Table 3–6. Document Revision History

Date	Version	Changes
November 2011	1.1	Updated the “Byte Enable Support” section.
November 2009	1.0	Initial release.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In PLLs of Cyclone IV devices, you can reconfigure both counter settings and phase shift the PLL output clock in real time. You can also change the charge pump and loop filter components, which dynamically affects PLL bandwidth. You can use these PLL components to update the output clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

The ability to reconfigure the PLL in real time is useful in applications that might operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output clock phase dynamically. For instance, a system generating test patterns is required to generate and send patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring PLL components in real time allows you to switch between two such output frequencies in a few microseconds.

You can also use this feature to adjust clock-to-out (t_{CO}) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

PLL Reconfiguration Hardware Implementation

The following PLL components are configurable in real time:

- Pre-scale counter (N)
- Feedback counter (M)
- Post-scale output counters (`C0`–`C4`)
- Dynamically adjust the charge pump current (I_{CP}) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth

Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 2 of 2)

Device		Data Size (bits)
Cyclone IV GX	EP4CGX15	3,805,568
	EP4CGX22	7,600,040
	EP4CGX30	7,600,040
		22,010,888 *(1)*
	EP4CGX50	22,010,888
	EP4CGX75	22,010,888
	EP4CGX110	39,425,016
	EP4CGX150	39,425,016

Note to Table 8–2:

(1) Only for the F484 package.

Use the data in Table 8–2 to estimate the file size before design compilation. Different configuration file formats, such as Hexadecimal (**.hex**) or Tabular Text File (**.ttf**) formats, have different file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you use compression, the file size varies after each compilation, because the compression ratio depends on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Software Settings* section in volume 2 of the *Configuration Handbook*.

Configuration and JTAG Pin I/O Requirements

Cyclone IV devices are manufactured using the TSMC 60-nm low-k dielectric process. Although Cyclone IV devices use TSMC 2.5-V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5, 3.0, and 3.3-V configuration voltage standards by following specific requirements.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. When using a serial configuration device in an AS configuration scheme, you must connect a 25-Ω series resistor for the `DATA[0]` pin. When cascading the Cyclone IV device family in a multi-device configuration for AS, AP, FPP, and PS configuration schemes, you must connect the repeater buffers between the master and slave devices for the `DATA` and `DCLK` pins. When using the JTAG configuration scheme in a multi-device configuration, connect 25-Ω resistors on both ends of the `TDO-TDI` path if the `TDO` output driver is a non-Cyclone IV device.

The output resistance of the repeater buffers and the `TDO` path for all cases must fit the maximum overshoot equation shown in Equation 8–1.

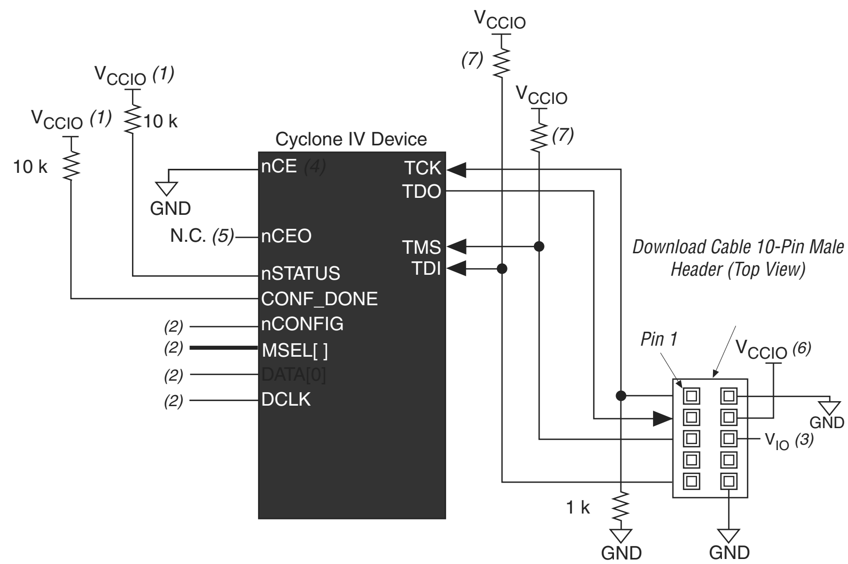
Equation 8–1. *(1)*

$$0.8Z_O \leq R_E \leq 1.8Z_O$$

Note to Equation 8–1:

(1) Z_O is the transmission line impedance and R_E is the equivalent resistance of the output buffer.

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.

