Intel - EP4CE6E22C6N 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	392
Number of Logic Elements/Cells	6272
Total RAM Bits	276480
Number of I/O	91
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	144-LQFP Exposed Pad
Supplier Device Package	144-EQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce6e22c6n

Email: info@E-XFL.COM

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

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.

Document Revision History

Table 4–3 lists the revision history for this chapter.

Table 4–3. Document Revision History

Date	Version	Changes
February 2010	1.1	Added Cyclone IV E devices in Table 4–1 for the Quartus II software version 9.1 SP1 release.
November 2009	1.0	Initial release.

GCLK Network Clock														GC	LK N	etwo	rks													
Sources		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
PLL_3_C0	—	—	—	—	—	—	—	—	—	—	—		~	—	—	\checkmark	—	\checkmark	—		_	—		_	\checkmark	—	—	\checkmark	—	\checkmark
PLL_3_C1	—	—	—	—	—	—	—	—	—		—	—	—	~		—	\checkmark			_	_	_	_	_		\checkmark	—		\checkmark	—
PLL_3_C2	—	—	—	—	—	—	—	—	—		_	_	\checkmark	_	\checkmark	—	—	_			_	_			~	_	\checkmark		—	—
PLL_3_C3	—	—	—	_	—	_	_	—	_		_	_		\checkmark	_	\checkmark	—					_				~	_	\checkmark	—	—
PLL_3_C4	-	—	—	—	—	—	—	—	—		—	_	—	—	\checkmark	—	\checkmark	\checkmark	_						_	—	\checkmark		\checkmark	\checkmark
PLL_4_C0	—	—	—	—	—	—	—	—	—				\checkmark	—		\checkmark	—	\checkmark	\checkmark			\checkmark		>					—	—
PLL_4_C1	-	—	—	—	—	—	—	—	—		—	_	—	\checkmark	_	—	\checkmark	_	_	>			>		_	—	_		—	—
PLL_4_C2	—	—	—	_	—	_	—	—	_	_			~	_	~	—	—		<	_	\checkmark	_	_	_					_	—
PLL_4_C3	_	—	—	—	—	—	—	—	—	_			—	\checkmark		\checkmark	—			\checkmark		\checkmark	_						_	—
PLL_4_C4	-	-	—	—	—	—	—	-	—	_		_	—	-	~	—	\checkmark	~		—	\checkmark	_	~	~					_	—
PLL_5_C0	\checkmark	—	\checkmark	—	—	—	—	—	—	_			—	—		—	—			_		_	_						_	—
PLL_5_C1	-	-	—	—	—	—	—	-	—	_		_	—	-		—	—			—	—	_	—	—					_	—
PLL_5_C2	_	—	—	—	—	—	—	—	—	_			—	—		—	—			_		_	_						_	—
PLL_5_C3	—	\checkmark	—	\checkmark	—	—	—	—	—		—	_	—	—	—	—	—					_		_		—	—		_	—
PLL_5_C4	—	—	\checkmark	—	\checkmark	\checkmark	—	—	—	_		—	—	—	—	—	—		_		_	_							_	—
PLL_6_C0	\checkmark	—	—	\checkmark	—	\checkmark	—	—	—		—	—	—	—	—	—	—		_	_		_	_	_		—			_	—
PLL_6_C1	—	—	—	—	—	—	—	—	—	_	—	—	—	—	—	—	—		_	_		_	_	_		—	—		_	—
PLL_6_C2	-	—	—	—	—	—	—	—	—		—	—	—	—	—	—	—		I	_			_	_	—	—	—	_	—	—
PLL_6_C3	—	—	—	—	—	—	—	—	—	_	—	—	—	—	—	—	—		_	_		_	_	_		—	—		_	—
PLL_6_C4	—	\checkmark	—	—	\checkmark	—	—	—	—	_	—	—	—	—	—	—	—			_		_	_	_	—	—	—		_	—
PLL_7_C0 (3)	-	—	—	—	—	—	\checkmark	—	—	~	—	\checkmark	—	—	—	—	—	—		_	_		_	_	—	—	—		—	—
PLL_7_C1 (3)	-	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C2 (3)	-	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C3 (3)	-	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C4 (3)	—	—	—	—	—	—	—	~	—	_	\checkmark		—	—		—	—	—	—	_	—	—	_	_	—	—		<u> </u>	—	—

