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	1803
Number of Logic Elements/Cells	28848
Total RAM Bits	608256
Number of I/O	328
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce30f23c8ln

Chapter Revision Dates

The chapters in this document, Cyclone IV Device Handbook,, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Cyclone IV FPGA Device Family Overview
Revised: *March 2016*
Part Number: *CYIV-51001-2.0*

Chapter 2. Logic Elements and Logic Array Blocks in Cyclone IV Devices
Revised: *November 2009*
Part Number: *CYIV-51002-1.0*

Chapter 3. Memory Blocks in Cyclone IV Devices
Revised: *November 2011*
Part Number: *CYIV-51003-1.1*

Chapter 4. Embedded Multipliers in Cyclone IV Devices
Revised: *February 2010*
Part Number: *CYIV-51004-1.1*

Chapter 5. Clock Networks and PLLs in Cyclone IV Devices
Revised: *October 2012*
Part Number: *CYIV-51005-2.4*

Chapter 6. I/O Features in Cyclone IV Devices
Revised: *March 2016*
Part Number: *CYIV-51006-2.7*

Chapter 7. External Memory Interfaces in Cyclone IV Devices
Revised: *March 2016*
Part Number: *CYIV-51007-2.6*

Chapter 8. Configuration and Remote System Upgrades in Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51008-1.7*

Chapter 9. SEU Mitigation in Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51009-1.3*

Chapter 10. JTAG Boundary-Scan Testing for Cyclone IV Devices
Revised: *December 2013*
Part Number: *CYIV-51010-1.3*

Chapter 11. Power Requirements for Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51011-1.3*

2. Logic Elements and Logic Array Blocks in Cyclone IV Devices

CYIV-51002-1.0

This chapter contains feature definitions for logic elements (LEs) and logic array blocks (LABs). Details are provided on how LEs work, how LABs contain groups of LEs, and how LABs interface with the other blocks in Cyclone® IV devices.

Logic Elements

Logic elements (LEs) are the smallest units of logic in the Cyclone IV device architecture. LEs are compact and provide advanced features with efficient logic usage. Each LE has the following features:

- A four-input look-up table (LUT), which can implement any function of four variables
- A programmable register
- A carry chain connection
- A register chain connection
- The ability to drive the following interconnects:
  - Local
  - Row
  - Column
  - Register chain
  - Direct link
- Register packing support
- Register feedback support

Control Signals

The clock-enable control signal controls the clock entering the input and output registers and the entire M9K memory block. This signal disables the clock so that the M9K memory block does not see any clock edges and does not perform any operations.

The `rden` and `wren` control signals control the read and write operations for each port of M9K memory blocks. You can disable the `rden` or `wren` signals independently to save power whenever the operation is not required.

Parity Bit Support

Parity checking for error detection is possible with the parity bit along with internal logic resources. Cyclone IV devices M9K memory blocks support a parity bit for each storage byte. You can use this bit as either a parity bit or as an additional data bit. No parity function is actually performed on this bit.

Byte Enable Support

Cyclone IV devices M9K memory blocks support byte enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previous written value. The `wren` signals, along with the byte-enable (`byteena`) signals, control the write operations of the RAM block. The default value of the `byteena` signals is high (enabled), in which case writing is controlled only by the `wren` signals. There is no clear port to the `byteena` registers. M9K blocks support byte enables when the write port has a data width of ×16, ×18, ×32, or ×36 bits.

Byte enables operate in one-hot manner, with the LSB of the `byteena` signal corresponding to the least significant byte of the data bus. For example, if `byteena = 01` and you are using a RAM block in ×18 mode, `data[8..0]` is enabled and `data[17..9]` is disabled. Similarly, if `byteena = 11`, both `data[8..0]` and `data[17..9]` are enabled. Byte enables are active high.

Table 3–2 lists the byte selection.

Table 3–2. byteena for Cyclone IV Devices M9K Blocks *(1)*

byteena[3..0]	Affected Bytes			
	datain ×16	datain ×18	datain ×32	datain ×36
[0] = 1	[7..0]	[8..0]	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]	[15..8]	[17..9]
[2] = 1	—	—	[23..16]	[26..18]
[3] = 1	—	—	[31..24]	[35..27]

Note to Table 3–2:

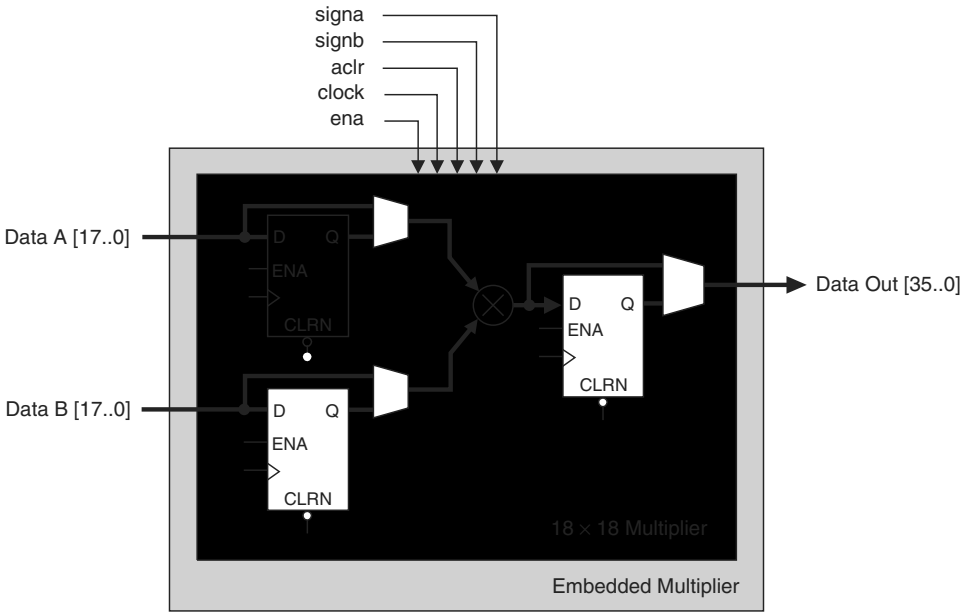
(1) Any combination of byte enables is possible.

