Intel - EP4CE55F23C8LN Datasheet





Welcome to <u>E-XFL.COM</u>

Understanding <u>Embedded - FPGAs (Field</u> <u>Programmable Gate Array)</u>

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	3491
Number of Logic Elements/Cells	55856
Total RAM Bits	2396160
Number of I/O	324
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/ep4ce55f23c8ln

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Read or Write Clock Mode

Cyclone IV devices M9K memory blocks can implement read or write clock mode for FIFO and simple dual-port memories. In this mode, a write clock controls the data inputs, write address, and wren registers. Similarly, a read clock controls the data outputs, read address, and rden registers. M9K memory blocks support independent clock enables for both the read and write clocks.

When using read or write mode, if you perform a simultaneous read or write to the same address location, the output read data is unknown. If you require the output data to be a known value, use either single-clock mode, input clock mode, or output clock mode and choose the appropriate read-during-write behavior in the MegaWizard Plug-In Manager.

Single-Clock Mode

Cyclone IV devices M9K memory blocks can implement single-clock mode for FIFO, ROM, true dual-port, simple dual-port, and single-port memories. In this mode, you can control all registers of the M9K memory block with a single clock together with clock enable.

Design Considerations

This section describes designing with M9K memory blocks.

Read-During-Write Operations

"Same-Port Read-During-Write Mode" on page 3–16 and "Mixed-Port Read-During-Write Mode" on page 3–16 describe the functionality of the various RAM configurations when reading from an address during a write operation at that same address.

There are two read-during-write data flows: same-port and mixed-port. Figure 3–13 shows the difference between these flows.





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.





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.

- Low time count = 1 cycle
- rselodd = 1 effectively equals:
 - High time count = 1.5 cycles
 - Low time count = 1.5 cycles
 - Duty cycle = (1.5/3)% high time count and (1.5/3)% low time count

Scan Chain Description

Cyclone IV PLLs have a 144-bit scan chain.

Table 5–7 lists the number of bits for each component of the PLL.

Table 5–7.	Cyclone	IV PLL	Reprogramming Bits
------------	---------	--------	---------------------------

Plack Nomo	Number of Bits							
DIUCK Name	Counter	Other	Total					
C4 (1)	16	2 (2)	18					
C3	16	2 (2)	18					
C2	16	2 (2)	18					
C1	16	2 (2)	18					
CO	16	2 (2)	18					
М	16	2 (2)	18					
Ν	16	2 (2)	18					
Charge Pump	9	0	9					
Loop Filter ⁽³⁾	9	0	9					
Total number of bits:	number of bits: 144							

Notes to Table 5-7:

(1) LSB bit for C4 low-count value is the first bit shifted into the scan chain.

- (2) These two control bits include <code>rbypass</code>, for bypassing the counter, and <code>rselodd</code>, to select the output clock duty cycle.
- (3) MSB bit for loop filter is the last bit shifted into the scan chain.

Figure 5–24 shows the scan chain order of the PLL components.

Figure 5–24. PLL Component Scan Chain Order



Signal Name	Description	Source	Destination
scanclk	Free running clock from core used in combination with phasestep to enable or disable dynamic phase shifting. Shared with scanclk for dynamic reconfiguration.	GCLK or I/O pins	PLL reconfiguration circuit
phasedone	When asserted, it indicates to core logic that the phase adjustment is complete and PLL is ready to act on a possible second adjustment pulse. Asserts based on internal PLL timing. De-asserts on the rising edge of scanclk.	PLL reconfiguration circuit	Logic array or I/O pins

Table 5-12.	Dvnamic	Phase	Shiftina	Control	Signals	(Part 2 of 2	(
						1	

Table 5–13 lists the PLL counter selection based on the corresponding PHASECOUNTERSELECT setting.

