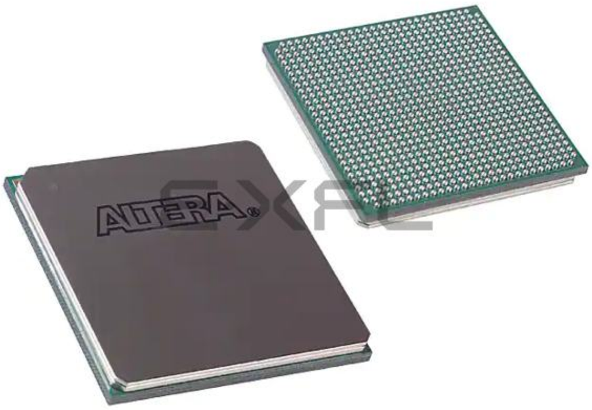


Welcome to E-XFL.COM

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.

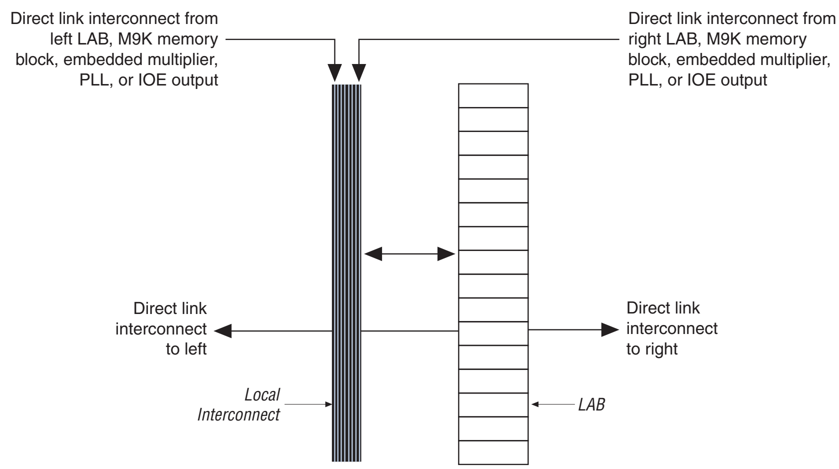
Details	
Product Status	Active
Number of LABs/CLBs	4713
Number of Logic Elements/Cells	75408
Total RAM Bits	2810880
Number of I/O	426
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	780-BGA
Supplier Device Package	780-FBGA (29x29)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce75f29c8n

LAB Interconnects

The LAB local interconnect is driven by column and row interconnects and LE outputs in the same LAB. Neighboring LABs, phase-locked loops (PLLs), M9K RAM blocks, and embedded multipliers from the left and right can also drive the local interconnect of a LAB through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LE can drive up to 48 LEs through fast local and direct link interconnects.

Figure 2–5 shows the direct link connection.

Figure 2–5. Cyclone IV Device Direct Link Connection



LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs. The control signals include:

- Two clocks
- Two clock enables
- Two asynchronous clears
- One synchronous clear
- One synchronous load

You can use up to eight control signals at a time. Register packing and synchronous load cannot be used simultaneously.

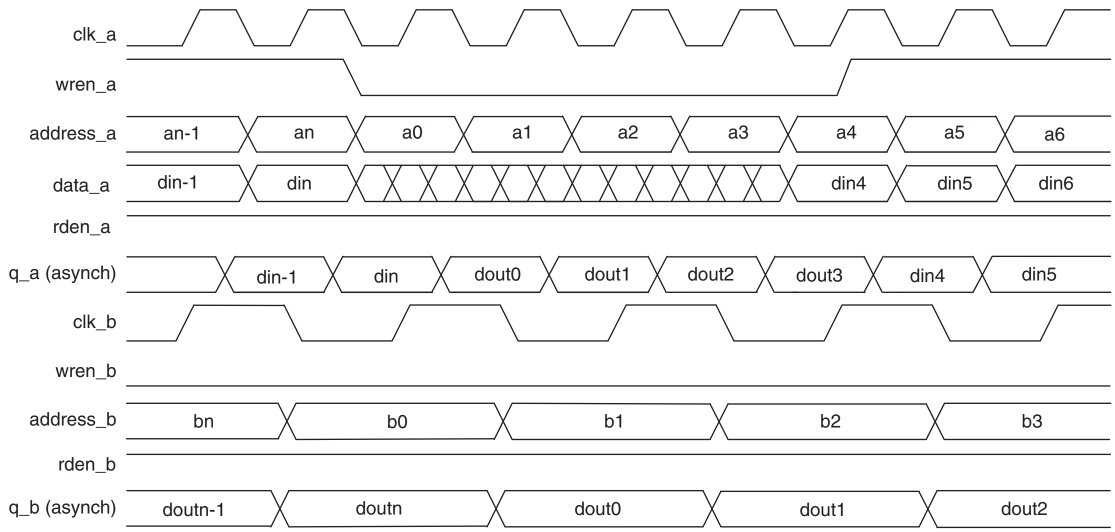
Each LAB can have up to four non-global control signals. You can use additional LAB control signals as long as they are global signals.

Synchronous clear and load signals are useful for implementing counters and other functions. The synchronous clear and synchronous load signals are LAB-wide signals that affect all registers in the LAB.

In true dual-port mode, you can access any memory location at any time from either port A or port B. However, when accessing the same memory location from both ports, you must avoid possible write conflicts. When you attempt to write to the same address location from both ports at the same time, a write conflict happens. This results in unknown data being stored to that address location. There is no conflict resolution circuitry built into the Cyclone IV devices M9K memory blocks. You must handle address conflicts external to the RAM block.