18-Bit Multipliers

You can configure each embedded multiplier to support a single 18 × 18 multiplier for input widths of 10 to 18 bits.

Figure 4–3 shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 4–3. 18-Bit Multiplier Mode



All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the `signa` and `signb` signals and send these signals through dedicated input registers.

For more information about the number of GCLK networks in each device density, refer to the *Cyclone IV FPGA Device Family Overview* chapter.

GCLK Network

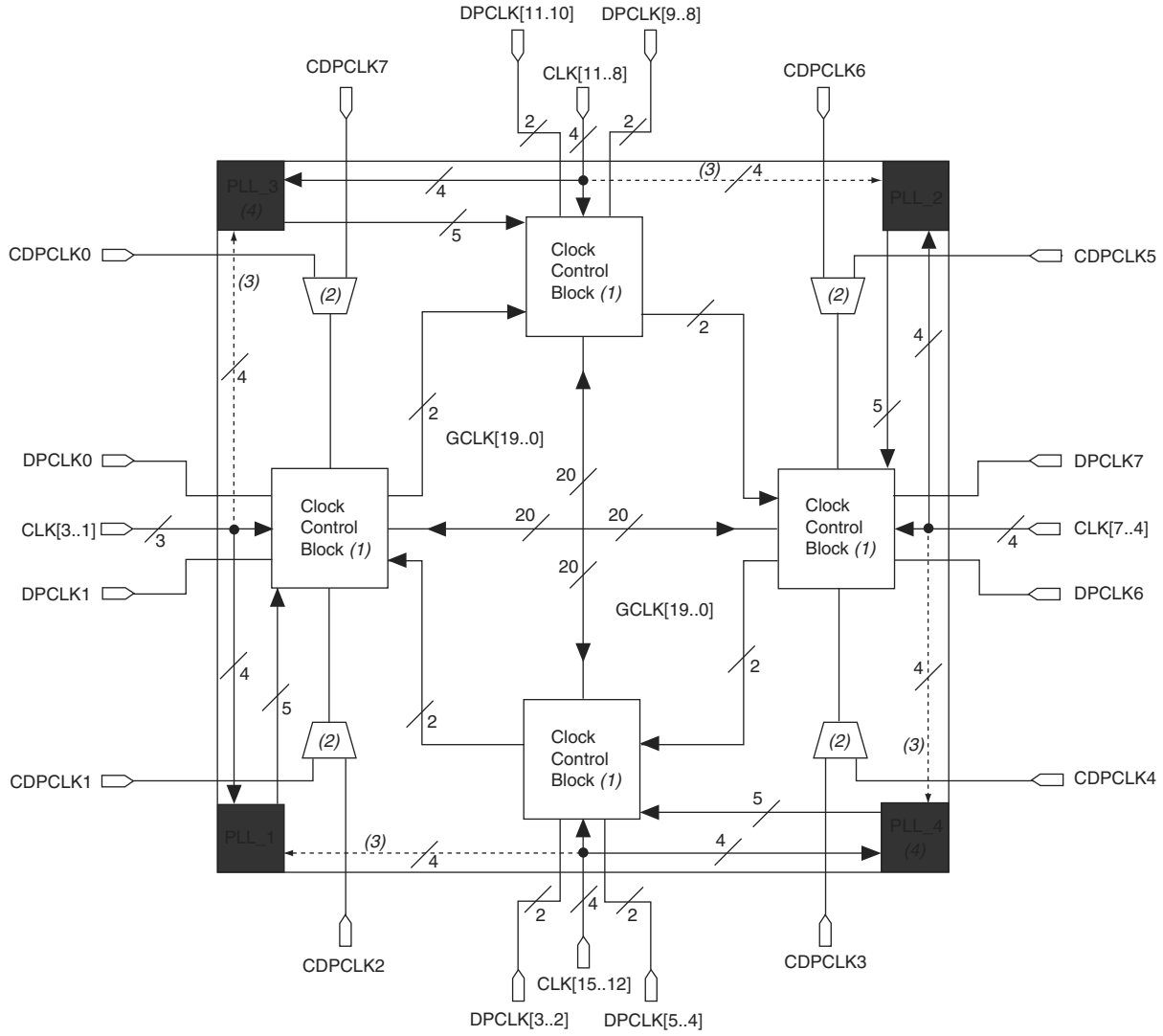
GCLKs drive throughout the entire device, feeding all device quadrants. All resources in the device (I/O elements, logic array blocks (LABs), dedicated multiplier blocks, and M9K memory blocks) can use GCLKs as clock sources. Use these clock network resources for control signals, such as clock enables and clears fed by an external pin. Internal logic can also drive GCLKs for internally generated GCLKs and asynchronous clears, clock enables, or other control signals with high fan-out.