Table 5–13. Phase Counter Select Mapping

	phasecounterselec	Salaata		
[2]	[1]	[0]	Selects	
0	0	0	All Output Counters	
0	0	1	M Counter	
0	1	0	C0 Counter	
0	1	1	C1 Counter	
1	0	0	C2 Counter	
1	0	1	C3 Counter	
1	1	0	C4 Counter	

To perform one dynamic phase-shift, follow these steps:

- 1. Set PHASEUPDOWN and PHASECOUNTERSELECT as required.
- 2. Assert PHASESTEP for at least two SCANCLK cycles. Each PHASESTEP pulse allows one phase shift.
- 3. Deassert PHASESTEP after PHASEDONE goes low.
- 4. Wait for PHASEDONE to go high.
- 5. Repeat steps 1 through 4 as many times as required to perform multiple phaseshifts.

<code>PHASEUPDOWN</code> and <code>PHASECOUNTERSELECT</code> signals are synchronous to <code>SCANCLK</code> and must meet the t_{su} and t_h requirements with respect to the <code>SCANCLK</code> edges.

You can repeat dynamic phase-shifting indefinitely. For example, in a design where the VCO frequency is set to 1,000 MHz and the output clock frequency is set to 100 MHz, performing 40 dynamic phase shifts (each one yields 125 ps phase shift) results in shifting the output clock by 180°, in other words, a phase shift of 5 ns.

Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 2 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSELO	POR Delay	Configuration Voltage Standard (V) $^{(1)}$
JTAG-based configuration (2)	(3)	(3)	(3)	(3)	_	_

Notes to Table 8-4:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

> Smaller Cyclone IV E devices or package options (E144 and F256 packages) do not have the MSEL[3] pin. The AS Fast POR configuration scheme at 3.0- or 2.5-V configuration voltage standard and the AP configuration scheme are not supported in Cyclone IV E devices without the MSEL[3] pin. To configure these devices with other supported configuration schemes, select MSEL[2..0] pins according to the MSEL settings in Table 8–5.

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSELO	POR Delay	Configuration Voltage Standard (V) ⁽¹⁾
	1	1	0	1	Fast	3.3
٨٩	0	1	0	0	Fast	3.0, 2.5
AU	0	0	1	0	Standard	3.3
	0	0	1	1	Standard	3.0, 2.5
	0	1	0	1	Fast	3.3
	0	1	1	0	Fast	1.8
AP	0	1	1	1	Standard	3.3
	1	0	1	1	Standard	3.0, 2.5
	1	0	0	0	Standard	1.8
DC	1	1	0	0	Fast	3.3, 3.0, 2.5
10	0	0	0	0	Standard	3.3, 3.0, 2.5
EDD	1	1	1	0	Fast	3.3, 3.0, 2.5
	1	1	1	1	Fast	1.8, 1.5
JTAG-based configuration (2)	(3)	(3)	(3)	(3)	_	_

 Table 8–5.
 Configuration Schemes for Cyclone IV E Devices

Notes to Table 8-5:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration. For more information about the USB-Blaster download cable, refer to the USB-Blaster *Download Cable User Guide*. For more information about the ByteBlaster II download cable, refer to the *ByteBlaster II Download Cable User Guide*.

Figure 8-6 shows the download cable connections to the serial configuration device.





Notes to Figure 8-6:

- (1) Connect these pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.
- (2) 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.
- (3) Power up the V_{CC} of the ByteBlaster II or USB-Blaster download cable with the 3.3-V supply.
- (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) The diodes and capacitors must be placed as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V. The external diodes and capacitors are required to prevent damage to the Cyclone IV device AS configuration input pins due to possible overshoot when programming the serial configuration device with a download cable. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.
- (6) When cascading Cyclone IV devices in a multi-device AS configuration, connect the repeater buffers between the master and slave devices for DATA [0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.
- (7) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA [1] pin in AP and FPP modes.
- (8) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

FPP Configuration

The FPP configuration in Cyclone IV devices is designed to meet the increasing demand for faster configuration time. Cyclone IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Cyclone IV devices with an intelligent host, such as a MAX II device or microprocessor with flash memory. If your system already contains a CFI flash memory, you can use it for the Cyclone IV device configuration storage as well. The MAX II PFL feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control configuration from the flash memory device to the Cyclone IV device.