Figure 3–11 shows true dual-port timing waveforms for the write operation at port A and read operation at port B. Registering the outputs of the RAM simply delays the q outputs by one clock cycle.

Figure 3–11. Cyclone IV Devices True Dual-Port Timing Waveform



Shift Register Mode

Cyclone IV devices M9K memory blocks can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto-correlation and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift register block, which saves logic cell and routing resources.

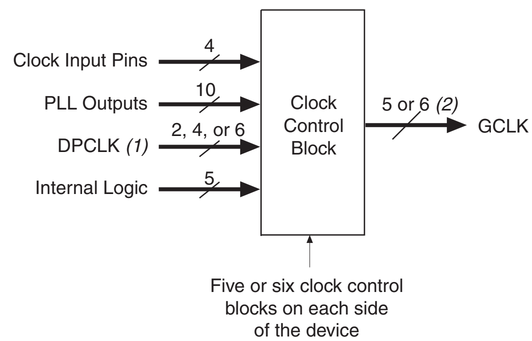
The size of a ($w \times m \times n$) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n), and must be less than or equal to the maximum number of memory bits, which is 9,216 bits. In addition, the size of ($w \times n$) must be less than or equal to the maximum width of the block, which is 36 bits. If you need a larger shift register, you can cascade the M9K memory blocks.

From the clock sources listed above, only two clock input pins, two out of four PLL clock outputs (two clock outputs from either adjacent PLLs), one DPCLK pin, and one source from internal logic can drive into any given clock control block, as shown in Figure 5–1 on page 5–11.

Out of these six inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

Figure 5–5 shows a simplified version of the clock control blocks on each side of the Cyclone IV GX device periphery.

Figure 5–5. Clock Control Blocks on Each Side of Cyclone IV GX Device



Notes to Figure 5–5:

(1) The EP4CGX15 device has two DPCLK pins; the EP4CGX22 and EP4CGX30 devices have four DPCLK pins; the EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have six DPCLK pins.

(2) Each clock control block in the EP4CGX15, EP4CGX22, and EP4CGX30 devices can drive five GCLK networks. Each clock control block in the EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices can drive six GCLK networks.

The inputs to the five clock control blocks on each side of the Cyclone IV E device must be chosen from among the following clock sources:

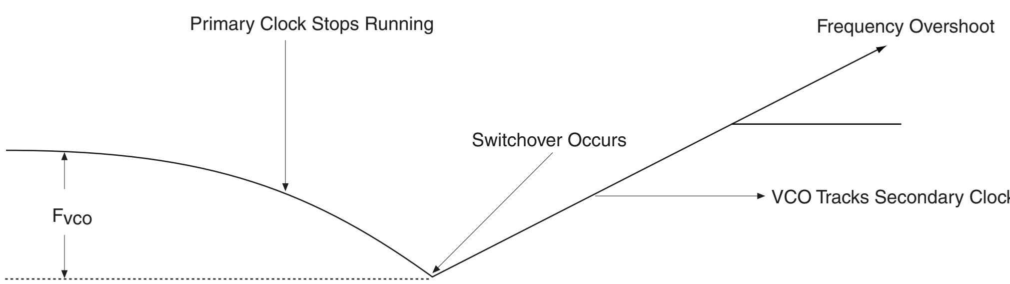
- Three or four clock input pins, depending on the specific device
- Five PLL counter outputs
- Two DPCLK pins and two CDPCLK pins from both the left and right sides and four DPCLK pins from both the top and bottom
- Five signals from internal logic

From the clock sources listed above, only two clock input pins, two PLL clock outputs, one DPCLK or CDPCLK pin, and one source from internal logic can drive into any given clock control block, as shown in Figure 5–1 on page 5–11.

Out of these six inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 20%. However, differences between the two clock sources (frequency, phase, or both) can cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between the input and output clocks.
- Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to start the manual clock switchover event. Failing to meet this requirement causes the clock switchover to malfunction.
- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. When referencing input clock changes, the low-bandwidth PLL reacts slower than a high-bandwidth PLL. When the switchover happens, the low-bandwidth PLL propagates the stopping of the clock to the output slower than the high-bandwidth PLL. The low-bandwidth PLL filters out jitter on the reference clock. However, the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock is dependent on the PLL configuration.
- If the phase relationship between the input clock to the PLL and output clock from the PLL is important in your design, assert `areset` for 10 ns after performing a clock switchover. Wait for the locked signal (or gated lock) to go high before re-enabling the output clocks from the PLL.
- Figure 5–20 shows how the VCO frequency gradually decreases when the primary clock is lost and then increases as the VCO locks on to the secondary clock. After the VCO locks on to the secondary clock, some overshoot can occur (an over-frequency condition) in the VCO frequency.

