

Welcome to [E-XFL.COM](https://www.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	645
Number of Logic Elements/Cells	10320
Total RAM Bits	423936
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/ep4ce10e22c6n

 For more information, refer to the *External Memory Interfaces in Cyclone IV Devices* chapter.

Configuration

Cyclone IV devices use SRAM cells to store configuration data. Configuration data is downloaded to the Cyclone IV device each time the device powers up. Low-cost configuration options include the Altera EPCS family serial flash devices and commodity parallel flash configuration options. These options provide the flexibility for general-purpose applications and the ability to meet specific configuration and wake-up time requirements of the applications.

Table 1-9 lists which configuration schemes are supported by Cyclone IV devices.

Table 1-9. Configuration Schemes for Cyclone IV Device Family

Devices	Supported Configuration Scheme
Cyclone IV GX	AS, PS, JTAG, and FPP ⁽¹⁾
Cyclone IV E	AS, AP, PS, FPP, and JTAG

Note to Table 1-9:

(1) The FPP configuration scheme is only supported by the EP4CGX30F484 and EP4CGX50/75/110/150 devices.

IEEE 1149.6 (AC JTAG) is supported on all transceiver I/O pins. All other pins support IEEE 1149.1 (JTAG) for boundary scan testing.

 For more information, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

For Cyclone IV GX devices to meet the PCIe 100 ms wake-up time requirement, you must use passive serial (PS) configuration mode for the EP4CGX15/22/30 devices and use fast passive parallel (FPP) configuration mode for the EP4CGX30F484 and EP4CGX50/75/110/150 devices.

 For more information, refer to the *Configuration and Remote System Upgrades in Cyclone IV Devices* chapter.

The cyclical redundancy check (CRC) error detection feature during user mode is supported in all Cyclone IV GX devices. For Cyclone IV E devices, this feature is only supported for the devices with the core voltage of 1.2 V.

 For more information about CRC error detection, refer to the *SEU Mitigation in Cyclone IV Devices* chapter.

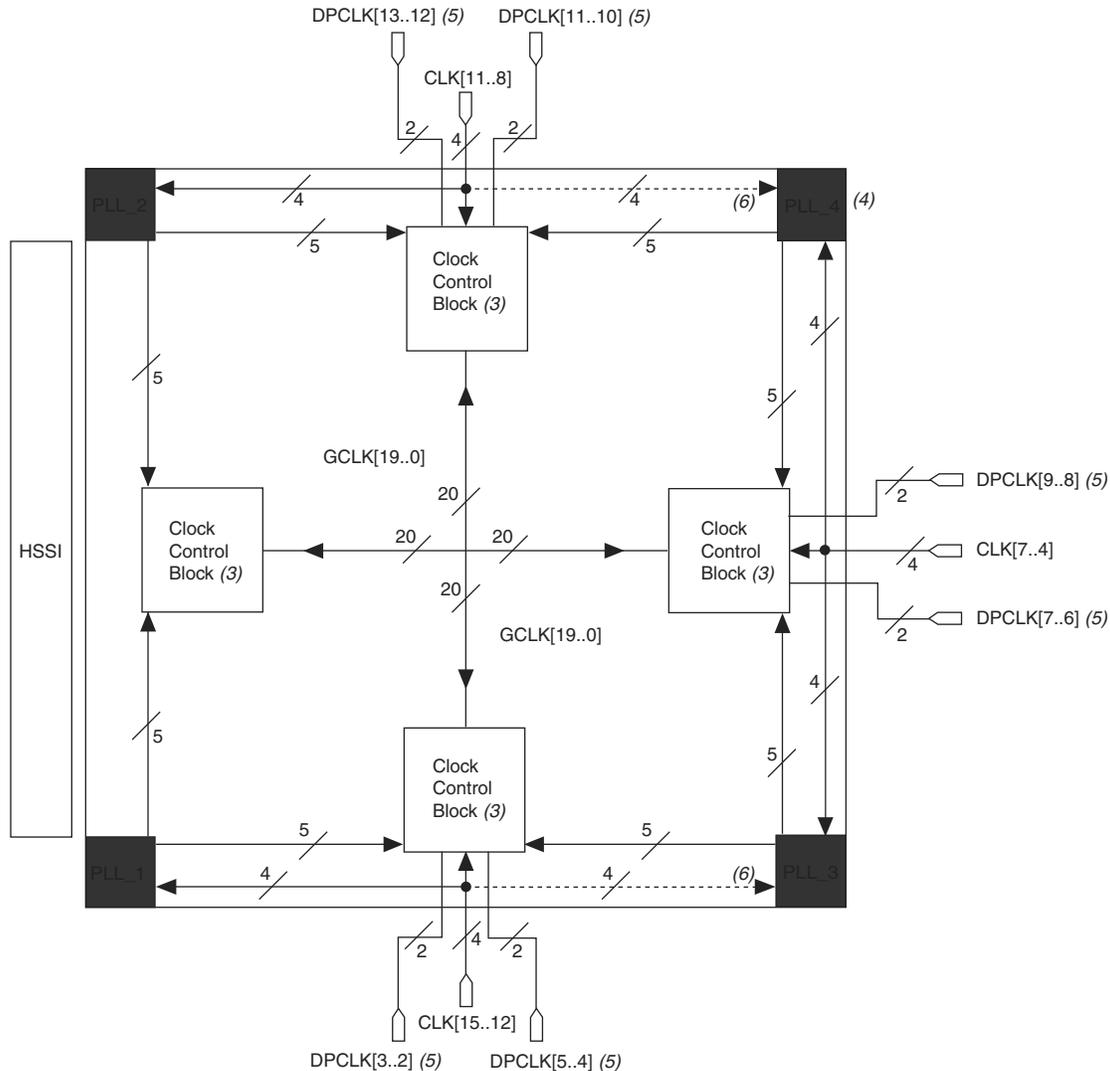
High-Speed Transceivers (Cyclone IV GX Devices Only)

Cyclone IV GX devices contain up to eight full duplex high-speed transceivers that can operate independently. These blocks support multiple industry-standard communication protocols, as well as Basic mode, which you can use to implement your own proprietary protocols. Each transceiver channel has its own pre-emphasis and equalization circuitry, which you can set at compile time to optimize signal integrity and reduce bit error rates. Transceiver blocks also support dynamic reconfiguration, allowing you to change data rates and protocols on-the-fly.