- ***** For more information about the PFL, refer to *AN* 386: Using the Parallel Flash Loader with the Quartus II Software.
- FPP configuration is supported in EP4CGX30 (only for F484 package), EP4CGX50, EP4CGX75, EP4CGX110, EP4CGX150, and all Cyclone IV E devices.
- The FPP configuration is not supported in E144 package of Cyclone IV E devices.
- Cyclone IV devices do not support enhanced configuration devices for FPP configuration.

FPP Configuration Using an External Host

FPP configuration using an external host provides a fast method to configure Cyclone IV devices. In the FPP configuration scheme, you can use an external host device to control the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store configuration data in an **.rbf**, **.hex**, or **.ttf** format. When using the external host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to You can perform JTAG testing on Cyclone IV devices before, during, and after configuration. Cyclone IV devices support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration. All other JTAG instructions can only be issued by first interrupting configuration and reprogramming I/O pins with the ACTIVE_DISENGAGE and CONFIG_IO instructions.

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

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

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

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

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

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



Figure 8–26. JTAG Configuration of Multiple Devices Using a Download Cable (1.2, 1.5, and 1.8-V V_{CCIO} Powering the JTAG Pins)

Notes to Figure 8-26:

- (1) Connect these pull-up resistors to the V_{CCI0} 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 a 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 cable, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.
- (4) You must connect the nCE pin to GND or driven low for successful JTAG configuration.
- (5) Power up the V_{CC} of the ByteBlaster II or USB-Blaster cable with supply from V_{CCI0}. The 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 and the USB-Blaster Download Cable User Guide.
- (6) Resistor value can vary from 1 k Ω to 10 k Ω .
 - IF a non-Cyclone IV device is cascaded in the JTAG-chain, TDO of the non-Cyclone IV device driving into TDI of the Cyclone IV device must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

The CONF_DONE and nSTATUS signals are shared in multi-device AS, AP, PS, and FPP configuration chains to ensure that the devices enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 8–25 or Figure 8–26, in which each of the CONF_DONE and nSTATUS signals are isolated so that each device can enter user mode individually.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device drives the nCE pin of the next device low when it has successfully been JTAG configured. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

ACTIVE_DISENGAGE

The ACTIVE_DISENGAGE instruction places the active configuration controller (AS and AP) into an idle state prior to JTAG programming. The two purposes of placing the active controller in an idle state are:

- To ensure that it is not trying to configure the device during JTAG programming
- To allow the controllers to properly recognize a successful JTAG programming that results in the device reaching user mode

The ACTIVE_DISENGAGE instruction is required before JTAG programming regardless of the current state of the Cyclone IV device if the MSEL pins are set to an AS or AP configuration scheme. If the ACTIVE_DISENGAGE instruction is issued during a passive configuration scheme (PS or FPP), it has no effect on the Cyclone IV device. Similarly, the CONFIG_IO instruction is issued after an ACTIVE_DISENGAGE instruction, but is no longer required to properly halt configuration. Table 8–17 lists the required, recommended, and optional instructions for each configuration mode. The ordering of the required instructions is a hard requirement and must be met to ensure functionality.

	Configuration Scheme and Current State of the Cyclone IV Device												
JTAG Instruction	Prior to User Mode (Interrupting Configuration)					User Mode				Power Up			
	PS	FPP	AS	AP	PS	FPP	AS	AP	PS	FPP	AS	AP	
ACTIVE_DISENGAGE	0	0	R	R	0	0	0	R	0	0	R	R	
CONFIG_IO	Rc	Rc	0	0	0	0	0	0	NA	NA	NA	NA	
Other JTAG instructions	0	0	0	0	0	0	0	0	0	0	0	0	
JTAG_PROGRAM	R	R	R	R	R	R	R	R	R	R	R	R	
CHECK_STATUS	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	
JTAG_STARTUP	R	R	R	R	R	R	R	R	R	R	R	R	
JTAG TAP Reset/other instruction	R	R	R	R	R	R	R	R	R	R	R	R	

Table 8–17. JTAG Programming Instruction Flows (1)

Note to Table 8-17:

(1) "R" indicates that the instruction must be executed before the next instruction, "O" indicates the optional instruction, "Rc" indicates the recommended instruction, and "NA" indicates that the instruction is not allowed in this mode.