When Cyclone IV devices successfully load the application configuration, they enter user mode. In user mode, the soft logic (the Nios II processor or state machine and the remote communication interface) assists the Cyclone IV device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and starts system reconfiguration.

Document Revision History

Table 10–3 lists the revision history for this chapter.

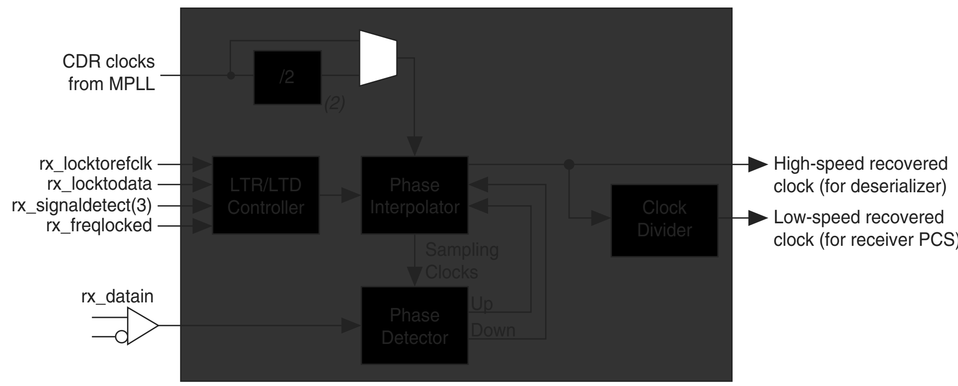
Table 10–3. Document Revision History

Date	Version	Changes
December 2013	1.3	■ Updated the "EXTEST_PULSE" section.
November 2011	1.2	■ Updated the "BST Operation Control" section. ■ Updated Table 10–2.
February 2010	1.1	■ Added Cyclone IV E devices in Table 10–1 and Table 10–2 for the Quartus II software version 9.1 SP1 release. ■ Updated Figure 10–1 and Figure 10–2. ■ Minor text edits.
November 2009	1.0	Initial release.

Clock Data Recovery

Each receiver channel has an independent CDR unit to recover the clock from the incoming serial data stream. The high-speed recovered clock is used to clock the deserializer for serial-to-parallel conversion of the received input data, and low-speed recovered clock to clock the receiver PCS blocks. Figure 1–15 illustrates the CDR unit block diagram.

Figure 1–15. CDR Unit Block Diagram



Notes to Figure 1–15:

(1) Optional RX local divider for CDR clocks from multipurpose PLL is only available in each CDR unit for EP4CGX30 (F484 package), EP4CGX50, and EP4CGX75 devices. This block is used with the transceiver dynamic reconfiguration feature. For more information, refer to the *Cyclone IV Dynamic Reconfiguration* chapter and *AN 609: Implementing Dynamic Reconfiguration in Cyclone IV GX Devices*.

(2) CDR state transition in automatic lock mode is not dependent on `rx_signaldetect` signal, except when configured in PCI Express (PIPE) mode only.

Each CDR unit gets the reference clock from one of the two multipurpose phase-locked loops (PLLs) adjacent to the transceiver block. The CDR works by tracking the incoming data with a phase detector and finding the optimum sampling clock phase from the phase interpolator unit. The CDR operations are controlled by the LTR/LTD controller block, where the CDR may operate in the following states:

- Lock-to-reference (LTR) state—phase detector disabled and CDR ignores incoming data
- Lock-to-data (LTD) state—phase detector enabled and CDR tracks incoming data to find the optimum sampling clock phase

State transitions are supported with automatic lock mode and manual lock mode.

Automatic Lock Mode

Upon receiver power-up and reset cycle, the CDR is put into LTR state. Transition to the LTD state is performed automatically when both of the following conditions are met:

- Signal detection circuitry indicates the presence 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 within the configured part per million (ppm) frequency threshold setting with respect to the CDR clocks from multipurpose PLL.

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.

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

Channel Configuration		Quartus II Selection
Bonded	With rate match FIFO (1)	`coreclkout` clock feeds the FIFO read clock for the bonded channels. `coreclkout` clock is the common bonded low-speed clock, which also feeds the FIFO read clock and transmitter PCS in the bonded channels.
	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.

Note to Table 1–13:

(1) Configuration with rate match FIFO is supported in transmitter and receiver operation.