GCLK Network Clock Source Generation

Figure 5-2, Figure 5-3, and Figure 5-4 on page 5-14 show the Cyclone IV PLLs, clock inputs, and clock control block location for different Cyclone IV device densities.

Figure 5-2. Clock Networks and Clock Control Block Locations in EP4CGX15, EP4CGX22, and EP4CGX30 Devices (1), (2)

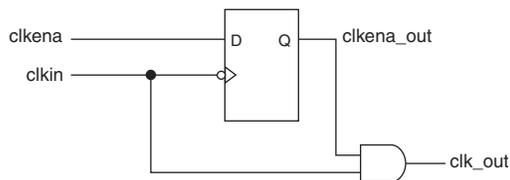


Notes to Figure 5-2:

- (1) The clock networks and clock control block locations apply to all EP4CGX15, EP4CGX22, and EP4CGX30 devices except EP4CGX30 device in F484 package.
- (2) PLL_1 and PLL_2 are multipurpose PLLs while PLL_3 and PLL_4 are general purpose PLLs.
- (3) There are five clock control blocks on each side.
- (4) PLL_4 is only available in EP4CGX22 and EP4CGX30 devices in F324 package.
- (5) The EP4CGX15 device has two DPCLK pins on three sides of the device: DPCLK2 and DPCLK5 on bottom side, DPCLK7 and DPCLK8 on the right side, DPCLK10 and DPCLK13 on the top side of device.
- (6) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.

Figure 5-7 shows how to implement the `clkena` signal with a single register.

Figure 5-7. `clkena` Implementation

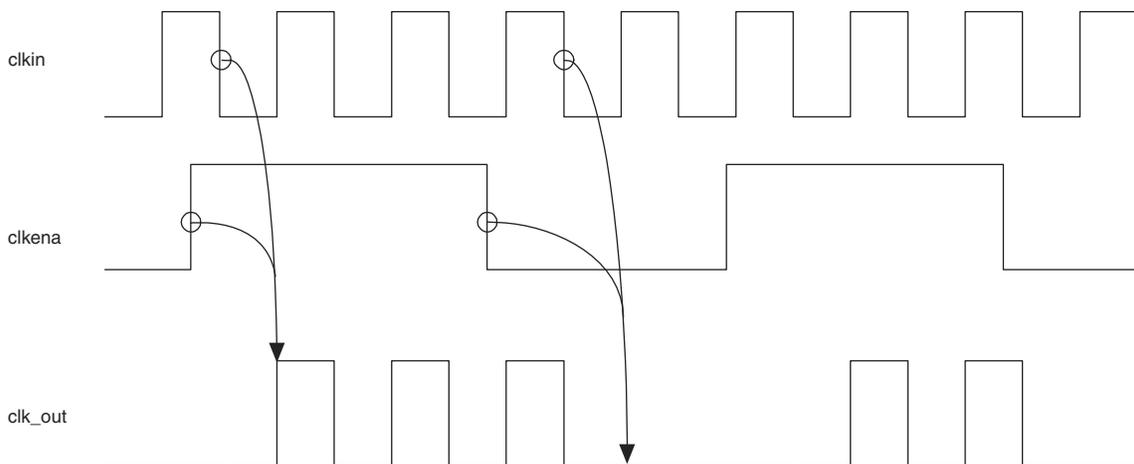


 The `clkena` circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5-7.

Figure 5-8 shows the waveform example for a clock output enable. The `clkena` signal is sampled on the falling edge of the clock (`clkina`).

 This feature is useful for applications that require low power or sleep mode.

Figure 5-8. `clkena` Implementation: Output Enable



The `clkena` signal can also disable clock outputs if the system is not tolerant to frequency overshoot during PLL resynchronization.

Altera recommends using the `clkena` signals when switching the clock source to the PLLs or the GCLK. The recommended sequence is:

1. Disable the primary output clock by de-asserting the `clkena` signal.
2. Switch to the secondary clock using the dynamic select signals of the clock control block.
3. Allow some clock cycles of the secondary clock to pass before reasserting the `clkena` signal. The exact number of clock cycles you must wait before enabling the secondary clock is design-dependent. You can build custom logic to ensure glitch-free transition when switching between different clock sources.

Table 6-2 on page 6-7 shows the possible settings for I/O standards with current strength control. These programmable current strength settings are a valuable tool in helping decrease the effects of simultaneously switching outputs (SSO) in conjunction with reducing system noise. The supported settings ensure that the device driver meets the specifications for I_{OH} and I_{OL} of the corresponding I/O standard.

 When you use programmable current strength, on-chip series termination (R_S OCT) is not available.

Slew Rate Control

The output buffer for each Cyclone IV I/O pin provides optional programmable output slew-rate control. Table 6-2 on page 6-7 shows the possible slew rate option and the Quartus II default slew rate setting. However, these fast transitions may introduce noise transients in the system. A slower slew rate reduces system noise, but adds a nominal delay to rising and falling edges. Because each I/O pin has an individual slew-rate control, you can specify the slew rate on a pin-by-pin basis. The slew-rate control affects both the rising and falling edges. Slew rate control is available for single-ended I/O standards with current strength of 8 mA or higher.

 You cannot use the programmable slew rate feature when using OCT with calibration.