In the AS or AP configuration scheme, the ACTIVE_DISENGAGE instruction puts the active configuration controller into idle state. If a successful JTAG programming is executed, the active controller is automatically re-engaged after user mode is reached through JTAG programming. This causes the active controller to transition to their respective user mode states.

If JTAG programming fails to get the Cyclone IV device to enter user mode and re-engage active programming, there are available methods to achieve this:

■ In AS configuration scheme, you can re-engage the AS controller by moving the JTAG TAP controller to the reset state or by issuing the ACTIVE_ENGAGE instruction.

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

 Table 9–4.
 Error Detection Registers

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

Error Detection Timing

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

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

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

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

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

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

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

Equation 9-1.

ror detection frequency	=	80 MH
in accelering accelering		2^n

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

The high-speed serial link can be AC- or DC-coupled, depending on the serial protocol implementation. In an AC-coupled link, the AC-coupling capacitor blocks the transmitter DC common mode voltage as shown in Figure 1–12. Receiver OCT and on-chip biasing circuitry automatically restores the common mode voltage. The biasing circuitry is also enabled by enabling OCT. If you disable the OCT, then you must externally terminate and bias the receiver. AC-coupled links are required for PCIe, GbE, Serial RapidIO, SDI, XAUI, SATA, V-by-One and Display Port protocols.



Figure 1–12. AC-Coupled Link with OCT

In a DC-coupled link, the transmitter DC common mode voltage is seen unblocked at the receiver input buffer as shown in Figure 1–13. The link common mode voltage depends on the transmitter common mode voltage and the receiver common mode voltage. When using the receiver OCT and on-chip biasing circuitry in a DC coupled link, you must ensure the transmitter common mode voltage is compatible with the receiver common mode requirements. If you disable the OCT, you must terminate and bias the receiver externally and ensure compatibility between the transmitter and the receiver common mode voltage.





Figure 1–14 shows the receiver input buffer block diagram.





The receiver input buffers support the following features:

Clock Rate Compensation

In XAUI mode, the rate match FIFO compensates up to ± 100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The XAUI protocol requires the transmitter to send /R/ (/K28.0/) code groups simultaneously on all four lanes (denoted as ||R|| column) during inter-packet gaps, adhering to rules listed in the IEEE P802.3ae specification.

The rate match operation begins after rx_syncstatus and rx_channelaligned are asserted. The rx_syncstatus signal is from the word aligner, indicating that synchronization is acquired on all four channels, while rx_channelaligned signal is from the deskew FIFO, indicating channel alignment.

The rate match FIFO looks for the ||R|| column (simultaneous /R/ code groups on all four channels) and deletes or inserts ||R|| columns to prevent the rate match FIFO from overflowing or under running. The rate match FIFO can insert or delete as many ||R|| columns as necessary to perform the rate match operation.

The rx_rmfifodatadeleted and rx_rmfifodatainserted flags that indicate rate match FIFO deletion and insertion events, respectively, are forwarded to the FPGA fabric. If an ||R|| column is deleted, the rx_rmfifodeleted flag from each of the four channels goes high for one clock cycle per deleted ||R|| column. If an ||R|| column is inserted, the rx_rmfifoinserted flag from each of the four channels goes high for one clock cycle per deleted ||R|| column. If an ||R|| column is inserted, the rx_rmfifoinserted flag from each of the four channels goes high for one clock cycle per inserted ||R|| column.