Table 5–1, Table 5–2 on page 5–4, and Table 5–3 on page 5–7 list the connectivity of the clock sources to the GCLK networks.

Table 5–1. GCLK Network Connections for EP4CGX15, EP4CGX22, and EP4CGX30 (1), (2) (Part 1 of 2)

GCLK Network Clock Sources	GCLK Networks																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CLK4/DIFFCLK_2n	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
CLK5/DIFFCLK_2p	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
CLK6/DIFFCLK_3n	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—
CLK7/DIFFCLK_3p	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
CLK8/DIFFCLK_5n	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—
CLK9/DIFFCLK_5p	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
CLK10/DIFFCLK_4n/REFCLK1n	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—
CLK11/DIFFCLK_4p/REFCLK1p	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
CLK12/DIFFCLK_7p/REFCLK0p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓
CLK13/DIFFCLK_7n/REFCLK0n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—
CLK14/DIFFCLK_6p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓
CLK15/DIFFCLK_6n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL_1_C0	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL_1_C1	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓
PLL_1_C2	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL_1_C3	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—
PLL_1_C4	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓
PLL_2_C0	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
PLL_2_C1	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—
PLL_2_C2	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—
PLL_2_C3	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—
PLL_2_C4	—	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—
PLL_3_C0	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—

Figure 5–4. Clock Networks and Clock Control Block Locations in Cyclone IV E Devices



Notes to Figure 5–4:

(1) There are five clock control blocks on each side.

(2) Only one of the corner CDPCLK pins in each corner can feed the clock control block at a time. You can use the other CDPCLK pins as general-purpose I/O (GPIO) pins.

(3) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.

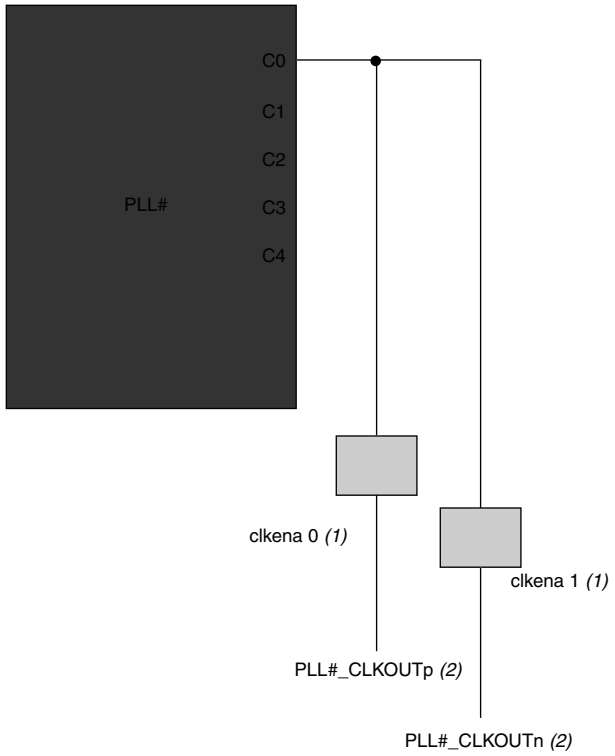
(4) PLL_3 and PLL_4 are not available in EP4CE6 and EP4CE10 devices.

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

- Four clock input pins
- Ten PLL counter outputs (five from each adjacent PLLs)
- Two, four, or six DPCLK pins from the top, bottom, and right sides of the device
- Five signals from internal logic

Figure 5–11 shows the external clock outputs for PLLs.

Figure 5–11. External Clock Outputs for PLLs



Notes to Figure 5–11:

(1) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

(2) PLL#_CLKOUTp and PLL#_CLKOUTn pins are dual-purpose I/O pins that you can use as one single-ended clock output or one differential clock output. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is configured as a regular user I/O.

Each pin of a differential output pair is 180° out of phase. The Quartus II software places the NOT gate in your design into the I/O element to implement 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins.

To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *Cyclone IV Device I/O Features* chapter.

Cyclone IV PLLs can drive out to any regular I/O pin through the GCLK. You can also use the external clock output pins as GPIO pins if external PLL clocking is not required.