 You cannot use the programmable slew rate feature when using the 3.0-V PCI, 3.0-V PCI-X, 3.3-V LVTTTL, or 3.3-V LVCMOS I/O standards. Only the fast slew rate (default) setting is available.

Open-Drain Output

Cyclone IV devices provide an optional open-drain (equivalent to an open-collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (for example, interrupt and write enable signals) that are asserted by multiple devices in your system.

Bus Hold

Each Cyclone IV device user I/O pin provides an optional bus-hold feature. The bus-hold circuitry holds the signal on an I/O pin at its last-driven state. Because the bus-hold feature holds the last-driven state of the pin until the next input signal is present, an external pull-up or pull-down resistor is not necessary to hold a signal level when the bus is tri-stated.

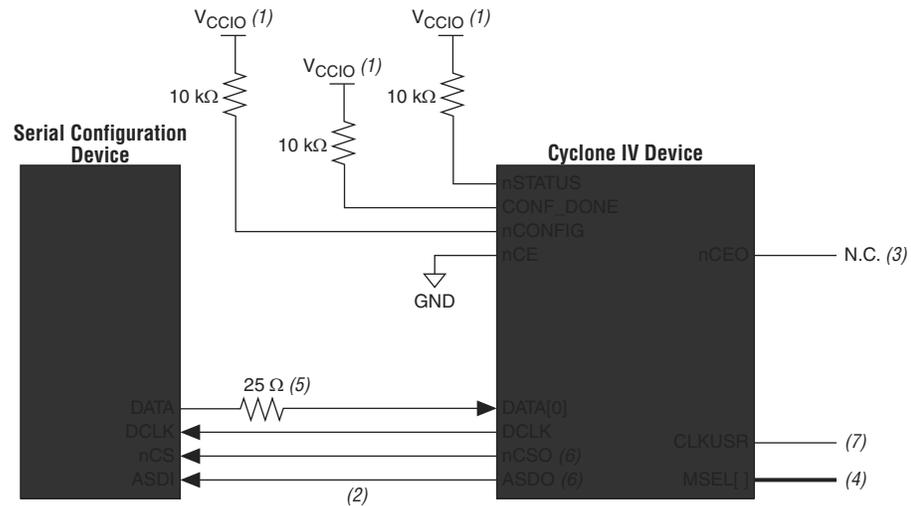
The bus-hold circuitry also pulls undriven pins away from the input threshold voltage in which noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent overdriving signals.

 If you enable the bus-hold feature, the device cannot use the programmable pull-up option. Disable the bus-hold feature when the I/O pin is configured for differential signals. Bus-hold circuitry is not available on dedicated clock pins.

Bus-hold circuitry is only active after configuration. When going into user mode, the bus-hold circuit captures the value on the pin present at the end of configuration.

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

Figure 8-2. Single-Device AS Configuration



Notes to Figure 8-2:

- (1) Connect the pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.
- (2) Cyclone IV devices use the ASDO-to-ASDI path to control the configuration device.
- (3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.
- (4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8-3 on page 8-8, Table 8-4 on page 8-8, and Table 8-5 on page 8-9. Connect the MSEL pins directly to V_{CCA} or GND.
- (5) Connect the series resistor at the near end of the serial configuration device.
- (6) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as the DATA[1] pin in AP and FPP modes.
- (7) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

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

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

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

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

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 LVTTTL 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.

Table 8-14. Dedicated JTAG Pins

Pin Name	Pin Type	Description
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data shifts in on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V _{CC} . TDI pin has weak internal pull-up resistors (typically 25 kΩ).
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data shifts out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by leaving this pin unconnected.
TMS	Test mode select	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, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. 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 kΩ).
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to GND. The TCK pin has an internal weak pull-down resistor.

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.

 WYSIWYG is an optimization technique that performs optimization on a VQM (Verilog Quartus Mapping) netlist in the Quartus II software.

Error Detection Block

Table 9-3 lists the types of CRC detection to check the configuration bits.

Table 9-3. Types of CRC Detection to Check the Configuration Bits

First Type of CRC Detection	Second Type of CRC Detection
<ul style="list-style-type: none"> ■ CRAM error checking ability (32-bit CRC) during user mode, for use by the CRC_ERROR pin. ■ There is only one 32-bit CRC value. This value covers all the CRAM data. 	<ul style="list-style-type: none"> ■ 16-bit CRC embedded in every configuration data frame. ■ During configuration, after a frame of data is loaded into the device, the pre-computed CRC is shifted into the CRC circuitry. ■ Simultaneously, the CRC value for the data frame shifted-in is calculated. If the pre-computed CRC and calculated CRC values do not match, nSTATUS is set low. ■ Every data frame has a 16-bit CRC. Therefore, there are many 16-bit CRC values for the whole configuration bit stream. ■ Every device has a different length of configuration data frame.

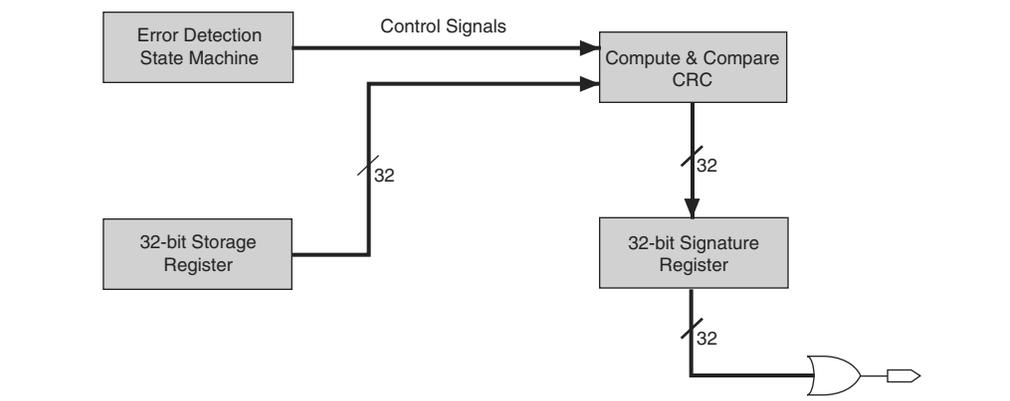
This section focuses on the first type—the 32-bit CRC when the device is in user mode.