The rate match FIFO does not insert or delete code groups automatically to overcome FIFO empty or full conditions. In this case, the rate match FIFO asserts the rx_rmfifofull and rx_rmfifoempty flags for at least three recovered clock cycles to indicate rate match FIFO full and empty conditions, respectively. You must then assert the rx_digitalreset signal to reset the receiver PCS blocks.

Deterministic Latency Mode

Deterministic Latency mode provides the transceiver configuration that allows no latency uncertainty in the datapath and features to strictly control latency variation. This mode supports non-bonded (×1) and bonded (×4) channel configurations, and is typically used to support CPRI and OBSAI protocols that require accurate delay measurements along the datapath. The Cyclone IV GX transceivers configured in Deterministic Latency mode provides the following features:

- registered mode phase compensation FIFO
- receive bit-slip indication
- transmit bit-slip control
- PLL PFD feedback

Figure 1–67 shows the transceiver configuration in Deterministic Latency mode.

Functional Mode			Deterministic Latency					
Channel Bonding			×1, ×4					
Low-Latency PCS				Disa	bled			
Word Aligner (Pattern Length)		Manual / (10	Alignment -Bit)			Bit (10	Slip -Bit)	
8B/10B Encoder/Decoder	Enab	led	Disa	bled	Ena	bled	Dise	abled
Rate Match FIFO	Disabled		Disabled		Disabled		Disabled	
Byte SERDES	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled
Data Rate (Gbps)	0.6- 3.125	0.6- 1.5625	0.6- 3.125	0.6- 1.5625	0.6- 3.125	0.6- 1.5625	0.6- 3.125	0.6- 1.5625
Byte Ordering	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
FPGA Fabric-to-Transceiver Interface Width	▼ 16-Bit	8-Bit	20-Bit	10-Bit	16-Bit	8-Bit	20-Bit	10-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)	60- 156.25	30- 156.25	60- 156.25	▼ 30- 156.25	60- 156.25	30- 156.25	60- 156.25	30- 156.25
TX PCS Latency (FPGA Fabric-Transceiver Interface Clock Cycles)	2.5 - 3.5	4 - 5	2.5 - 3.5	4 - 5	2.5 - 3	4	2.5 - 3	4
RX PCS Latency (FPGA Fabric-Transceiver Interface	5-6	8-9	5-6	8-9	5-6	8-9	5-6	8-9

Figure 1–67. Transceiver Configuration in Deterministic Latency Mode

Both CPRI and OBSAI protocols define the serial interface connecting the base station component (specifically channel cards) and remote radio heads (specifically radio frequency cards) in a radio base station system with fiber optics. The protocols require the accuracy of round trip delay measurement for single-hop and multi-hop connections to be within \pm 16.276 ns. The Cyclone IV GX transceivers support the following CPRI and OBSAI line rates using Deterministic Latency mode:

- CPRI —614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps, and 3.072 Gbps
- OBSAI—768 Mbps, 1.536 Gbps, and 3.072 Gbps

• For more information about deterministic latency implementation, refer to *AN 610: Implementing Deterministic Latency for CPRI and OBSAI Protocols in Stratix IV, HardCopy IV, Arria II GX, and Cyclone IV Devices.*

Registered Mode Phase Compensation FIFO

In Deterministic Latency mode, the RX phase compensation FIFO is set to registered mode while the TX phase compensation FIFO supports optional registered mode. When set into registered mode, the phase compensation FIFO acts as a register and eliminates the latency uncertainty through the FIFOs.

Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode

This configuration contains both a transmitter and a receiver channel. If you create a **Receiver and Transmitter** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in automatic lock mode, use the reset sequence shown in Figure 2–8.

Figure 2–8. Sample Reset Sequence of Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode



Notes to Figure 2-8:

- (1) For t_{LTD Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.
- (2) 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.