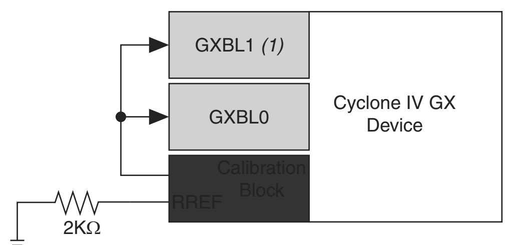
When using user-specified clock option, ensure that the clock feeding `rx_coreclk` port has 0 ppm difference with the RX phase compensation FIFO write clock.

Calibration Block

This block calibrates the OCT resistors and the analog portions of the transceiver blocks to ensure that the functionality is independent of process, voltage, and temperature (PVT) variations.

Figure 1–40 shows the location of the calibration block and how it is connected to the transceiver blocks.

Figure 1–40. Transceiver Calibration Blocks Location and Connection

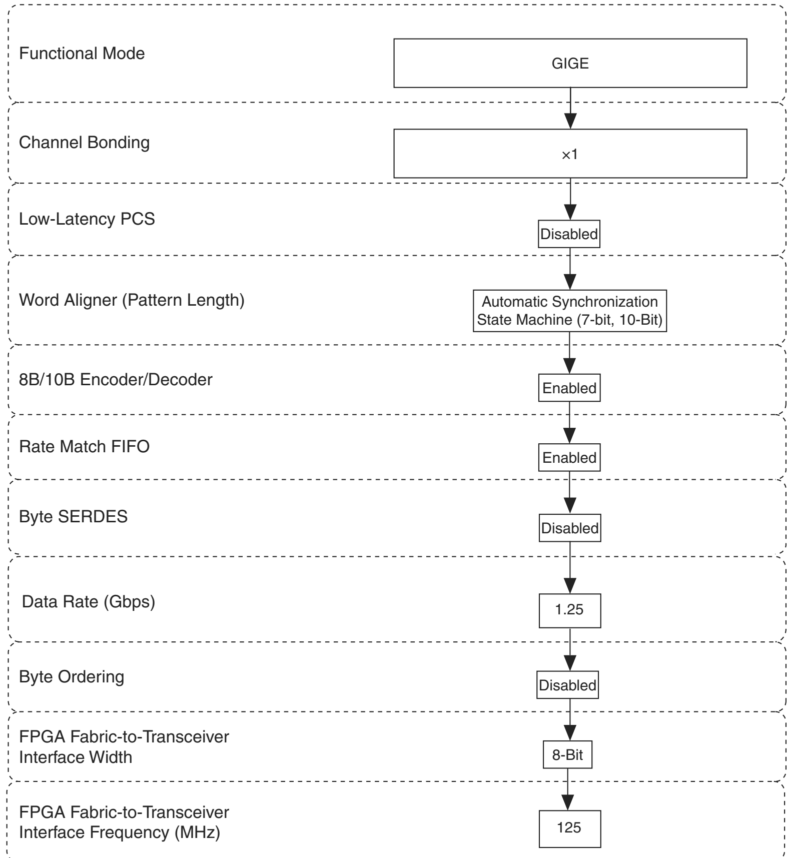


Note to Figure 1–40:

(1) Transceiver block GXBL1 is only available for devices in F484 and larger packages.

Figure 1–56 shows the transceiver configuration in GIGE mode.

Figure 1–56. Transceiver Configuration in GIGE Mode



When configured in GIGE mode, three encoded comma (/K28.5/) code groups are transmitted automatically after deassertion of `tx_digitalreset` and before transmitting user data on the `tx_datain` port. This could affect the synchronization state machine behavior at the receiver.

Depending on when you start transmitting the synchronization sequence, there could be an even or odd number of encoded data (/Dx.y/) code groups transmitted between the last of the three automatically sent /K28.5/ code groups and the first /K28.5/ code group of the synchronization sequence. If there is an even number of /Dx.y/ code groups received between these two /K28.5/ code groups, the first /K28.5/ code group of the synchronization sequence begins at an odd code group boundary. An IEEE802.3-compliant GIGE synchronization state machine treats this as an error condition and goes into the Loss-of-Sync state.

Transceiver Top-Level Port Lists

Table 1–26 through Table 1–29 provide descriptions of the ports available when instantiating a transceiver using the ALTGX megafunction. The ALTGX megafunction requires a relatively small number of signals. There are also a large number of optional signals that facilitate debugging by providing information about the state of the transceiver.