Error Detection Registers

There are two sets of 32-bit registers in the error detection circuitry that store the computed CRC signature and pre-calculated CRC value. A non-zero value on the signature register causes the CRC_ERROR pin to set high.

Figure 9-1 shows the block diagram of the error detection block and the two related 32-bit registers: the signature register and the storage register.

Figure 9-1. Error Detection Block Diagram

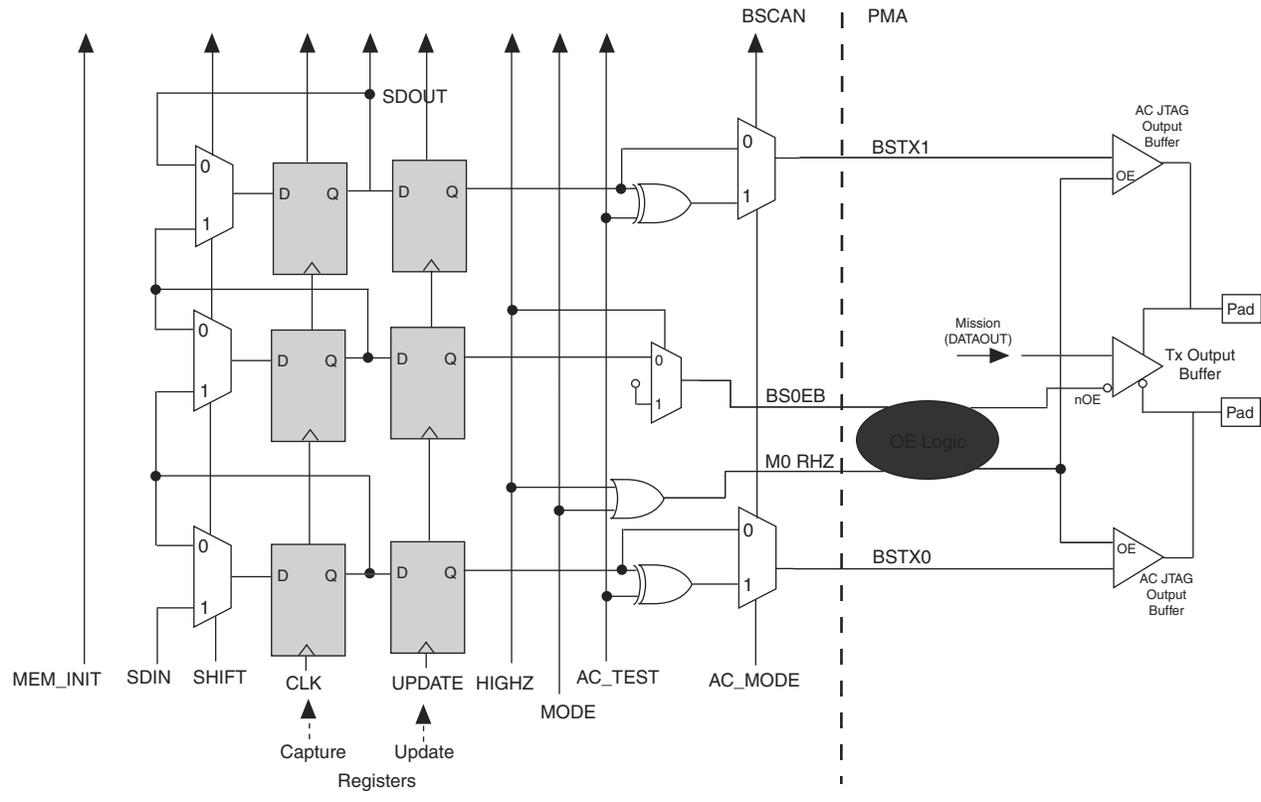


IEEE Std. 1149.6 Boundary-Scan Register

The boundary-scan cell (BSC) for HSSI transmitters (GXB_TX[p, n]) and receivers (GXB_RX[p, n]) in Cyclone IV GX devices are different from the BSCs for I/O pins.

Figure 10-1 shows the Cyclone IV GX HSSI transmitter boundary-scan cell.

Figure 10-1. HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Cyclone IV GX Devices



Section I. Transceivers

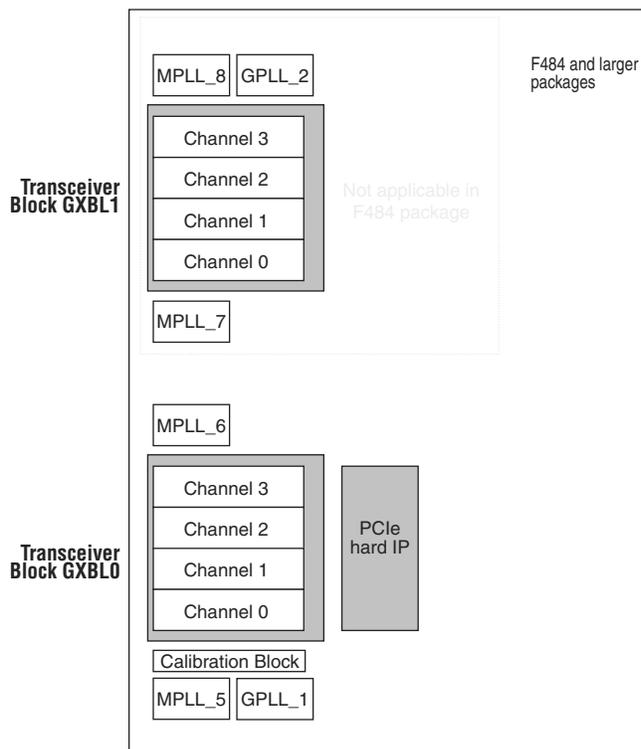
This section provides a complete overview of all features relating to the Cyclone[®] IV device transceivers. This section includes the following chapters:

- Chapter 1, Cyclone IV Transceivers Architecture
- Chapter 2, Cyclone IV Reset Control and Power Down
- Chapter 3, Cyclone IV Dynamic Reconfiguration

Revision History

Refer to the chapter for its own specific revision history. For information about when the chapter was updated, refer to the Chapter Revision Dates section, which appears in the complete handbook.