Table 5-2. GCLK Network Connections for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices ^{(1), (2)} (Part 2 of 4)

5 5 Figure 5–6 shows a simplified version of the five clock control blocks on each side of the Cyclone IV E device periphery.





Note to Figure 5–6:

(1) The left and right sides of the device have two DPCLK pins; the top and bottom of the device have four DPCLK pins.

GCLK Network Power Down

You can disable a Cyclone IV device's GCLK (power down) using both static and dynamic approaches. In the static approach, configuration bits are set in the configuration file generated by the Quartus II software, which automatically disables unused GCLKs. The dynamic clock enable or disable feature allows internal logic to control clock enable or disable the GCLKs in Cyclone IV devices.

When a clock network is disabled, all the logic fed by the clock network is in an off-state, thereby reducing the overall power consumption of the device. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 5–1 on page 5–11.

You can set the input clock sources and the clkena signals for the GCLK multiplexers through the Quartus II software using the ALTCLKCTRL megafunction.

For more information, refer to the *ALTCLKCTRL Megafunction User Guide.*

clkena Signals

Cyclone IV devices support clkena signals at the GCLK network level. This allows you to gate-off the clock even when a PLL is used. Upon re-enabling the output clock, the PLL does not need a resynchronization or re-lock period because the circuit gates off the clock at the clock network level. In addition, the PLL can remain locked independent of the clkena signals because the loop-related counters are not affected. The first Cyclone IV device in the chain is the configuration master and it controls the configuration of the entire chain. Other Altera devices that support PS configuration can also be part of the chain as configuration slaves.

IP In the multi-device AS configuration, the board trace length between the serial configuration device and the master device of the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–3 on page 8–13. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin. However, the subsequent devices in the chain keep this shared CONF_DONE line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives a high level on CONF_DONE line and all devices simultaneously enter initialization mode.

Although you can cascade Cyclone IV devices, serial configuration devices cannot be cascaded or chained together.

If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device, enable the compression feature, or both. When configuring multiple devices, the size of the bitstream is the sum of the individual device's configuration bitstream.

Configuring Multiple Cyclone IV Devices with the Same Design

Certain designs require that you configure multiple Cyclone IV devices with the same design through a configuration bitstream, or a **.sof**. You can do this through the following methods:

- Multiple .sof
- Single .sof
- For both methods, the serial configuration devices cannot be cascaded or chained together.

Multiple SRAM Object Files

Two copies of the **.sof** are stored in the serial configuration device. Use the first copy to configure the master device of the Cyclone IV device and the second copy to configure all remaining slave devices concurrently. All slave devices must have the same density and package. The setup is similar to Figure 8–3 on page 8–13.

To configure four identical Cyclone IV devices with the same **.sof**, you must set up the chain similar to the example shown in Figure 8–4. The first device is the master device and its MSEL pins must be set to select AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select PS configuration. The nCEO pin from the master device drives the nCE input pins on all three slave devices, as well as the DATA and DCLK pins that connect in parallel to all

For more information about the operation of the Micron P30 Parallel NOR and P33 Parallel NOR flash memories, search for the keyword "P30" or "P33" on the Micron website (www.micron.com) to obtain the P30 or P33 family datasheet.