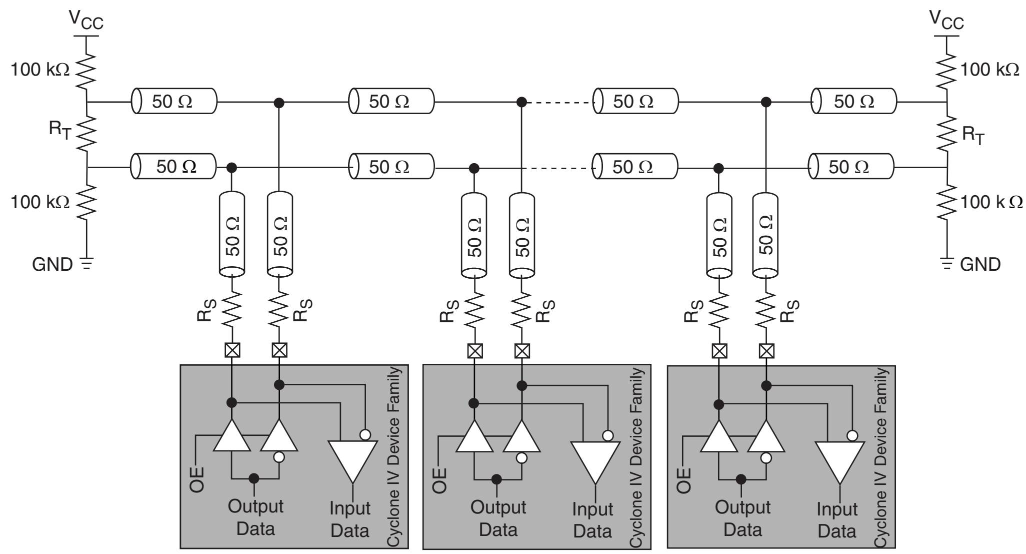
Figure 5–20. VCO Switchover Operating Frequency



- Disable the system during switchover if the system is not tolerant to frequency variations during the PLL resynchronization period. You can use the `clkbad0` and `clkbad1` status signals to turn off the PFD (`pfdena` = `0`) so the VCO maintains its last frequency. You can also use the switchover state machine to switch over to the secondary clock. Upon enabling the PFD, output clock enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. After the lock indication is stable, the system can re-enable the output clock or clocks.

Figure 6–14 shows a typical BLVDS topology with multiple transmitter and receiver pairs.

Figure 6–14. BLVDS Topology with Cyclone IV Devices Transmitters and Receivers



The BLVDS I/O standard is supported on the top, bottom, and right I/O banks of Cyclone IV devices. The BLVDS transmitter uses two single-ended output buffers with the second output buffer programmed as inverted, while the BLVDS receiver uses a true LVDS input buffer. The transmitter and receiver share the same pins. An output-enabled (`OE`) signal is required to tristate the output buffers when the LVDS input buffer receives a signal.

For more information, refer to the *Cyclone IV Device Datasheet* chapter.

Designing with BLVDS

The BLVDS bidirectional communication requires termination at both ends of the bus in BLVDS. The termination resistor (R_T) must match the bus differential impedance, which in turn depends on the loading on the bus. Increasing the load decreases the bus differential impedance. With termination at both ends of the bus, termination is not required between the two signals at the input buffer. A single series resistor (R_S) is required at the output buffer to match the output buffer impedance to the transmission line impedance. However, this series resistor affects the voltage swing at the input buffer. The maximum data rate achievable depends on many factors.

Altera recommends that you perform simulation using the IBIS model while considering factors such as bus loading, termination values, and output and input buffer location on the bus to ensure that the required performance is achieved.

For more information about BLVDS interface support in Altera devices, refer to *AN 522: Implementing Bus LVDS Interface in Supported Altera Device Families*.

Table 6–12. Document Revision History (Part 2 of 2)

Date	Version	Changes
February 2010	2.0	■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release. ■ Updated Table 6–2, Table 6–3, and Table 6–10. ■ Updated “I/O Banks” section. ■ Added Figure 6–9. ■ Updated Figure 6–10 and Figure 6–11. ■ Added Table 6–4, Table 6–6, and Table 6–8.
November 2009	1.0	Initial release.

In Cyclone IV devices, the DM pins are preassigned in the device pinouts. The Quartus II Fitter treats the `DQ` and DM pins in a `DQS` group equally for placement purposes. The preassigned `DQ` and DM pins are the preferred pins to use.

Some DDR2 SDRAM and DDR SDRAM devices support error correction coding (ECC), a method of detecting and automatically correcting errors in data transmission. In 72-bit DDR2 or DDR SDRAM, there are eight ECC pins and 64 data pins. Connect the DDR2 and DDR SDRAM ECC pins to a separate `DQS` or `DQ` group in Cyclone IV devices. The memory controller needs additional logic to encode and decode the ECC data.

Address and Control/Command Pins

The address signals and the control or command signals are typically sent at a single data rate. You can use any of the user I/O pins on all I/O banks of Cyclone IV devices to generate the address and control or command signals to the memory device.

Cyclone IV devices do not support QDR II SRAM in the burst length of two.

Memory Clock Pins

In DDR2 and DDR SDRAM memory interfaces, the memory clock signals (CK and CK#) are used to capture the address signals and the control or command signals. Similarly, QDR II SRAM devices use the write clocks (K and K#) to capture the address and command signals. The CK/CK# and K/K# signals are generated to resemble the write-data strobe using the DDIO registers in Cyclone IV devices.

CK/CK# pins must be placed on differential I/O pins (DIFFIO in Pin Planner) and in the same bank or on the same side as the data pins. You can use either side of the device for wraparound interfaces. As seen in the Pin Planner Pad View, `CK0` cannot be located in the same row and column pad group as any of the interfacing DQ pins.