Figure 1–2. F484 and Larger Packages with Transceiver Channels for Cyclone IV GX Devices



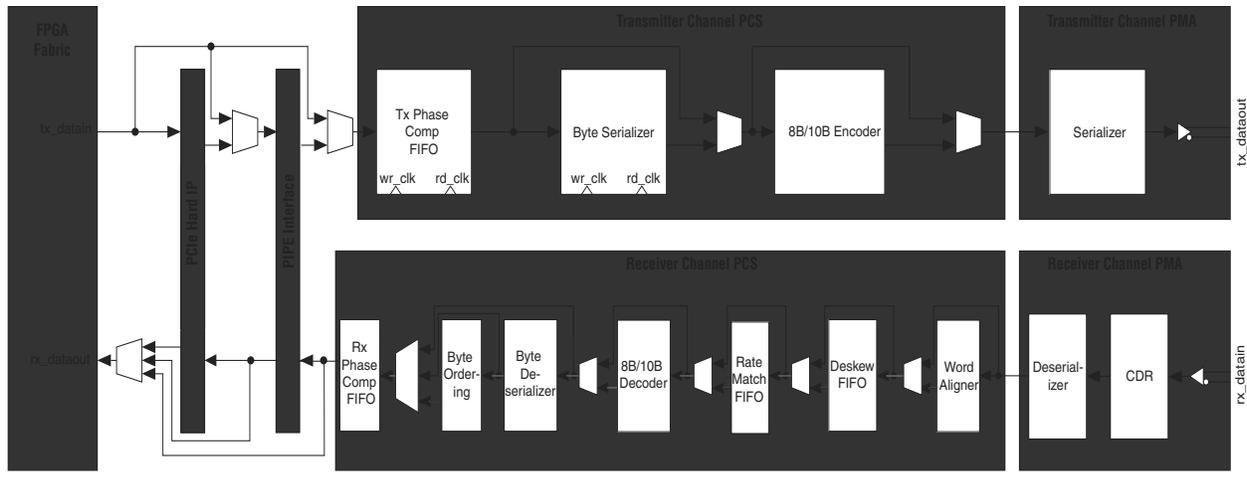
For more information about the transceiver architecture, refer to the following sections:

- “Architectural Overview” on page 1–4
- “Transmitter Channel Datapath” on page 1–5
- “Receiver Channel Datapath” on page 1–11
- “Transceiver Clocking Architecture” on page 1–26
- “Transceiver Channel Datapath Clocking” on page 1–29
- “FPGA Fabric-Transceiver Interface Clocking” on page 1–43
- “Calibration Block” on page 1–45
- “PCI-Express Hard IP Block” on page 1–46

Architectural Overview

Figure 1-3 shows the Cyclone IV GX transceiver channel datapath.

Figure 1-3. Transceiver Channel Datapath for Cyclone IV GX Devices



Each transceiver channel consists of a transmitter and a receiver datapath. Each datapath is further structured into the following:

- Physical media attachment (PMA)—includes analog circuitry for I/O buffers, clock data recovery (CDR), serializer/deserializer (SERDES), and programmable pre-emphasis and equalization to optimize serial data channel performance.
- Physical coding sublayer (PCS)—includes hard logic implementation of digital functionality within the transceiver that is compliant with supported protocols.

Outbound parallel data from the FPGA fabric flows through the transmitter PCS and PMA, is transmitted as serial data. Received inbound serial data flows through the receiver PMA and PCS into the FPGA fabric. The transceiver supports the following interface widths:

- FPGA fabric-transceiver PCS—8, 10, 16, or 20 bits
- PMA-PCS—8 or 10 bits

- The transceiver channel interfaces through the PIPE when configured for PCIe protocol implementation. The PIPE is compliant with version 2.00 of the *PHY Interface for the PCI Express Architecture* specification.

Byte Deserializer

The byte deserializer halves the FPGA fabric-transceiver interface frequency while doubles the parallel data width to the FPGA fabric.

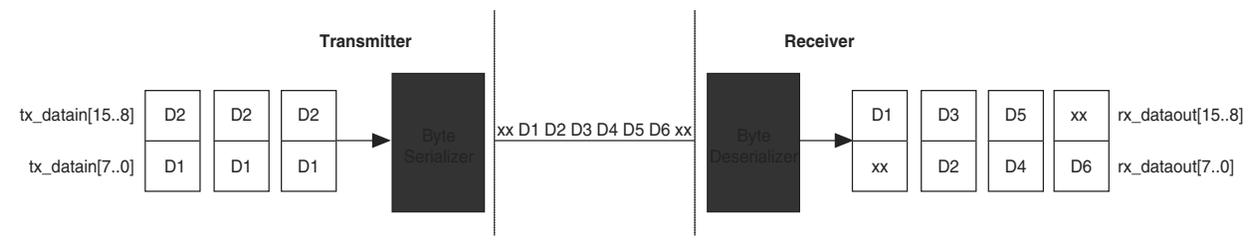
For example, when operating an EP4CGX150 receiver channel at 3.125 Gbps with deserialization factor of 10, the receiver PCS datapath runs at 312.5 MHz. The byte deserializer converts the 10-bit data at 312.5 MHz into 20-bit data at 156.25 MHz before forwarding the data to the FPGA fabric.