Single-Device AP Configuration

The following groups of interface pins are supported in Micron P30 and P33 flash memories:

- Control pins
- Address pins
- Data pins

The following control signals are from the supported parallel flash memories:

- CLK
- active-low reset (RST#)
- active-low chip enable (CE#)
- active-low output enable (OE#)
- active-low address valid (ADV#)
- active-low write enable (WE#)

The supported parallel flash memories output a control signal (WAIT) to Cyclone IV E devices to indicate when synchronous data is ready on the data bus. Cyclone IV E devices have a 24-bit address bus connecting to the address bus (A[24:1]) of the flash memory. A 16-bit bidirectional data bus (DATA[15..0]) provides data transfer between the Cyclone IV E device and the flash memory.

The following control signals are from the Cyclone IV E device to flash memory:

- DCLK
- active-low hard rest (nRESET)
- active-low chip enable (FLASH_nCE)
- active-low output enable for the DATA [15..0] bus and WAIT pin (nOE)
- active-low address valid signal and is used to write data into the flash (nAVD)
- active-low write enable and is used to write data into the flash (nWE)

PS Configuration Using an External Host

In the PS configuration scheme, you can use an intelligent host such as a MAX II device or microprocessor that controls the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store the configuration data in **.rbf**, **.hex**, or **.ttf** format.

Figure 8–13 shows the configuration interface connections between a Cyclone IV device and an external host device for single-device configuration.

Figure 8–13. Single-Device PS Configuration Using an External Host



Notes to Figure 8-13:

- (1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.
- (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) 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.
- (4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA [0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

To begin the configuration, the external host device must generate a low-to-high transition on the nCONFIG pin. When nSTATUS is pulled high, the external host device must place the configuration data one bit at a time on DATA[0]. If you use configuration data in **.rbf**, **.ttf**, or **.hex**, you must first send the LSB of each data byte. For example, if the **.rbf** contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must send to the device is:

0100-0000 1101-1000 0111-0111 1000-0000 0101-1111

Cyclone IV devices receive configuration data on DATA[0] and the clock is received on DCLK. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high and the device enters initialization state.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

INIT_DONE is released and pulled high when initialization is complete. The external host device must be able to detect this low-to-high transition which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

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. The second device in the chain begins configuration in one clock cycle. Therefore, the transfer of data destinations is transparent to the external host device. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that DCLK and DATA lines are buffered. All devices initialize and enter user mode at the same time because all CONF_DONE pins are tied together.

If any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain because all nSTATUS and CONF_DONE pins are tied together. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

You can have multiple devices that contain the same configuration data in your system. To support this configuration scheme, all device nCE inputs are tied to GND, while the nCEO pins are left floating. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered. Devices must be of the same density and package. All devices start and complete configuration at the same time.

Figure 8–15 shows a multi-device PS configuration when both Cyclone IV devices are receiving the same configuration data.



Figure 8-15. Multi-Device PS Configuration When Both Devices Receive the Same Data

Notes to Figure 8-15:

- (1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.
- (2) The nCEO pins of both devices are left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.
- (3) 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.
- (4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA [0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

JTAG instructions have precedence over any other configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration in Cyclone IV devices during PS configuration, PS configuration terminates and JTAG configuration begins. If the MSEL pins are set to AS mode, the Cyclone IV device does not output a DCLK signal when JTAG configuration takes place.

The four required pins for a device operating in JTAG mode are TDI, TDO, TMS, and TCK. All the JTAG input pins are powered by the V_{CCIO} pin and support the LVTTL I/O standard only. All user I/O pins are tri-stated during JTAG configuration. Table 8-14 explains the function of each JTAG pin.