For more information about memory clock pin placement, refer to *Volume 2: Device, Pin, and Board Layout Guidelines* of the *External Memory Interface Handbook.*

Cyclone IV Devices Memory Interfaces Features

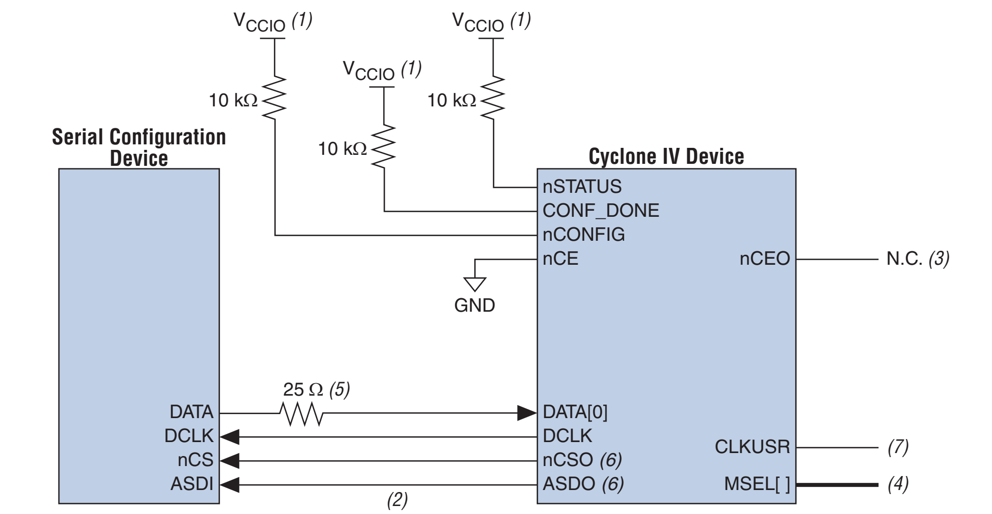
This section discusses Cyclone IV memory interfaces, including DDR input registers, DDR output registers, OCT, and phase-lock loops (PLLs).

DDR Input Registers

The DDR input registers are implemented with three internal logic element (LE) registers for every `DQ` pin. These LE registers are located in the logic array block (LAB) adjacent to the DDR input pin.

This four-pin interface connects to Cyclone IV device pins, as shown in Figure 8–2.

Figure 8–2. Single-Device AS Configuration



Notes to Figure 8–2:

(1) Connect the pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.

(2) Cyclone IV devices use the ASDO-to-ASDI path to control the configuration device.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as the DATA[1] pin in AP and FPP modes.

(7) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

To tri-state the configuration bus for AS configuration schemes, you must tie nCE high and nCONFIG low.

The 25-Ω resistor at the near end of the serial configuration device for DATA[0] works to minimize the driver impedance mismatch with the board trace and reduce the overshoot seen at the Cyclone IV device DATA[0] input pin.

In the single-device AS configuration, the maximum board loading and board trace length between the supported serial configuration device and the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The DCLK generated by the Cyclone IV device controls the entire configuration cycle and provides timing for the serial interface. Cyclone IV devices use an internal oscillator or an external clock source to generate the DCLK. For Cyclone IV E devices, you can use a 40-MHz internal oscillator to generate the DCLK and for Cyclone IV GX devices you can use a slow clock (20 MHz maximum) or a fast clock (40 MHz maximum) from the internal oscillator or an external clock from CLKUSR to generate the DCLK. There are some variations in the internal oscillator frequency because of the process, voltage, and temperature (PVT) conditions in Cyclone IV

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition, but before the factory configuration is loaded.

Table 8–26. Control Register Contents After an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting In Remote Update
`nCONFIG` reset	All bits are 0
`nSTATUS` error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
Wd time out	All bits are 0

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from indefinitely stalling the device. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Cyclone IV device.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The remote system upgrade circuitry appends 17'b1000 to form the 29-bits value for the watchdog timer. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator or `CLKUSR` (maximum frequency of 40 MHz).

Table 8–27 lists the operating range of the 10-MHz internal oscillator.

Table 8–27. 10-MHz Internal Oscillator Specifications

Minimum	Typical	Maximum	Unit
5	6.5	10	MHz

The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (`Wd`) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

To allow the remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the `RU_nRSTIMER` signal active for a minimum of 250 ns. This is equivalent to strobing the `reset_timer` input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

Errors during configuration are detected by the CRC engine. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

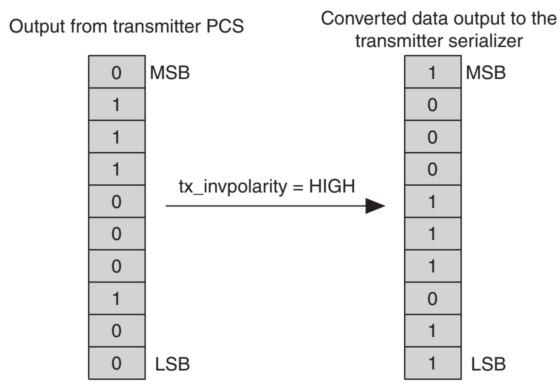
at time $n + 2$ is encoded as a positive disparity code group. In the same example, the current running disparity at time $n + 5$ indicates that the K28.5 in time $n + 6$ should be encoded with a positive disparity. Because `tx_forcedisp` is high at time $n + 6$, and `tx_dispval` is high, the K28.5 at time $n + 6$ is encoded as a negative disparity code group.