As shown in Figure 2–8, perform the following reset procedure for the receiver in CDR automatic lock mode:

- 1. After power up, assert pll_areset for a minimum period of 1 µs (the time between markers 1 and 2).
- 2. Keep the tx_digitalreset, rx_analogreset, and rx_digitalreset signals asserted during this time period. After you deassert the pll_areset signal, the multipurpose PLL starts locking to the transmitter input reference clock.
- 3. After the multipurpose PLL locks, as indicated by the pll_locked signal going high (marker 3), deassert tx_digitalreset. For receiver operation, after deassertion of busy signal, wait for two parallel clock cycles to deassert the rx_analogreset signal.
- 4. Wait for the rx freqlocked signal to go high (marker 7).
- 5. After the rx_freqlocked signal goes high, wait for at least t_{LTD_Auto}, then deassert the rx_digitalreset signal (marker 8). At this point, the transmitter and receiver are ready for data traffic.

Port Name	Input/ Output	Description				
		This is an optional equalizer DC gain write control.				
		The width of this signal is fixed to 2 bits if you enable either the Use 'logical_channel_address' port for Analog controls reconfiguration option or the Use same control signal for all the channels option in the Analog controls screen. Otherwise, the width of this signal is 2 bits per channel.				
		The following values are the legal settings allowed for this signal:				
		rx_eqdcgain[10] Corresponding ALTGX Corresponding				
rx_eqdcgain [10] ⁽¹⁾	Input	(dB) DC Gain value				
		2'b00 0 0				
		2′b01 1 3 ⁽²⁾				
		2'b10 2 6				
		All other values => N/A				
		For more information, refer to the "Programmable Equalization and DC Gain" section of the <i>Cyclone IV GX Device Datasheet</i> chapter.				
<pre>tx_vodctrl_out [20]</pre>	Output	This is an optional transmit V_{0D} read control signal. This signal reads out the value written into the V_{0D} control register. The width of this output signal depends on the number of channels controlled by the dynamic reconfiguration controller and also the configuration of the Use 'logical_channel_address' port for Analog controls reconfiguration option and the Use same control signal for all the channels option.				
tx_preemp_out [40]	Output	This is an optional pre-emphasis read control signal. This signal reads out the value written by its input control signal. The width of this output signal depends on the number of channels controlled by the dynamic reconfiguration controller and also the configuration of the Use 'logical_channel_address' port for Analog controls reconfiguration option and the Use same control signal for all the channels option.				
rx_eqctrl_out [30]	Output	This is an optional read control signal to read the setting of equalization setting of the ALTGX instance. The width of this output signal depends on the number of channels controlled by the dynamic reconfiguration controller and also the configuration of the Use 'logical_channel_address' port for Analog controls reconfiguration option and the Use same control signal for all the channels option.				
rx_eqdcgain_out [10]	Output	This is an optional equalizer DC gain read control signal. This signal reads out the settings of the ALTGX instance DC gain. The width of this output signal depends on the number of channels controlled by the dynamic reconfiguration controller and also the configuration of the Use 'logical_channel_address' port for Analog controls reconfiguration option and the Use same control signal for all the channels option.				
Transceiver Channel Reconfiguration Control/Status Signals						
		Set the following values at this signal to activate the appropriate dynamic reconfiguration mode:				
nononfig and		3'b000 = PMA controls reconfiguration mode. This is the default value.				
sel[20] ⁽³⁾	Input	3'b001 = Channel reconfiguration mode				
		All other values => N/A				
		reconfig_mode_sel[] is available as an input only when you enable more than one dynamic reconfiguration mode.				

Table 3–2. Dynamic Reconfiguration Controller Port List (ALTGX_RECONFIG Instance) (Part 6 of 7)

Table 3–7 lists the ALTGX megafunction ports for PLL Reconfiguration mode.