Pin Name Pin Type Description Serial input pin for instructions as well as test and programming data. Data shifts in on the Test data rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry is TDI disabled by connecting this pin to V_{CC} . TDI pin has weak internal pull-up resistors (typically 25 input kΩ). Serial data output pin for instructions as well as test and programming data. Data shifts out on Test data the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the TDO output JTAG interface is not required on the board, the JTAG circuitry is disabled by leaving this pin unconnected. Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions in the state machine occur on the rising edge of TCK. Therefore, Test mode TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. TMS select If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC} . TMS pin has weak internal pull-up resistors (typically 25 k Ω). The clock input to the BST circuitry. Some operations occur at the rising edge, while others Test clock occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry TCK input

Table 8–14. Dedicated JTAG Pins

You can download data to the device through the USB-Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV download cable, or the EthernetBlaster communications cable during JTAG configuration. Configuring devices with a cable is similar to programming devices in-system. Figure 8-23 and Figure 8-24 show the JTAG configuration of a single Cyclone IV device.

is disabled by connecting this pin to GND. The TCK pin has an internal weak pull-down resistor.

If you configure a master device with an SFL design, the master device enters user mode even though the slave devices in the multiple device chain are not being configured. The master device enters user mode with a SFL design even though the CONF_DONE signal is externally held low by the other slave devices in chain. Figure 8–29 shows the JTAG configuration of a single Cyclone IV device with a SFL design.





Notes to Figure 8-29:

- (1) Connect the pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.
- (2) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS 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.
- (3) Pin 6 of the header is a V₁₀ reference voltage for the MasterBlaster output driver. The V₁₀ 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) You must connect the nCE pin to GND or driven low for successful JTAG configuration.
- (5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.
- (6) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5- V V_{CCA} supply. 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*.
- (7) Connect the series resistor at the near end of the serial configuration device.
- (8) 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.
- (9) Resistor value can vary from 1 k Ω to 10 k Ω .
- (10) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

ISP of the Configuration Device

In the second stage, the SFL design in the master device allows you to write the configuration data for the device chain into the serial configuration device with the Cyclone IV device JTAG interface. The JTAG interface sends the programming data for the serial configuration device to the Cyclone IV device first. The Cyclone IV device then uses the ASMI pins to send the data to the serial configuration device.

Figure 10–2 shows the Cyclone IV GX HSSI receiver BSC.





To For more information about Cyclone IV devices user I/O boundary-scan cells, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

BST Operation Control

Table 10–1 lists the boundary-scan register length for Cyclone IV devices.

Table 10–1. Boundary-Scan Register Length for Cyclone IV Devices (Part 1 of 2)

Device	Boundary-Scan Register Length
EP4CE6	603
EP4CE10	603
EP4CE15	1080
EP4CE22	732
EP4CE30	1632
EP4CE40	1632
EP4CE55	1164
EP4CE75	1314
EP4CE115	1620
EP4CGX15	260
EP4CGX22	494
EP4CGX30 ⁽¹⁾	494
EP4CGX50	1006

The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration. The weak pull up resistors are not enabled prior to POR.

A possible concern for semiconductor devices in general regarding hot socketing is the potential for latch up. Latch up can occur when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins may be connected and driven by the active system before the power supply can provide current to the V_{CC} of the device and ground planes. This condition can lead to latch up and cause a low-impedance path from V_{CC} to GND in the device. As a result, the device extends a large amount of current, possibly causing electrical damage.

The design of the I/O buffers and hot-socketing circuitry ensures that Cyclone IV devices are immune to latch up during hot-socketing.

Hot-socketing Feature Implementation

The hot-socketing circuit does not include the CONF_DONE, nCEO, and nSTATUS pins to ensure that they are able to operate during configuration. The expected behavior for these pins is to drive out during power-up and power-down sequences.

Altera uses GND as reference for hot-socketing operation and I/O buffer designs. To ensure proper operation, Altera recommends connecting the GND between boards before connecting the power supplies. This prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND can otherwise cause an out-of-specification I/O voltage or current condition with the Altera device.

Power-On Reset Circuitry

Cyclone IV devices contain POR circuitry to keep the device in a reset state until the power supply voltage levels have stabilized during power up. During POR, all user I/O pins are tri-stated until the power supplies reach the recommended operating levels. In addition, the POR circuitry also ensures the V_{CCIO} level of I/O banks that contain configuration pins reach an acceptable level before configuration is triggered.