Miscellaneous Transmitter PCS Features

The transmitter PCS supports the following additional features:

- Polarity inversion—corrects accidentally swapped positive and negative signals from the serial differential link during board layout by inverting the polarity of each bit. An optional `tx_invpolarity` port is available to dynamically invert the polarity of every bit of the 8-bit or 10-bit input data to the serializer in the transmitter datapath. Figure 1–9 shows the transmitter polarity inversion feature.

Figure 1–9. Transmitter Polarity Inversion



☞ `tx_invpolarity` is a dynamic signal and might cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

Table 1–4 lists the synchronization state machine parameters for the word aligner in this mode.

Table 1–4. Synchronization State Machine Parameters

Parameter	Allowed Values
Number of erroneous code groups received to lose synchronization	1–64
Number of continuous good code groups received to reduce the error count by one	1–256

After deassertion of the `rx_digitalreset` signal in automatic synchronization state machine mode, the word aligner starts looking for the synchronization code groups, word alignment pattern or its complement in the received data stream. When the programmed number of valid synchronization code groups or ordered sets are received, the `rx_syncstatus` signal is driven high to indicate that synchronization is acquired. The `rx_syncstatus` signal is constantly driven high until the programmed number of erroneous code groups are received without receiving intermediate good groups; after which the `rx_syncstatus` signal is driven low. The word aligner indicates loss of synchronization (`rx_syncstatus` signal remains low) until the programmed number of valid synchronization code groups are received again.

In addition to restoring word boundaries, the word aligner supports the following features:

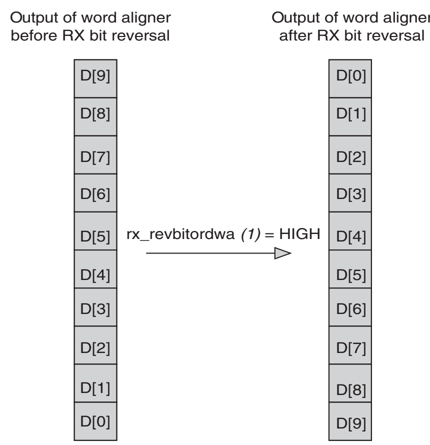
- Programmable run length violation detection—detects consecutive 1s or 0s in the data stream, and asserts run length violation signal (`rx_rlv`) when a preset run length threshold (maximum number of consecutive 1s or 0s) is detected. The `rx_rlv` signal in each channel is clocked by its parallel recovered clock and is asserted for a minimum of two recovered clock cycles to ensure that the FPGA fabric clock can latch the `rx_rlv` signal reliably because the FPGA fabric clock might have phase differences, ppm differences (in asynchronous systems), or both, with the recovered clock. Table 1–5 lists the run length violation circuit detection capabilities.

Table 1–5. Run Length Violation Circuit Detection Capabilities

Supported Data Width	Detector Range		Increment Step Settings
	Minimum	Maximum	
8-bit	4	128	4
10-bit	5	160	5

synchronization state machine mode. In bit-slip mode, you can dynamically enable the receiver bit reversal using the `rx_revbitorderwa` port. When enabled, the 8-bit or 10-bit data `D[7..0]` or `D[9..0]` at the output of the word aligner is rewired to `D[0..7]` or `D[0..9]` respectively. Figure 1–20 shows the receiver bit reversal feature.

Figure 1–20. Receiver Bit Reversal (1)



Note to Figure 1–20:

(1) The `rx_revbitordwa` port is dynamic and is only available when the word aligner is configured in bit-slip mode.

☞ When using the receiver bit reversal feature to receive MSB-to-LSB transmission, reversal of the word alignment pattern is required.

- Receiver bit-slip indicator—provides the number of bits slipped in the word aligner for synchronization with `rx_bitslipboundaryselectout` signal. For usage details, refer to “Receive Bit-Slip Indication” on page 1–76.