Programmable Bandwidth

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

Phase Shift Implementation

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

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

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

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

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

Equation 5–1. Fine Resolution Phase Shift

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

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

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

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

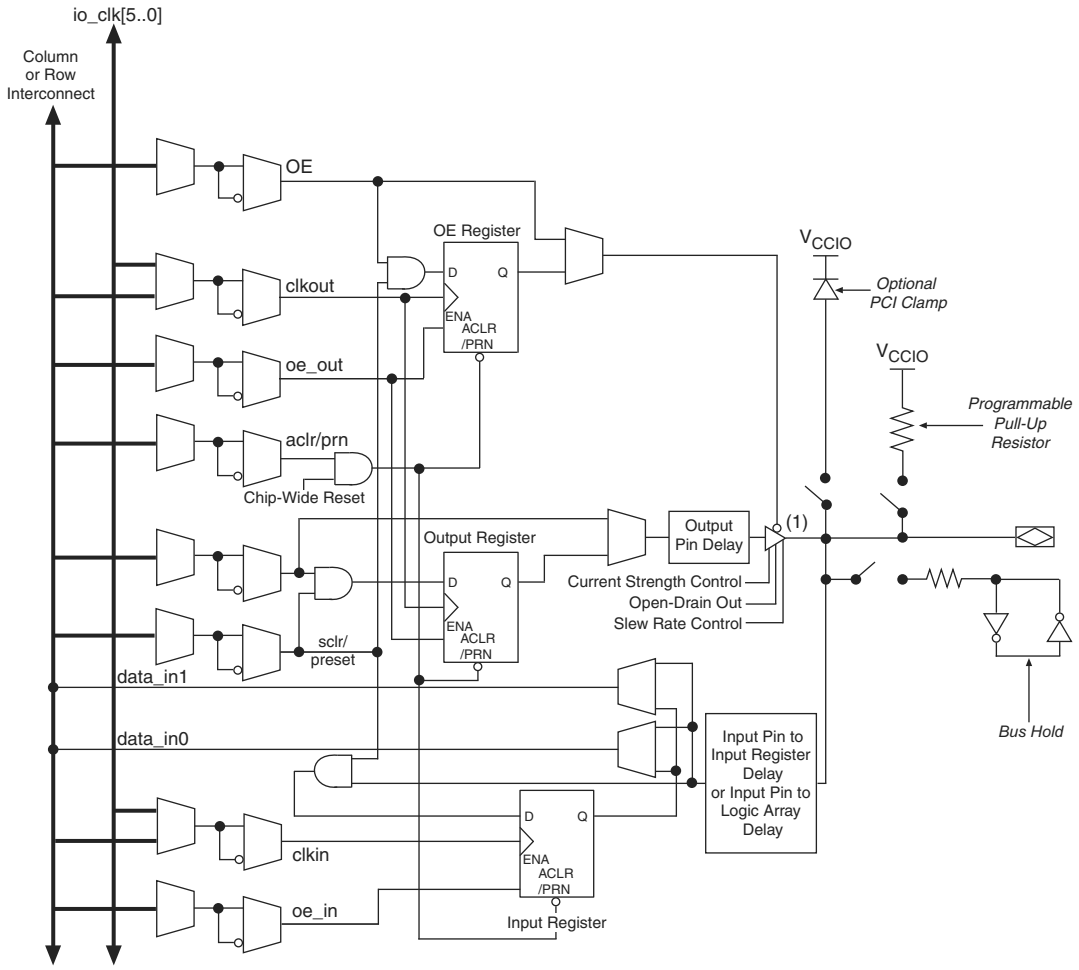
Equation 5–2. Coarse Resolution Phase Shift

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

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

Figure 6–1 shows the Cyclone IV devices IOE structure for single data rate (SDR) operation.

Figure 6–1. Cyclone IV IOEs in a Bidirectional I/O Configuration for SDR Mode



Note to Figure 6–1:

(1) Tri-state control is not available for outputs configured with true differential I/O standards.

I/O Element Features

The Cyclone IV IOE offers a range of programmable features for an I/O pin. These features increase the flexibility of I/O utilization and provide a way to reduce the usage of external discrete components, such as pull-up resistors and diodes.

Programmable Current Strength

The output buffer for each Cyclone IV I/O pin has a programmable current strength control for certain I/O standards.

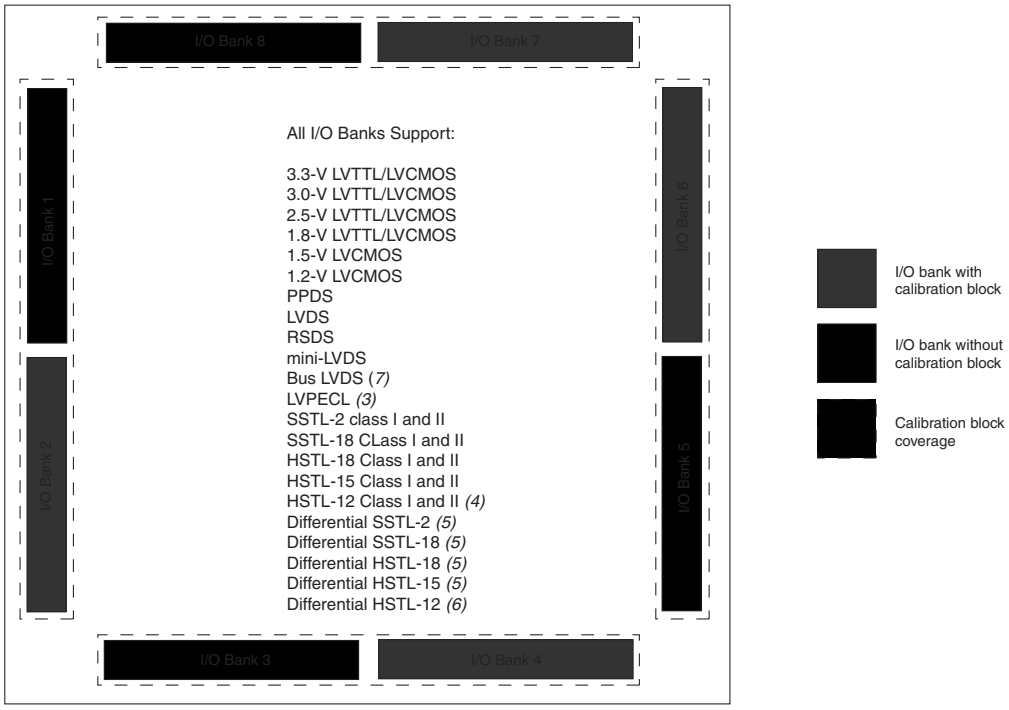
The LVTTL, LVCMOS, SSTL-2 Class I and II, SSTL-18 Class I and II, HSTL-18 Class I and II, HSTL-15 Class I and II, and HSTL-12 Class I and II I/O standards have several levels of current strength that you can control.