The POR circuit of the Cyclone IV device monitors the V_{CCINT}, V_{CCA}, and V_{CCIO} that contain configuration pins during power-on. You can power up or power down the V_{CCINT}, V_{CCA}, and V_{CCIO} pins in any sequence. The V_{CCINT}, V_{CCA}, and V_{CCIO} must have a monotonic rise to their steady state levels. All V_{CCA} pins must be powered to 2.5V (even when PLLs are not used), and must be powered up and powered down at the same time.

After the Cyclone IV device enters the user mode, the POR circuit continues to monitor the V_{CCINT} and V_{CCA} pins so that a brown-out condition during user mode is detected. If the V_{CCINT} or V_{CCA} voltage sags below the POR trip point during user mode, the POR circuit resets the device. If the V_{CCIO} voltage sags during user mode, the POR circuit does not reset the device.

[•] For more information about the hot-socketing specification, refer to the *Cyclone IV Device Datasheet* chapter and the *Hot-Socketing and Power-Sequencing Feature and Testing for Altera Devices* white paper.

After updating the word boundary, word aligner status signals (rx_syncstatus and rx_patterndetect) are driven high for one parallel clock cycle synchronous to the most significant byte of the word alignment pattern. The rx_syncstatus and rx_patterndetect signals have the same latency as the datapath and are forwarded to the FPGA fabric to indicate the word aligner status. Any word alignment pattern received thereafter in the same word boundary causes only the rx_patterndetect signal to go high for one clock cycle.

Figure 1–17 shows the manual alignment mode word aligner operation in 10-bit data width mode. In this example, a /K28.5/ (10'b0101111100) is specified as the word alignment pattern.

The word aligner aligns to the /K28.5/ alignment pattern (red) in cycle *n* because the rx_enapatternalign signal is asserted high. The rx_syncstatus signal goes high for one clock cycle indicating alignment to a new word boundary. The rx_patterndetect signal also goes high for one clock cycle to indicate initial word alignment.

At time n + 1, the rx_enapatternalign signal is deasserted to instruct the word aligner to lock the current word boundary.

The alignment pattern is detected again (green) in a new word boundary across cycles n + 2 and n + 3. The word aligner does not align to this new word boundary because the rx_enapatternalign signal is held low.

The /K28.5/ word alignment pattern is detected again (blue) in the current word boundary during cycle n + 5 causing the rx_patterndetect signal to go high for one parallel clock cycle.



Figure 1–17. Word Aligner in 10-bit Manual Alignment Mode

If the word alignment pattern is known to be unique and does not appear between word boundaries, you can hold the rx_enapatternalign signal constantly high because there is no possibility of false word alignment. If there is a possibility of the word alignment pattern occurring across word boundaries, you must control the rx_enapatternalign signal to lock the word boundary after the desired word alignment is achieved to avoid re-alignment to an incorrect word boundary.