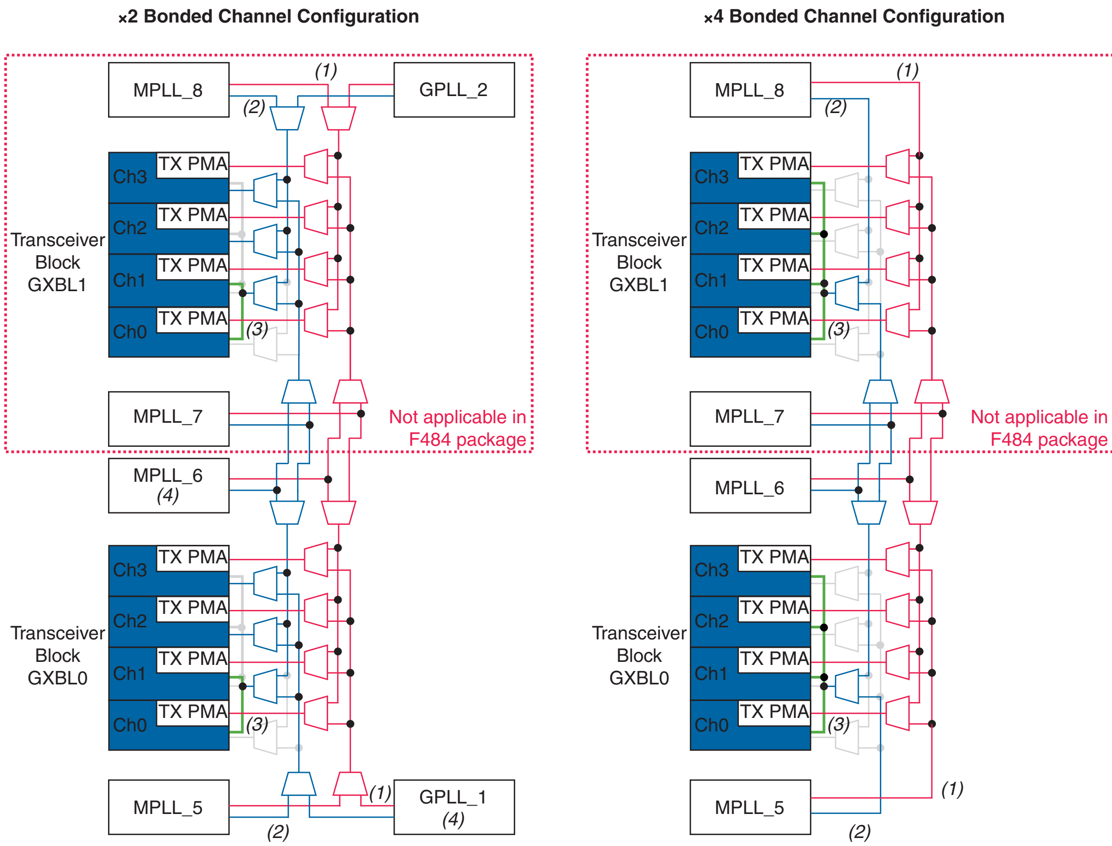
Deskew FIFO

This module is only available when used for the XAUI protocol and is used to align all four channels to meet the maximum skew requirement of 40 UI (12.8 ns) as seen at the receiver of the four lanes. The deskew operation is compliant to the PCS deskew state machine diagram specified in clause 48 of the IEEE P802.3ae specification.

The deskew circuitry consists of a 16-word deep deskew FIFO in each of the four channels, and control logics in the central control unit of the transceiver block that controls the deskew FIFO write and read operations in each channel.

For details about the deskew FIFO operations for channel deskewing, refer to “XAUI Mode” on page 1–67.

Figure 1–37. Clock Distribution in Bonded (×2 and ×4) Channel Configuration for Transceivers in F484 and Larger Packages



Notes to Figure 1–37:

(1) High-speed clock.
(2) Low-speed clock.
(3) Bonded common low-speed clock path.
(4) These PLLs have restricted clock driving capability and may not reach all connected channels. For details, refer to Table 1–10.

The channel datapath clocking is similar between bonded channels in ×2 and ×4 configurations.

Figure 1–38 shows the datapath clocking in Transmitter Only operation for ×2 and ×4 bonded configurations. In these configurations, each bonded channel selects the high-speed clock from one the supported PLLs. The high-speed clock in each bonded channel feeds the respective serializer for parallel to serial operation. The common bonded low-speed clock feeds to each bonded channel that is used for the following blocks in each transmitter PCS channel:

- 8B/10B encoder
- read clock of byte serializer
- read clock of TX phase compensation FIFO

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

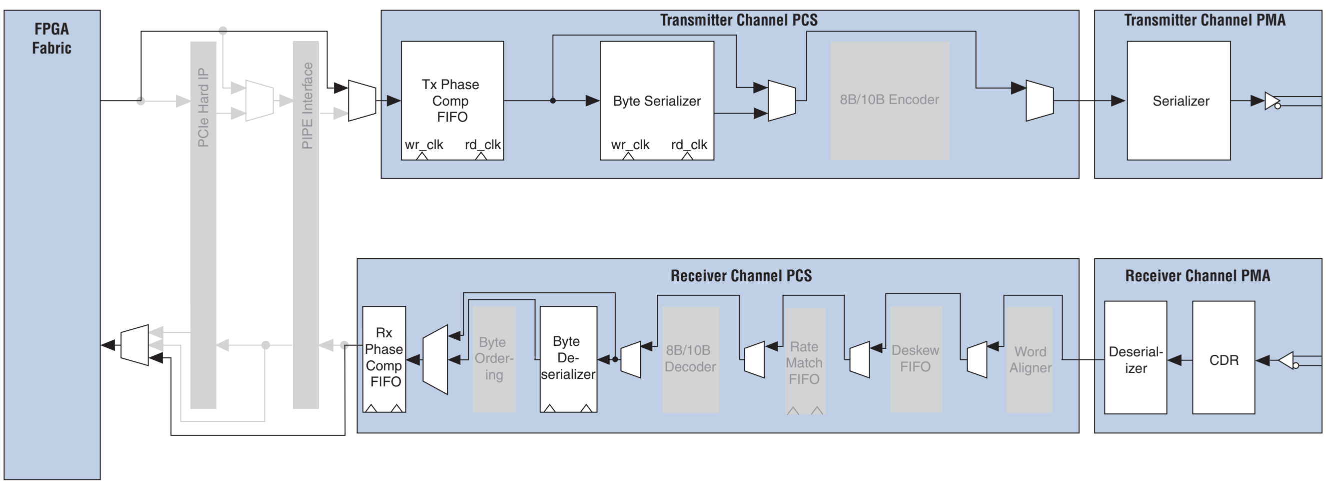
Low-Latency PCS Operation

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.