If you are reconfiguring the multipurpose PLL with a different M counter value, follow these steps:

1. During transceiver PLL reconfiguration, assert `tx_digitalreset`, `rx_digitalreset`, and `rx_analogreset` signals.
2. Perform PLL reconfiguration to update the multipurpose PLL with the PLL **.mif** files.
3. Perform channel reconfiguration and update the transceiver with the GXB reconfiguration **.mif** files. If you have multiple channel instantiations connected to the same multipurpose PLL, reconfigure each channel.
4. Deassert `tx_digitalreset` and `rx_analogreset` signals.
5. After the `rx_freqlocked` signal goes high, wait for at least 4 µs, and then deassert the `rx_digitalreset` signal.

Error Indication During Dynamic Reconfiguration

The ALTGX_RECONFIG MegaWizard Plug-In Manager provides an error status signal when you select the **Enable illegal mode checking** option or the **Enable self recovery** option in the **Error checks/data rate switch** screen. The conditions under which the error signal is asserted are:

- **Enable illegal mode checking option**—when you select this option, the dynamic reconfiguration controller checks whether an attempted operation falls under one of the conditions listed below. The dynamic reconfiguration controller detects these conditions within two `reconfig_clk` cycles, deasserts the `busy` signal, and asserts the error signal for two `reconfig_clk` cycles.
  - PMA controls, read operation—none of the output ports (`rx_eqctrl_out`, `rx_eqdcgain_out`, `tx_vodctrl_out`, and `tx_preemp_out`) are selected in the ALTGX_RECONFIG instance and the `read` signal is asserted.
  - PMA controls, write operation—none of the input ports (`rx_eqctrl`, `rx_eqdcgain`, `tx_vodctrl`, and `tx_preemp`) are selected in the ALTGX_RECONFIG instance and the `write_all` signal is asserted.
- **Channel reconfiguration** and **PMA reconfiguration mode** select - read operation option:
  - The `reconfig_mode_sel` input port is set to **3'b001** (Channel reconfiguration mode)
  - The `read` signal is asserted
- **Enable self recovery** option—when you select this option, the ALTGX_RECONFIG MegaWizard Plug-In Manager provides the error output port. The dynamic reconfiguration controller quits an operation if it did not complete within the expected number of clock cycles. After recovering from the illegal operation, the dynamic reconfiguration controller deasserts the `busy` signal and asserts the error output port for two `reconfig_clk` cycles.

☞ The error signal is not asserted when an illegal value is written to any of the PMA controls.

Cyclone IV Device Handbook,

Volume 3

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

CYIV-5V3-2.1

Contents

Chapter Revision Dates v

Additional Information
How to Contact Altera Info–1
Typographic Conventions Info–1

Section I. Device Datasheet

Chapter 1. Cyclone IV Device Datasheet
Operating Conditions 1–1
Absolute Maximum Ratings 1–2
Maximum Allowed Overshoot or Undershoot Voltage 1–2
Recommended Operating Conditions 1–4
ESD Performance 1–6
DC Characteristics 1–7
Supply Current 1–7
Bus Hold 1–7
OCT Specifications 1–8
Pin Capacitance 1–10
Internal Weak Pull-Up and Weak Pull-Down Resistor 1–11
Hot-Socketing 1–11
Schmitt Trigger Input 1–12
I/O Standard Specifications 1–12
Power Consumption 1–16
Switching Characteristics 1–16
Transceiver Performance Specifications 1–17
Core Performance Specifications 1–23
Clock Tree Specifications 1–23
PLL Specifications 1–24
Embedded Multiplier Specifications 1–26
Memory Block Specifications 1–26
Configuration and JTAG Specifications 1–26
Periphery Performance 1–27
High-Speed I/O Specifications 1–28
External Memory Interface Specifications 1–32
Duty Cycle Distortion Specifications 1–33
OCT Calibration Timing Specification 1–33
IOE Programmable Delay 1–34
I/O Timing 1–37
Glossary 1–37
Document Revision History 1–42

The OCT resistance may vary with the variation of temperature and voltage after calibration at device power-up. Use Table 1–10 and Equation 1–1 to determine the final OCT resistance considering the variations after calibration at device power-up. Table 1–10 lists the change percentage of the OCT resistance with voltage and temperature.

Table 1–10. OCT Variation After Calibration at Device Power-Up for Cyclone IV Devices *(1)*

Nominal Voltage	dR/dT (%/°C)	dR/dV (%/mV)
3.0	0.262	–0.026
2.5	0.234	–0.039
1.8	0.219	–0.086
1.5	0.199	–0.136
1.2	0.161	–0.288

Note to Table 1–10:

(1) This specification is not applicable to EP4CGX15, EP4CGX22, and EP4CGX30 devices.