Byte Ordering

In the 16- or 20-bit FPGA fabric-transceiver interface, the byte deserializer receives one data byte (8 or 10 bits) and deserializes it into two data bytes (16 or 20 bits). Depending on when the receiver PCS logic comes out of reset, the byte ordering at the output of the byte deserializer may not match the original byte ordering of the transmitted data. The byte misalignment resulting from byte deserialization is unpredictable because it depends on which byte is being received by the byte deserializer when it comes out of reset.

Figure 1–23 shows a scenario where the most significant byte and the least significant byte of the two-byte transmitter data appears straddled across two word boundaries after the data is deserialized at the receiver.

Figure 1–23. Example of Byte Deserializer at the Receiver



The byte ordering block restores the proper byte ordering by performing the following actions:

- Look for the user-programmed byte ordering pattern in the byte-deserialized data
- Inserts a user-programmed pad byte if the user-programmed byte ordering pattern is found in the most significant byte position

You must select a byte ordering pattern that you know appears at the least significant byte position of the parallel transmitter data.

The byte ordering block is supported in the following receiver configurations:

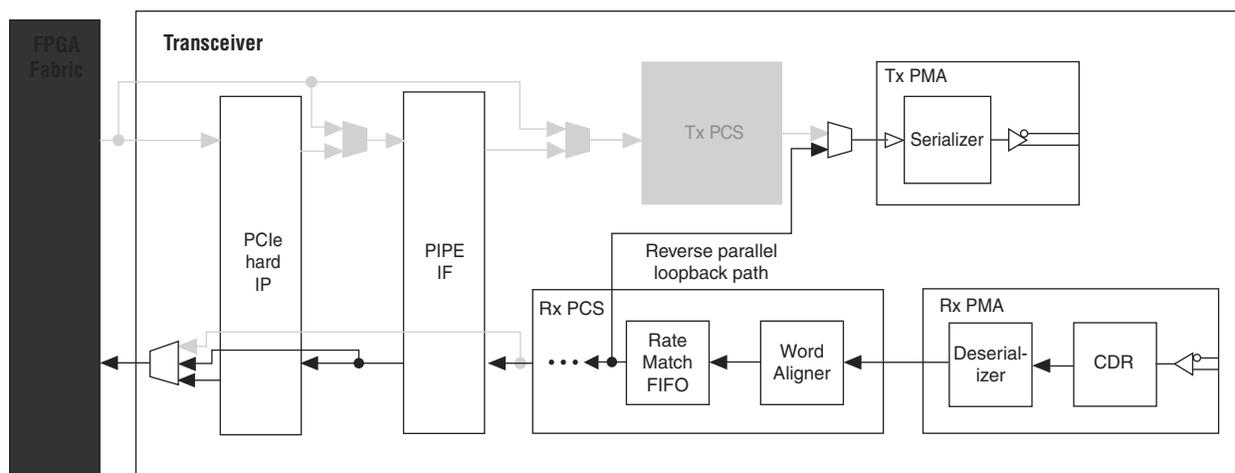
- 16-bit FPGA fabric-transceiver interface, 8B/10B disabled, and the word aligner in manual alignment mode. Program a custom 8-bit byte ordering pattern and 8-bit pad byte.
- 16-bit FPGA fabric-transceiver interface, 8B/10B enabled, and the word aligner in automatic synchronization state machine mode. Program a custom 9-bit byte ordering pattern and 9-bit pad byte. The MSB of the 9-bit byte ordering pattern and pad byte represents the control identifier of the 8B/10B decoded data.

Reverse Parallel Loopback

The reverse parallel loopback option is only available for PIPE mode. In this mode, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate match FIFO before looping back to the transmitter serializer and transmitted out through the TX buffer, as shown in Figure 1-70. The received data is also available to the FPGA fabric. This loopback mode is compliant with version 2.00 of the *PHY Interface for the PCI Express Architecture* specification.

To enable the reverse parallel loopback mode, assert the `tx_detectrxloopback` port in P0 power state.

Figure 1-70. PIPE Reverse Parallel Loopback Path ⁽¹⁾



Note to Figure 1-70:

(1) Grayed-Out Blocks are Not Active in this mode.

Serial Loopback

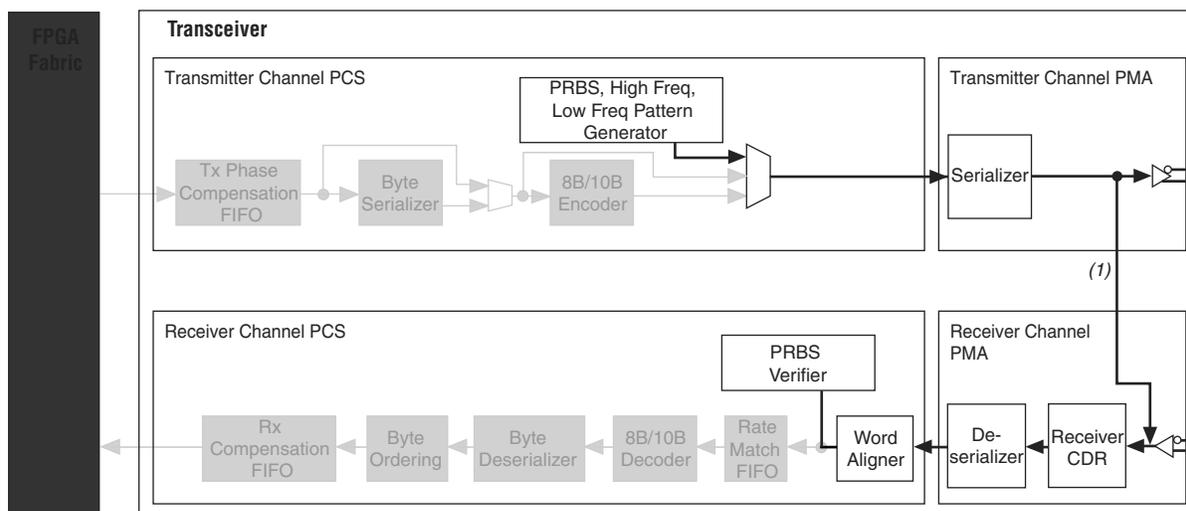
The serial loopback option is available for all functional modes except PIPE mode. In this mode, the data from the FPGA fabric passes through the transmitter channel and looped back to the receiver channel, bypassing the receiver buffer, as shown in Figure 1-71. The received data is available to the FPGA logic for verification. The receiver input buffer is not active in this mode. With this option, you can check the operation of all enabled PCS and PMA functional blocks in the transmitter and receiver channels.