/																	`,
Functional Mode							Bas	sic (10-Bit	PMA-PCS	Interfac	ce Width)						
> [•••••								
Channel Bonding									×1, ×2, >	<4							
}																	·
Low-Latency PCS							Disable	d								Enal	bled
) - [,														
Word Aligner (Pattern Length)	Ν	Manual A (7-Bit,	lignment 10-Bit)			Bit (7-Bit,	Slip 10-Bit)]		A Sta	utomatic Syn ate Machine	chronizatio (7-Bit, 10-E	n Bit)			Disa	oled
· ·																	·
8B/10B Encoder/Decoder	Disabl	ed	Enal	bled	Disa	bled	Ena	abled	Disa	abled			Enabled	1		Disa	bled
) 								L		L		_			1		, ,
Rate Match FIFO	Disabl	ed	Disa	bled	Disa	bled	Disa	abled	Disa	abled	D	isabled		Ena	bled	Disa	oled
·	<u> </u>		<u>1</u>					<u></u>		<u>+</u>			L			<u> </u>	
Byte SERDES	Disabled E	nabled	Disabled	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled	Enable	ed Disabled	Ena	bled	Disabled	Enabled	Disabled	Enabled
}		[{
Data Rate (Gbps)	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.6-	0.	6-	0.6-	0.6-	0.6-	0.6-
	0.6-	0.6- 3.125	0.6-	0.6- 3.125	0.6-	0.6- 3.125	0.6-	0.6- 3.125	0.6-	0.6-	0.6-	0.	.6- 125	0.6-	0.6- 3.125	0.6-	0.6- 3.125
· 							•••••	····									{
Byte Ordering	Disabled Di	isabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable	ed Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
EPGA Eabria-to-				····	····				····	····	····						
Transceiver	10-Bit	20-Bit	8-Bit	16-Bit	10-Bit	20-Bit	8-Bit	16-Bit	10-Bit	20-Bit	t 8-Bit	16-Bit	16-Bit	8-Bit	16-Bit	∎ 10-Bit	20-Bit
FPGA Fabric-to-	60-	30-	60-	30-	60-	30-	60-	30-	60-	30-	60-	30-	30-	60-	30-	60-	30-
Transceiver Interface	125 60-	125 30-	125 60-	125 30-	125 60-	125 30-	125 60-	125 30-	125 60-	125 30-	125 60-	125 30-	125 30-	125 60-	125 30-	125 60-	125 30-
Fredquency (MHz)	156.25	156.25	156.25	156.25	156.25	156.25	156.25	156.25	156.25	156.2	5 156.25	156.25	156.25	156.25	156.25	156.25	156.25
				Applica F324 a	able for d and small	evices in er packa	ı Iges			Applic F484	cable for de and larger	vices in packages					
						-	-				-						

Figure 1-46. Transceiver Configurations in Basic Mode with a 10-Bit Wide PMA-to-PCS Interface

Rate Match FIFO Operation in Basic Mode

In Basic mode, the rate match FIFO performs the following operations:

- Deletes a maximum of four skip patterns from a cluster, if there is one skip pattern left in the cluster after deletion
- Insert a maximum of four skip patterns in a cluster, if there are less than five skip patterns in the cluster after deletion
- Automatically deletes the data byte that causes the FIFO to go full and asserts the rx_rmfifofull flag synchronous to the subsequent data byte
- Automatically inserts /K30.7/ (9'h1FE) after the data byte that causes the FIFO to go empty and asserts the rx-fifoempty flag synchronous to the inserted /K30.7/ (9'h1FE)

Additional Options in Basic Mode

In Basic mode, the transceiver supports the following additional options:

low-latency PCS operation

PIPE Interface

The PIPE interface provides a standard interface between the PCIe-compliant PHY and MAC layer as defined by the version 2.00 of the PIPE Architecture specification for Gen1 (2.5 Gbps) signaling rate. Any core or IP implementing the PHY MAC, data link, and transaction layers that supports PIPE 2.00 can be connected to the Cyclone IV GX transceiver configured in PIPE mode. Table 1–15 lists the PIPE-specific ports available from the Cyclone IV GX transceiver configured in PIPE mode and the corresponding port names in the PIPE 2.00 specification.

Transceiver Port Name	PIPE 2.00 Port Name						
tx_datain[150] ⁽¹⁾	TxData[150]						
<pre>tx_ctrlenable[10] (1)</pre>	TxDataK[10]						
rx_dataout[150] ⁽¹⁾	RxData[150]						
rx_ctrldetect[10] ⁽¹⁾	RxDataK[10]						
tx_detectrxloop	TxDetectRx/Loopback						
tx_forceelecidle	TxElecIdle						
tx_forcedispcompliance	TxCompliance						
pipe8b10binvpolarity	RxPolarity						
powerdn[10] ⁽²⁾	PowerDown[10]						
pipedatavalid	RxValid						
pipephydonestatus	PhyStatus						
pipeelecidle	RxElecIdle						
pipestatus	RxStatus[20]						

Table 1–15. Transceiver-FPGA Fabric Interface Ports in PIPE Mode

Notes to Table 1-15:

(1) When used with PCIe hard IP block, the byte SERDES is not used. In this case, the data ports are 8 bits wide and control identifier is 1 bit wide.