Table 6–3. Cyclone IV Devices Supported I/O Standards and Constraints (Part 2 of 3)

I/O Standard	Type	Standard Support	V_{CCIO} Level (in V)		Column I/O Pins			Row I/O Pins (1)	
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
2.5-V LVTTL / LVCMOS	Single-ended	JESD8-5	3.3/3.0/2.5 (3)	2.5	✓	✓	✓	✓	✓
1.8-V LVTTL / LVCMOS	Single-ended	JESD8-7	1.8/1.5 (3)	1.8	✓	✓	✓	✓	✓
1.5-V LVCMOS	Single-ended	JESD8-11	1.8/1.5 (3)	1.5	✓	✓	✓	✓	✓
1.2-V LVCMOS (4)	Single-ended	JESD8-12A	1.2	1.2	✓	✓	✓	✓	✓
SSTL-2 Class I, SSTL-2 Class II	voltage-referenced	JESD8-9A	2.5	2.5	✓	✓	✓	✓	✓
SSTL-18 Class I, SSTL-18 Class II	voltage-referenced	JESD815	1.8	1.8	✓	✓	✓	✓	✓
HSTL-18 Class I, HSTL-18 Class II	voltage-referenced	JESD8-6	1.8	1.8	✓	✓	✓	✓	✓
HSTL-15 Class I, HSTL-15 Class II	voltage-referenced	JESD8-6	1.5	1.5	✓	✓	✓	✓	✓
HSTL-12 Class I	voltage-referenced	JESD8-16A	1.2	1.2	✓	✓	✓	✓	✓
HSTL-12 Class II (9)	voltage-referenced	JESD8-16A	1.2	1.2	✓	✓	✓	—	—
PCI and PCI-X	Single-ended	—	3.0	3.0	✓	✓	✓	✓	✓
Differential SSTL-2 Class I or Class II	Differential (5)	JESD8-9A	—	2.5	—	✓	—	—	—
			2.5	—	✓	—	—	✓	—
Differential SSTL-18 Class I or Class II	Differential (5)	JESD815	—	1.8	—	✓	—	—	—
			1.8	—	✓	—	—	✓	—
Differential HSTL-18 Class I or Class II	Differential (5)	JESD8-6	—	1.8	—	✓	—	—	—
			1.8	—	✓	—	—	✓	—
Differential HSTL-15 Class I or Class II	Differential (5)	JESD8-6	—	1.5	—	✓	—	—	—
			1.5	—	✓	—	—	✓	—
Differential HSTL-12 Class I or Class II	Differential (5)	JESD8-16A	—	1.2	—	✓	—	—	—
			1.2	—	✓	—	—	✓	—
PPDS (6)	Differential	—	—	2.5	—	✓	✓	—	✓
LVDS (10)	Differential	ANSI/TIA/EIA-644	2.5	2.5	✓	✓	✓	✓	✓
RSDS and mini-LVDS (6)	Differential	—	—	2.5	—	✓	✓	—	✓
BLVDS (8)	Differential	—	2.5	2.5	—	—	✓	—	✓

Figure 6–9 shows the overview of Cyclone IV E I/O banks.

Figure 6–9. Cyclone IV E I/O Banks *(1), (2)*



Notes to Figure 6–9:

(1) This is a top view of the silicon die. This is only a graphical representation. For exact pin locations, refer to the pin list and the Quartus II software.

(2) True differential (PPDS, LVDS, mini-LVDS, and RSDS I/O standards) outputs are supported in row I/O banks 1, 2, 5, and 6 only. External resistors are needed for the differential outputs in column I/O banks.

(3) The LVPECL I/O standard is only supported on clock input pins. This I/O standard is not supported on output pins.

(4) The HSTL-12 Class II is supported in column I/O banks 3, 4, 7, and 8 only.

(5) The differential SSTL-18 and SSTL-2, differential HSTL-18, and HSTL-15 I/O standards are supported only on clock input pins and phase-locked loops (PLLs) output clock pins. Differential SSTL-18, differential HSTL-18, and HSTL-15 I/O standards do not support Class II output.