The transmitter channel sends the data to both the serial output port and the receiver channel. The differential output voltage on the serial ports is based on the selected V_{OD} settings. The data is looped back to the receiver CDR and is retimed through different clock domains. You must provide an alignment pattern for the word aligner to enable the receiver channel to retrieve the byte boundary.

PRBS

Figure 1-74 shows the datapath for the PRBS, high and low frequency pattern test modes. The pattern generator is located in TX PCS before the serializer, and PRBS pattern verifier located in RX PCS after the word aligner.

Figure 1-74. PRBS Pattern Test Mode Datapath



Note to Figure 1-74:

(1) Serial loopback path is optional and can be enabled for the PRBS verifier to check the PRBS pattern

Table 1-25 lists the supported PRBS, high and low frequency patterns, and corresponding channel settings. The PRBS pattern repeats after completing an iteration. The number of bits a PRBS X pattern sends before repeating the pattern is $2^{(X-1)}$ bits.

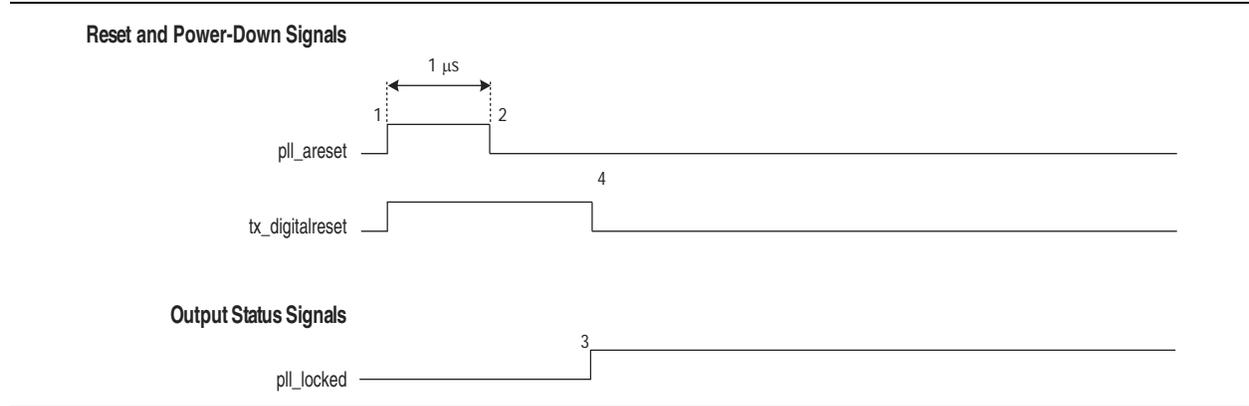
Table 1-25. PRBS, High and Low Frequency Patterns, and Channel Settings (Part 1 of 2)

Patterns	Polynomial	8-bit Channel Width				10-bit Channel Width			
		Channel Width of 8 bits (1)	Word Alignment Pattern	Maximum Data Rate (Gbps) for F324 and Smaller Packages	Maximum Data Rate (Gbps) for F484 and Larger Packages	Channel Width of 10-bits (1)	Word Alignment Pattern	Maximum Data Rate (Gbps) for F324 and Smaller Packages	Maximum Data Rate (Gbps) for F484 and Larger Packages
PRBS 7	$X^7 + X^6 + 1$	Y	16'h3040	2.0	2.5	N	—	—	—
PRBS 8	$X^8 + X^7 + 1$	Y	16'hFF5A	2.0	2.5	N	—	—	—
PRBS 10	$X^{10} + X^7 + 1$	N	—	—	—	Y	10'h3FF	2.5	3.125
PRBS 23	$X^{23} + X^{18} + 1$	Y	16'hFFFF	2.0	2.5	N	—	—	—
High frequency (2)	1010101010	Y	—	2.0	2.5	Y	—	2.5	3.125

Transmitter Only Channel

This configuration contains only a transmitter channel. If you create a **Transmitter Only** instance in the ALTGX MegaWizard Plug-In Manager in Basic $\times 4$ functional mode, use the reset sequence shown in Figure 2-3.