(2) Cyclone IV GX transceivers do not implement power saving measures in lower power states (P0s, P1, and P2), except when putting the transmitter buffer in electrical idle in the lower power states.

Receiver Detection Circuitry

In PIPE mode, the transmitter supports receiver detection function with a built-in circuitry in the transmitter PMA. The PCIe protocol requires the transmitter to detect if a receiver is present at the far end of each lane as part of the link training and synchronization state machine sequence. This feature requires the following conditions:

- transmitter output buffer to be tri-stated
- have OCT utilization
- 125 MHz clock on the fixedclk port

The circuit works by sending a pulse on the common mode of the transmitter. If an active PCIe receiver is present at the far end, the time constant of the step voltage on the trace is higher compared to when the receiver is not present. The circuitry monitors the time constant of the step signal seen on the trace to decide if a receiver was detected.

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

- 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.





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.

1. Cyclone IV Device Datasheet

This chapter describes the electrical and switching characteristics for Cyclone[®] IV devices. Electrical characteristics include operating conditions and power consumption. Switching characteristics include transceiver specifications, core, and periphery performance. This chapter also describes I/O timing, including programmable I/O element (IOE) delay and programmable output buffer delay.

This chapter includes the following sections:

- "Operating Conditions" on page 1–1
- "Power Consumption" on page 1–16
- "Switching Characteristics" on page 1–16
- "I/O Timing" on page 1–37
- "Glossary" on page 1–37

Operating Conditions

When Cyclone IV devices are implemented in a system, they are rated according to a set of defined parameters. To maintain the highest possible performance and reliability of Cyclone IV devices, you must consider the operating requirements described in this chapter.

Cyclone IV devices are offered in commercial, industrial, extended industrial and, automotive grades. Cyclone IV E devices offer –6 (fastest), –7, –8, –8L, and –9L speed grades for commercial devices, –8L speed grades for industrial devices, and –7 speed grade for extended industrial and automotive devices. Cyclone IV GX devices offer –6 (fastest), –7, and –8 speed grades for commercial devices and –7 speed grade for industrial devices.



• For more information about the supported speed grades for respective Cyclone IV devices, refer to the *Cyclone IV FPGA Device Family Overview* chapter.

Cyclone IV E devices are offered in core voltages of 1.0 and 1.2 V. Cyclone IV E devices with a core voltage of 1.0 V have an 'L' prefix attached to the speed grade.

In this chapter, a prefix associated with the operating temperature range is attached to the speed grades; commercial with a "C" prefix, industrial with an "I" prefix, and automotive with an "A" prefix. Therefore, commercial devices are indicated as C6, C7, C8, C8L, or C9L per respective speed grade. Industrial devices are indicated as I7, I8, or I8L. Automotive devices are indicated as A7.

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





Table 1-46. Glossary (Part 3 of 5)

Letter	Term	Definitions											
	RL	Receiver differential input discrete resistor (external to Cyclone IV devices).											
		Receiver input waveform for LVDS and LVPECL differential standards:											
		Single-Ended Waveform											
		Positive Channel (p) = V _{IH}											
		Variative Channel (n) = V _{IL}											
	Deschargheret	Ground											
_	Waveform												
R	Wateren												
		Differential Waveform (Mathematical Function of Positive & Negative Channel)											
		V _{ID}											
		p-n											
	Receiver input	High-speed I/O block: The total margin left after accounting for the sampling window and TCCS.											
	(RSKM)	RSKM = (TUI - SW - TCCS) / 2.											
S	Single-ended voltage- referenced I/O Standard SW (Sampling Window)												