Equation 1–1. Final OCT Resistance *(1), (2), (3), (4), (5), (6)*

$\Delta R_V = (V_2 - V_1) \times 1000 \times dR/dV$ ——— *(7)*

$\Delta R_T = (T_2 - T_1) \times dR/dT$ ——— *(8)*

For $\Delta R_x < 0$; $MF_x = 1/ (|\Delta R_x|/100 + 1)$ ——— *(9)*

For $\Delta R_x > 0$; $MF_x = \Delta R_x/100 + 1$ ——— *(10)*

$MF = MF_V \times MF_T$ ——— *(11)*

$R_{final} = R_{initial} \times MF$ ——— *(12)*

Notes to Equation 1–1:

(1) T_2 is the final temperature.
(2) T_1 is the initial temperature.
(3) MF is multiplication factor.
(4) R_{final} is final resistance.
(5) $R_{initial}$ is initial resistance.
(6) Subscript $_x$ refers to both $_V$ and $_T$.
(7) ΔR_V is a variation of resistance with voltage.
(8) ΔR_T is a variation of resistance with temperature.
(9) dR/dT is the change percentage of resistance with temperature after calibration at device power-up.
(10) dR/dV is the change percentage of resistance with voltage after calibration at device power-up.
(11) V_2 is final voltage.
(12) V_1 is the initial voltage.

Table 1–46. Glossary (Part 2 of 5)

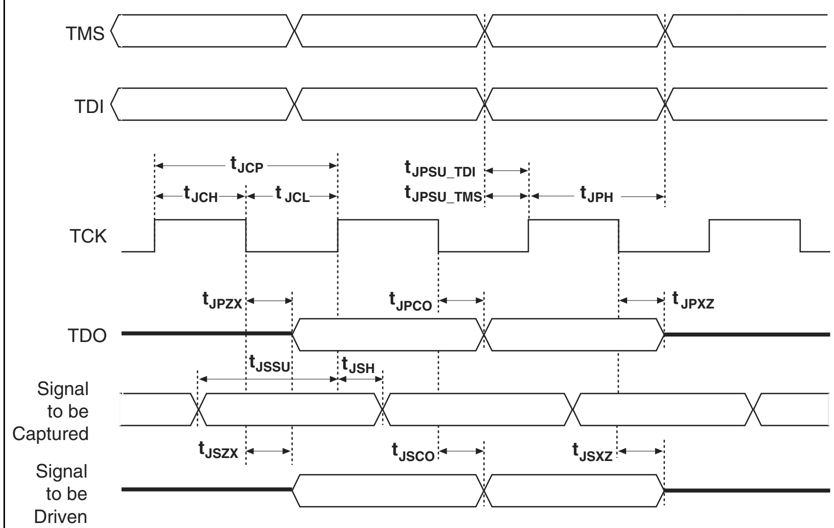
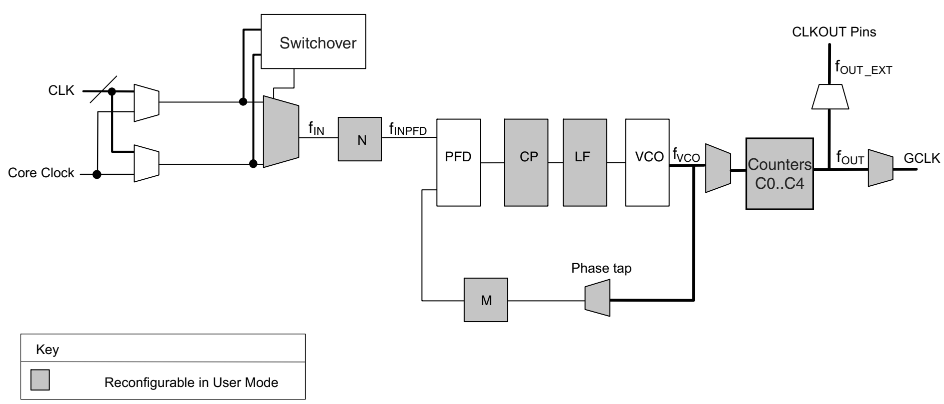
Letter	Term	Definitions
J	JTAG Waveform	
K	—	—
L	—	—
M	—	—
N	—	—
O	—	—
P	PLL Block	The following highlights the PLL specification parameters: 
Q	—	—

Table 1–47. Document Revision History

Date	Version	Changes
February 2010	1.1	■ Updated Table 1–3 through Table 1–44 to include information for Cyclone IV E devices and Cyclone IV GX devices for Quartus II software version 9.1 SP1 release. ■ Minor text edits.
November 2009	1.0	Initial release.