Figure 2-3. Sample Reset Sequence for Bonded and Non-Bonded Configuration Transmitter Only Channels



As shown in Figure 2-3, perform the following reset procedure for the **Transmitter Only** channel configuration:

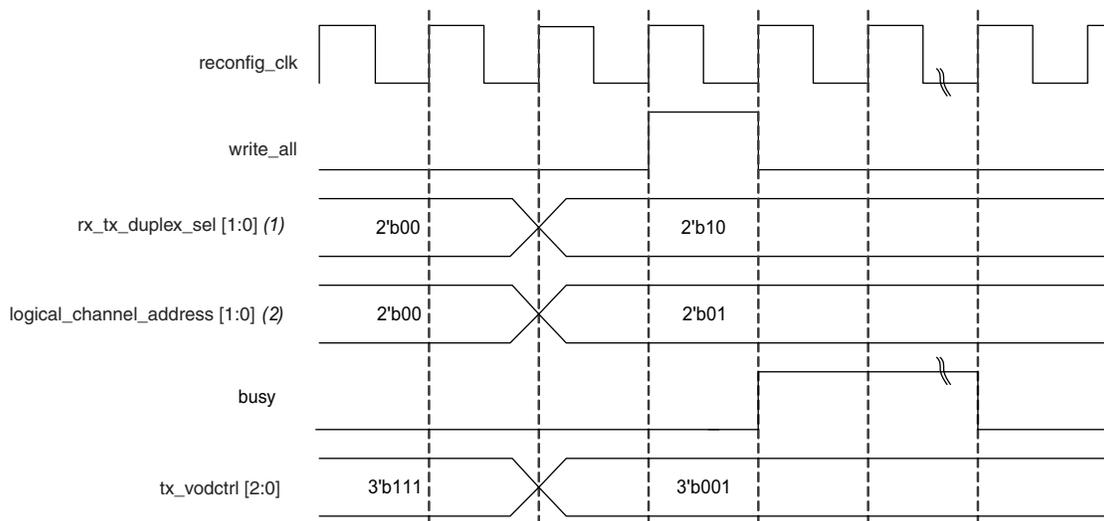
1. After power up, assert `pll_areset` for a minimum period of 1 μs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset` signal asserted during this time period. After you de-assert the `pll_areset` signal, the multipurpose PLL starts locking to the transmitter input reference clock.
3. When the multipurpose PLL locks, as indicated by the `pll_locked` signal going high (marker 3), de-assert the `tx_digitalreset` signal (marker 4). At this point, the transmitter is ready for transmitting data.

Table 3-2. Dynamic Reconfiguration Controller Port List (ALTGX_RECONFIG Instance) (Part 4 of 7)

Port Name	Input/ Output	Description																					
Analog Settings Control/Status Signals																							
<code>tx_vodctrl[2..0]</code> (1)	Input	<p>This is an optional transmit buffer V_{OD} control signal. It is 3 bits per transmitter channel. The number of settings varies based on the transmit buffer supply setting and the termination resistor setting on the TX Analog screen of the ALTGX MegaWizard Plug-In Manager.</p> <p>The width of this signal is fixed to 3 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 3 bits per channel.</p> <p>The following shows the V_{OD} values corresponding to the <code>tx_vodctrl</code> settings for 100-Ω termination.</p> <p>For more information, refer to the “Programmable Output Differential Voltage” section of the <i>Cyclone IV GX Device Datasheet</i> chapter.</p> <table border="1" data-bbox="542 764 1456 1073"> <thead> <tr> <th><code>tx_vodctrl[2:0]</code></th> <th>Corresponding ALTGX instance settings</th> <th>Corresponding V_{OD} settings (mV)</th> </tr> </thead> <tbody> <tr> <td>3'b001</td> <td>1</td> <td>400</td> </tr> <tr> <td>3'b010</td> <td>2</td> <td>600</td> </tr> <tr> <td>3'b011</td> <td>3</td> <td>800</td> </tr> <tr> <td>3'b111</td> <td>4 (2)</td> <td>900 (2)</td> </tr> <tr> <td>3'b100</td> <td>5</td> <td>1000</td> </tr> <tr> <td>3'b101</td> <td>6</td> <td>1200</td> </tr> </tbody> </table> <p>All other values => N/A</p>	<code>tx_vodctrl[2:0]</code>	Corresponding ALTGX instance settings	Corresponding V_{OD} settings (mV)	3'b001	1	400	3'b010	2	600	3'b011	3	800	3'b111	4 (2)	900 (2)	3'b100	5	1000	3'b101	6	1200
<code>tx_vodctrl[2:0]</code>	Corresponding ALTGX instance settings	Corresponding V_{OD} settings (mV)																					
3'b001	1	400																					
3'b010	2	600																					
3'b011	3	800																					
3'b111	4 (2)	900 (2)																					
3'b100	5	1000																					
3'b101	6	1200																					

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

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



Notes to Figure 3-4:

- (1) In this waveform example, you are writing to only the transmitter portion of the channel.
- (2) In this waveform example, the number of channels connected to the dynamic reconfiguration controller is four. Therefore, the logical_channel_address port is 2 bits wide.

Read Transaction

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

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

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

Table 3–5 describes the rx_dataoutfull[31..0] FPGA fabric-Transceiver channel interface signals.

Table 3–5. rx_dataoutfull[31..0] FPGA Fabric-Transceiver Channel Interface Signal Descriptions (Part 1 of 3)

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
8-bit FPGA fabric-Transceiver Channel Interface	The following signals are used in 8-bit 8B/10B modes:
	rx_dataoutfull[7:0]: 8-bit decoded data (rx_dataout)
	rx_dataoutfull[8]: Control bit (rx_ctrlldetect)
	rx_dataoutfull[9]: Code violation status signal (rx_errrdetect)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Disparity error status signal (rx_disperr)
	rx_dataoutfull[12]: Pattern detect status signal (rx_patterndetect)
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfifoatadeleted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfiostatinserted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14:13]: PCI Express (PIPE) functional mode (rx_pipestatus)
10-bit FPGA fabric-Transceiver Channel Interface	rx_dataoutfull[9:0]: 10-bit un-encoded data (rx_dataout)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: 8B/10B disparity error indicator (rx_disperr)
	rx_dataoutfull[12]: rx_patterndetect
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfiostatadeleted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfiostatinserted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[15]: 8B/10B running disparity indicator (rx_runningdisp)