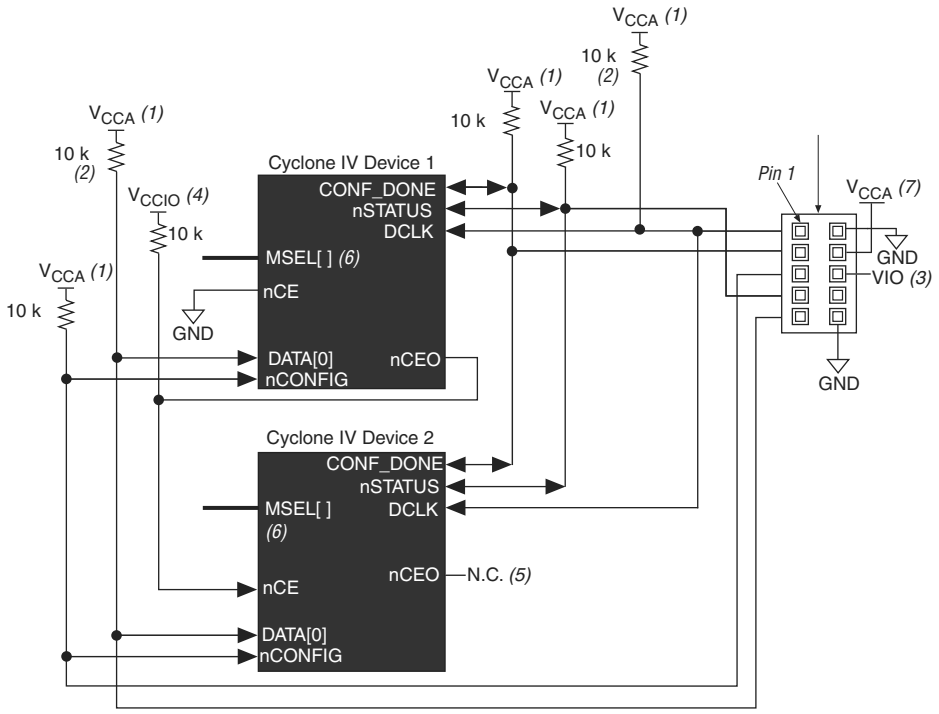
(6) The differential HSTL-12 I/O standard is only supported on clock input pins and PLL output clock pins. Differential HSTL-12 Class II is supported only in column I/O banks 3, 4, 7, and 8.

(7) BLVDS output uses two single-ended outputs with the second output programmed as inverted. BLVDS input uses true LVDS input buffer.

You can use a download cable to configure multiple Cyclone IV device configuration pins. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE are connected to every device in the chain. All devices in the chain utilize and enter user mode at the same time because all CONF_DONE pins are tied together.

In addition, the entire chain halts configuration if any device detects an error because the nSTATUS pins are tied together. Figure 8–18 shows the PS configuration for multiple Cyclone IV devices using a MasterBlaster, USB-Blaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 8–18. Multi-Device PS Configuration Using a Download Cable



Notes to Figure 8–18:

(1) You must connect the pull-up resistor to the same supply voltage as the V_{CCA} supply.

(2) The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This ensures that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming. Otherwise, it is a no connect.

(4) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(5) The nCEO pin of the last device in the chain is left unconnected or used as a user I/O pin.

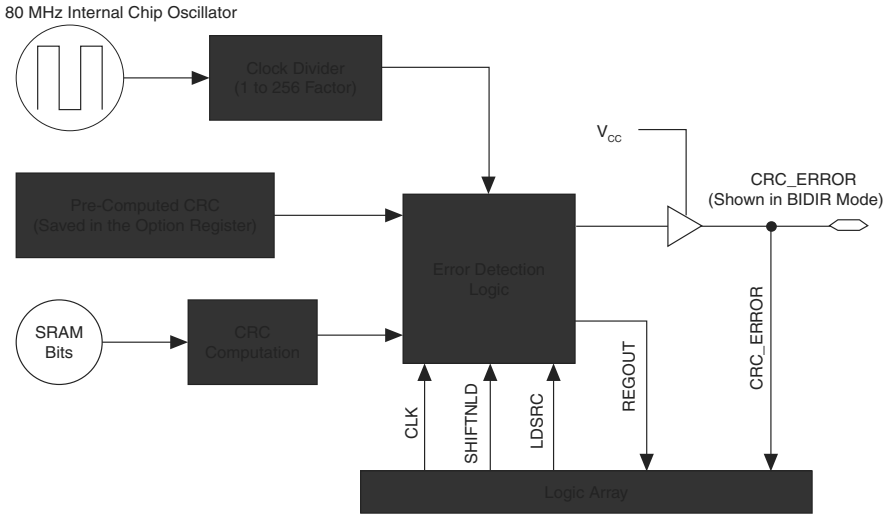
(6) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for PS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(7) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5 V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

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.

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.

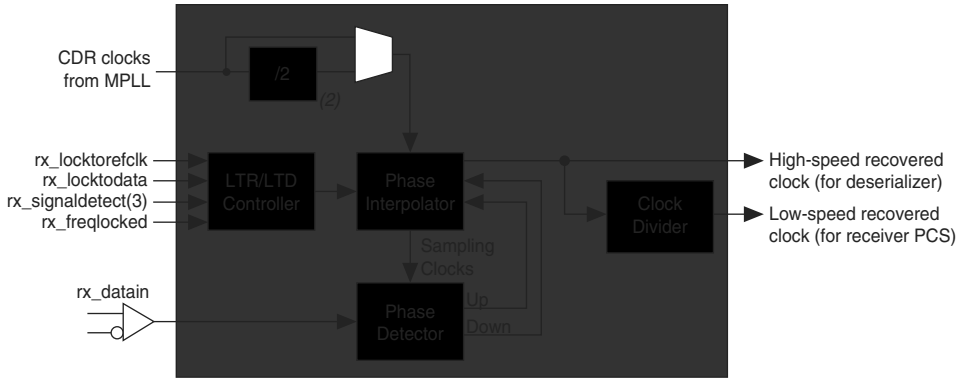
Example 9–1. Error Detection Block Diagram

```
cycloneiv_crcblock<crcblock_name>

(

.clk(<clock source>),

.shiftnld(<shiftnld source>),

.ldsrc(<ldsrc source>),

.crcerror(<crcerror out destination>),

.regout(<output destination>),

);
```