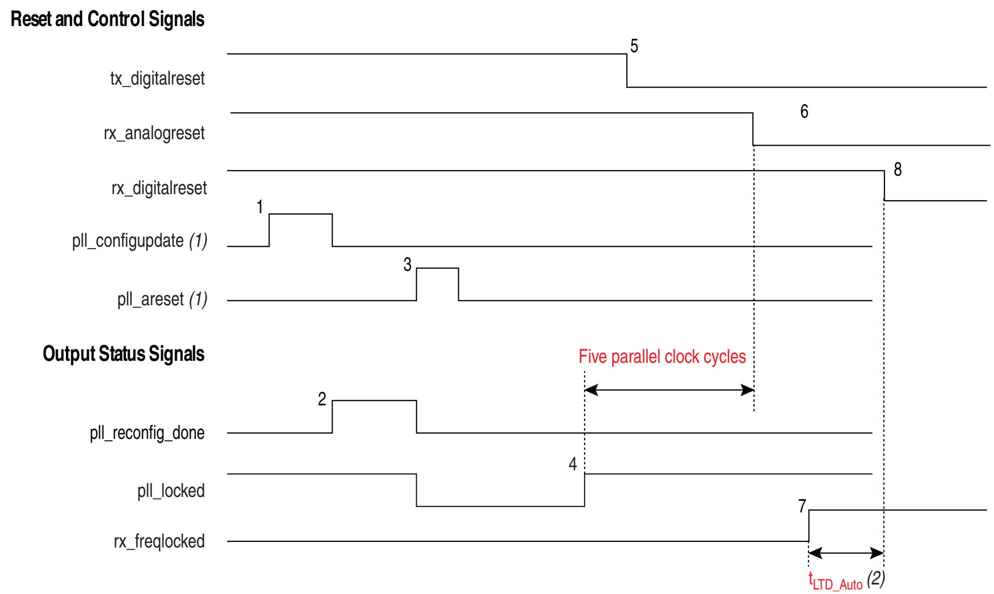
Dynamic Reconfiguration Reset Sequences

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

Reset Sequence in PLL Reconfiguration Mode

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

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



Notes to Figure 2–11:

(1) The `pll_configupdate` and `pll_areset` signals are driven by the ALTPLL_RECONFIG megafunction. For more information, refer to *AN 609: Implementing Dynamic Reconfiguration in Cyclone IV GX Devices* and the *Cyclone IV Dynamic Reconfiguration* chapter.

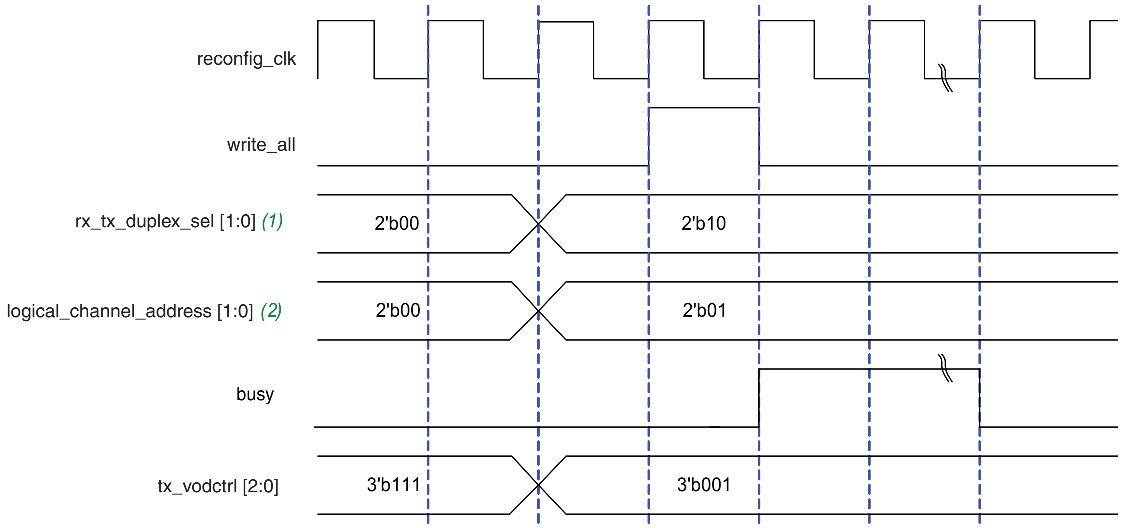
(2) For t_{LTD_Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.

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

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

Figure 3–4 shows the write transaction waveform for Method 1.

Figure 3–4. Write Transaction Waveform—Use 'logical_channel_address port' Option



Notes to Figure 3–4:

(1) In this waveform example, you are writing to only the transmitter portion of the channel.

(2) In this waveform example, the number of channels connected to the dynamic reconfiguration controller is four. Therefore, the `logical_channel_address` port is 2 bits wide.

Read Transaction

For example, to read the existing V_{OD} values from the transmit V_{OD} control registers of the transmitter portion of a specific channel controlled by the ALTGX_RECONFIG instance, perform the following steps:

1. Set the `logical_channel_address` input port to the logical channel address of the transceiver channel whose PMA controls you want to read (for example, `tx_vodctrl_out`).
2. Set the `rx_tx_duplex_sel` port to **2'b10** so that only the transmit PMA controls are read from the transceiver channel.
3. Ensure that the `busy` signal is low before you start a read transaction.
4. Assert the read signal for one `reconfig_clk` clock cycle. This initiates the read transaction.

The `busy` output status signal is asserted high to indicate that the dynamic reconfiguration controller is busy reading the PMA control values. When the read transaction has completed, the `busy` signal goes low. The `data_valid` signal is asserted to indicate that the data available at the read control signal is valid.