Port Name ⁽¹⁾	Input/ Output	Description	Comments
pll_areset [n0]	Input	 Resets the transceiver PLL. The pll_areset are asserted in two conditions: Used to reset the transceiver PLL during the reset sequence. During reset sequence, this signal is user controlled. After the transceiver PLL is reconfigured, this signal is asserted high by the ALTPLL_RECONFIG controller. At this time, this signal is not user controlled. 	You must connect the pll_areset port of ALTGX to the pll_areset port of the ALTPLL_RECONFIG megafunction. The ALTPLL_RECONFIG controller asserts the pll_areset port at the next rising clock edge after the pll_reconfig_done signal from the ALTGX megafunction goes high. After the pll_reconfig_done signal goes high, the transceiver PLL is reset. When the PLL reconfiguration is completed, this reset is performed automatically by the ALTPLL_RECONFIG megafunction and is not user controlled.
pll_scandata [n0]	Input	Receives the scan data input from the ALTPLL_RECONFIG megafunction.	The reconfigurable transceiver PLL received the scan data input through this port for the dynamically reconfigurable bits from the ALTPLL_RECONFIG controller.
pll_scanclk [n0]	Input	Drives the scanclk port on the reconfigurable transceiver PLL.	Connect the pll_scanclk port of the ALTGX megafunction to the ALTPLL_RECONFIG scanclk port.
pll_scancikena [n0]	Input	Acts as a clock enable for the scanclk port on the reconfigurable transceiver PLL.	Connect the pll_scanclkena port of the ALTGX megafunction to the ALTPLL_RECONFIG scanclk port.
pll_configupdate [n0]	Input	Drives the configupdate port on the reconfigurable transceiver PLL.	This port is connected to the pll_configupdate port from the ALTPLL_RECONFIG controller. After the final data bit is sent out, the ALTPLL_RECONFIG controller asserts this signal.
pll_reconfig_done[n0]	Output	This signal is asserted to indicate the reconfiguration process is done.	Connect the pll_reconfig_done port to the pll_scandone port on the ALTPLL_RECONFIG controller. The transceiver PLL scandone output signal drives this port and determines when the PLL is reconfigured.
pll_scandataout [n0]	Output	This port scan out the current configuration of the transceiver PLL.	Connect the pll_scandataout port to the pll_scandataout port of the ALTPLL_RECONFIG controller. This port reads the current configuration of the transceiver PLL and send it to the ALTPLL_RECONFIG megafunction.

Table 3–7. ALTGX Megafunction Port List for PLL Reconfiguration Mode

Note to Table 3-7:

(1) $\langle n \rangle$ = (number of transceiver PLLs configured in the ALTGX MegaWizard) - 1.

• For more information about the ALTPLL_RECONFIG megafunction port list, description and usage, refer to the *Phase-Locked Loop Reconfiguration* (*ALTPL_RECONFIG*) *Megafunction User Guide*.

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

Example 1–1. Impedance Change

$$\begin{split} \Delta R_V &= (3.15-3) \times 1000 \times -0.026 = -3.83 \\ \Delta R_T &= (85-25) \times 0.262 = 15.72 \\ \text{Because } \Delta R_V \text{ is negative,} \\ MF_V &= 1 \ / \ (3.83/100 + 1) = 0.963 \\ \text{Because } \Delta R_T \text{ is positive,} \\ MF_T &= 15.72/100 + 1 = 1.157 \\ MF &= 0.963 \times 1.157 = 1.114 \\ R_{\text{final}} &= 50 \times 1.114 = 55.71 \ \Omega \end{split}$$

Pin Capacitance

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

Table 1–11.	Pin Cap	acitance for	Cvclone I	V Devices	(1)
	I III Oup		0,0101101	I DUTIOUS	

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

Notes to Table 1-11:

(1) The pin capacitance applies to FBGA, UBGA, and MBGA packages.

(2) When you use the vref pin as a regular input or output, you can expect a reduced performance of toggle rate and t_{CO} because of higher pin capacitance.

(3) C_{VREFTB} for the EP4CE22 device is 30 pF.

Figure 1–2 shows the lock time parameters in manual mode.

LTD = lock-to-data. LTR = lock-to-reference.



Figure 1–2. Lock Time Parameters for Manual Mode

Figure 1–3 shows the lock time parameters in automatic mode.

Figure 1–3. Lock Time Parameters for Automatic Mode