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:

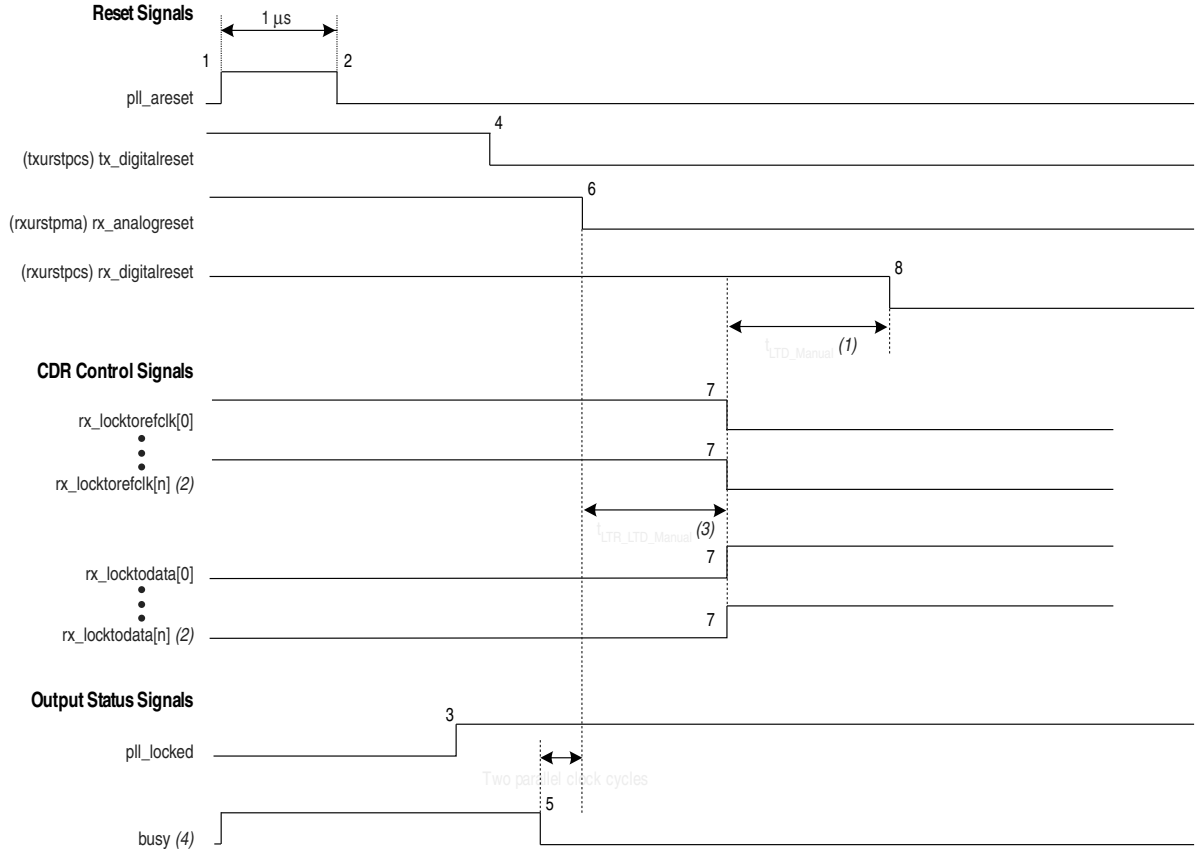
- Signal detection circuitry indicates the presence of valid signal levels at the receiver input buffer. This condition is valid for PCI Express (PIPE) mode only. CDR transitions are not dependent on signal detection circuitry in other modes.
- The recovered clock is within the configured part per million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

4. For the receiver operation, after deassertion of `busy` signal, wait for two parallel clock cycles to deassert the `rx_analogreset` signal.
5. Wait for the `rx_freqlocked` signal from each channel to go high. The `rx_freqlocked` signal of each channel may go high at different times (indicated by the slashed pattern at marker 7).
6. In a bonded channel group, when the `rx_freqlocked` signals of all the channels has gone high, from that point onwards, wait for at least t_{LTD_Auto} time for the receiver parallel clock to be stable, then deassert the `rx_digitalreset` signal (marker 8). At this point, all the receivers are ready for data traffic.

Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode

This configuration contains both a transmitter and receiver channel. When the receiver CDR is in manual lock mode, use the reset sequence shown in Figure 2–5.

Figure 2–5. Sample Reset Sequence for Bonded Configuration Receiver and Transmitter Channels—Receiver CDR in Manual Lock Mode

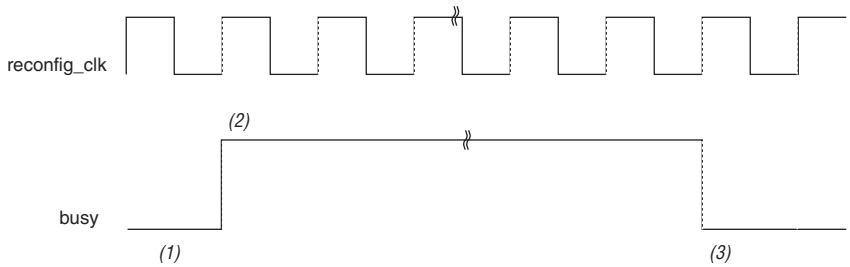


Notes to Figure 2–5:

(1) For t_{LTD_Manual} duration, refer to the *Cyclone IV Device Datasheet* chapter.
(2) The number of `rx_locktorefclk[n]` and `rx_locktodata[n]` signals depend on the number of channels configured. `n`=number of channels.
(3) For $t_{LTR\_LTD\_Manual}$ duration, refer to the *Cyclone IV Device Datasheet* chapter.
(4) The `busy` signal is asserted and deasserted only during initial power up when offset cancellation occurs. In subsequent reset sequences, the `busy` signal is asserted and deasserted only if there is a read or write operation to the ALTGX_RECONFIG megafunction.