Table 1–21. Transceiver Specification for Cyclone IV GX Devices (Part 4 of 4)

Symbol/ Description	Conditions	C6			C7, I7			C8			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
PLD-Transceiver Interface											
Interface speed (F324 and smaller package)	—	25	—	125	25	—	125	25	—	125	MHz
Interface speed (F484 and larger package)	—	25	—	156.25	25	—	156.25	25	—	156.25	MHz
Digital reset pulse width	—	Minimum is 2 parallel clock cycles									

Notes to Table 1–21:

(1) This specification is valid for transmitter output jitter specification with a maximum total jitter value of 112 ps, typically for 3.125 Gbps SRIO and XAUI protocols.

(2) The minimum `reconfig_clk` frequency is 2.5 MHz if the transceiver channel is configured in **Transmitter Only** mode. The minimum `reconfig_clk` frequency is 37.5 MHz if the transceiver channel is configured in **Receiver Only** or **Receiver and Transmitter** mode.

(3) The device cannot tolerate prolonged operation at this absolute maximum.

(4) The rate matcher supports only up to ±300 parts per million (ppm).

(5) Supported for the F169 and F324 device packages only.

(6) Supported for the F484, F672, and F896 device packages only. Pending device characterization.

(7) To support CDR ppm tolerance greater than ±300 ppm, implement ppm detector in user logic and configure CDR to Manual Lock Mode.

(8) Asynchronous spread-spectrum clocking is not supported.

(9) For the EP4CGX30 (F484 package only), EP4CGX50, and EP4CGX75 devices, the CDR ppl tolerance is ±200 ppm.

(10) Time taken until `pll_locked` goes high after `pll_powerdown` deasserts.

(11) Time that the CDR must be kept in lock-to-reference mode after `rx_analogreset` deasserts and before `rx_locktodata` is asserted in manual mode.

(12) Time taken to recover valid data after the `rx_locktodata` signal is asserted in manual mode (Figure 1–2), or after `rx_freqlocked` signal goes high in automatic mode (Figure 1–3).

(13) Time taken to recover valid data after the `rx_locktodata` signal is asserted in manual mode.

(14) Time taken to recover valid data after the `rx_freqlocked` signal goes high in automatic mode.

(15) To support data rates lower than the minimum specification through oversampling, use the CDR in LTR mode only.

Figure 1–4 shows the differential receiver input waveform.

Figure 1–4. Receiver Input Waveform

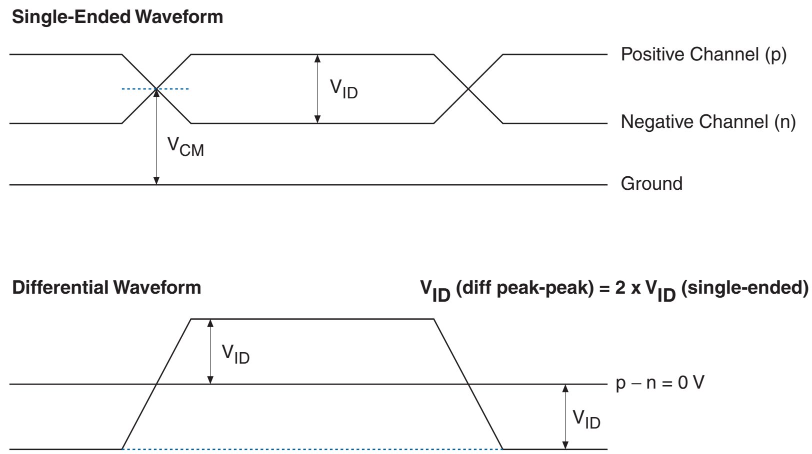


Figure 1–5 shows the transmitter output waveform.

Figure 1–5. Transmitter Output Waveform

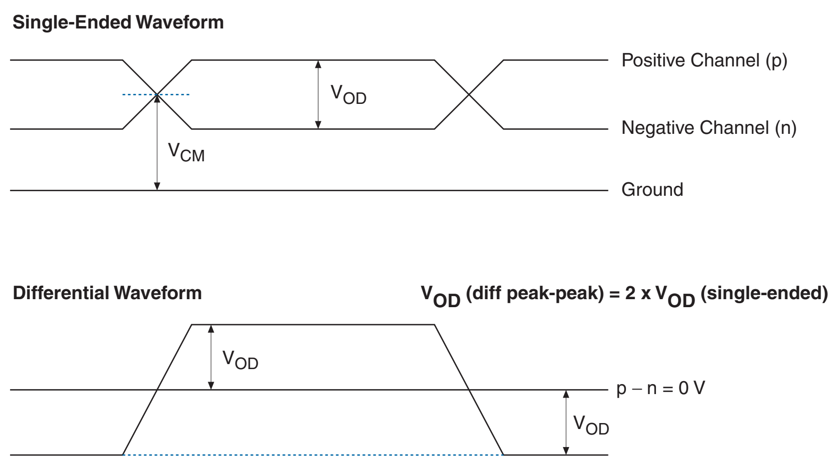


Table 1–22 lists the typical V_{OD} for Tx term that equals 100 Ω.

Table 1–22. Typical V_{OD} Setting, Tx Term = 100 Ω

Symbol	V_{OD} Setting (mV)					
	1	2	3	4 *(1)*	5	6
V_{OD} differential peak to peak typical (mV)	400	600	800	900	1000	1200

Note to Table 1–22:

(1) This setting is required for compliance with the PCIe protocol.

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.