Figure 3–3 shows the timing diagram for a offset cancellation process.

Figure 3–3. Dynamic Reconfiguration Signals Transition during Offset Cancellation



Notes to Figure 3–3:

(1) After device power up, the busy signal remains low for the first reconfig_clk cycle.
(2) The busy signal then gets asserted for the second reconfig_clk cycle, when the dynamic reconfiguration controller initiates the offset cancellation process.
(3) The deassertion of the busy signal indicates the successful completion of the offset cancellation process.

Functional Simulation of the Offset Cancellation Process

You must connect the ALTGX_RECONFIG instances to the ALTGX instances in your design for functional simulation. Functional simulation uses a reduced timing model of the dynamic reconfiguration controller. Therefore, the duration of the offset cancellation process is 16 reconfig_clk clock cycles for functional simulation only. The gxb_powerdown signal must not be asserted during the offset cancellation sequence (for functional simulation and silicon).

Dynamic Reconfiguration Modes

When you enable the dynamic reconfiguration feature, you can reconfigure the following portions of each transceiver channel dynamically, without powering down the other transceiver channels or the FPGA fabric of the device:

- Analog (PMA) controls reconfiguration
- Channel reconfiguration
- PLL reconfiguration

Table 3–3 lists the supported dynamic reconfiguration modes for Cyclone IV GX devices.

Table 3–3. Cyclone IV GX Supported Dynamic Reconfiguration Mode (Part 1 of 2)

Dynamic Reconfiguration Supported Mode	Operational Mode			Quartus II Instances			.mif Requirements
	Transmitter Only	Receiver Only	Transmitter and Receiver Only	ALTGX	ALTGX_RECONFIG	ALTPLL_RECONFIG	
Offset Cancellation	—	✓	✓	✓	✓	—	—
Analog (PMA) Controls Reconfiguration	✓	✓	✓	✓	✓	—	—

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

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
20-bit FPGA fabric-Transceiver Channel Interface with PCS-PMA set to 10 bits	Two 10-bit Data (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[25:16] - rx_dataout (MSByte)
	wo Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[27]: 8B/10B disparity error indicator (rx_disperr)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[28] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[29]: Rate Match FIFO deletion status indicator (rx_rmfifodatadeleted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[14] and rx_dataoutfull[30]: Rate Match FIFO insertion status indicator (rx_rmfifodatainserted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[15] and rx_dataoutfull[31]: 8B/10B running disparity indicator (rx_runningdisp)

Data Rate Reconfiguration Mode Using RX Local Divider

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

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

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

Embedded Multiplier Specifications

Table 1–26 lists the embedded multiplier specifications for Cyclone IV devices.

Table 1–26. Embedded Multiplier Specifications for Cyclone IV Devices

Mode	Resources Used	Performance					Unit
	Number of Multipliers	C6	C7, I7, A7	C8	C8L, I8L	C9L	
9 × 9-bit multiplier	1	340	300	260	240	175	MHz
18 × 18-bit multiplier	1	287	250	200	185	135	MHz

Memory Block Specifications

Table 1–27 lists the M9K memory block specifications for Cyclone IV devices.

Table 1–27. Memory Block Performance Specifications for Cyclone IV Devices

Memory	Mode	Resources Used		Performance					Unit
		LEs	M9K Memory	C6	C7, I7, A7	C8	C8L, I8L	C9L	
M9K Block	FIFO 256 × 36	47	1	315	274	238	200	157	MHz
	Single-port 256 × 36	0	1	315	274	238	200	157	MHz
	Simple dual-port 256 × 36 CLK	0	1	315	274	238	200	157	MHz
	True dual port 512 × 18 single CLK	0	1	315	274	238	200	157	MHz

Configuration and JTAG Specifications

Table 1–28 lists the configuration mode specifications for Cyclone IV devices.

Table 1–28. Passive Configuration Mode Specifications for Cyclone IV Devices (1)

Programming Mode	V_{CCINT} Voltage Level (V)	DCLK f_{MAX}	Unit
Passive Serial (PS)	1.0 (3)	66	MHz
	1.2	133	MHz
Fast Passive Parallel (FPP) (2)	1.0 (3)	66	MHz
	1.2 (4)	100	MHz

Notes to Table 1–28:

(1) For more information about PS and FPP configuration timing parameters, refer to the *Configuration and Remote System Upgrades in Cyclone IV Devices* chapter.

(2) FPP configuration mode supports all Cyclone IV E devices (except for E144 package devices) and EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 only.

(3) V_{CCINT} = 1.0 V is only supported for Cyclone IV E 1.0 V core voltage devices.

(4) Cyclone IV E devices support 1.2 V V_{CCINT}. Cyclone IV E 1.2 V core voltage devices support 133 MHz DCLK f_{MAX} for EP4CE6, EP4CE10, EP4CE15, EP4CE22, EP4CE30, and EP4CE40 only.