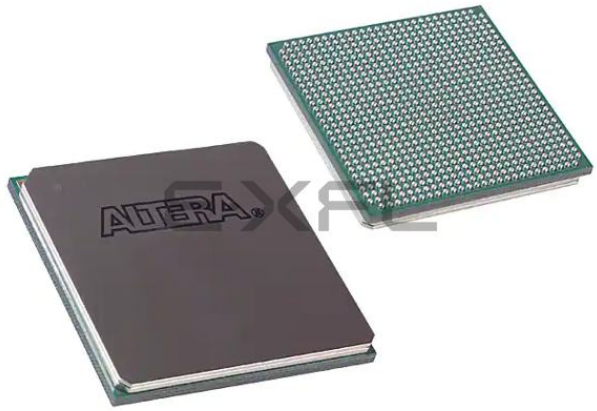



**Welcome to E-XFL.COM**

## Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

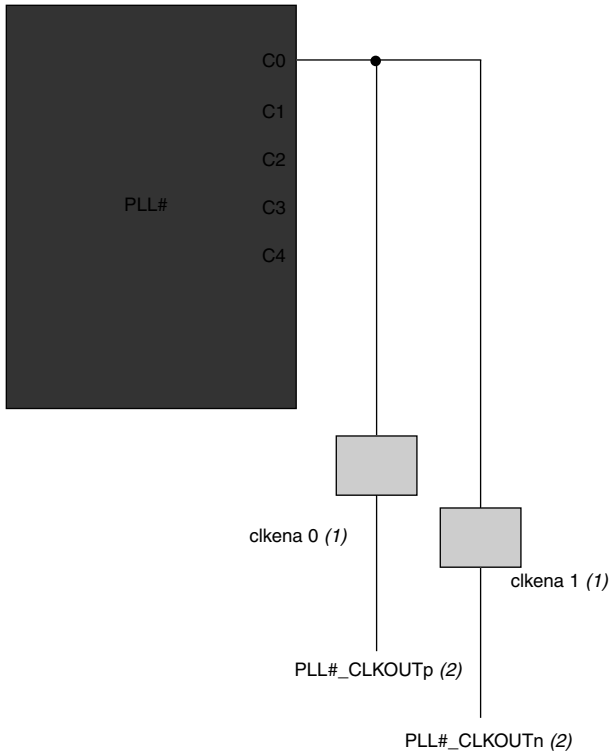
## Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

| Details | |
|---|---|
| Product Status | Active |
| Number of LABs/CLBs | 7155 |
| Number of Logic Elements/Cells | 114480 |
| Total RAM Bits | 3981312 |
| Number of I/O | 528 |
| Number of Gates | - |
| Voltage - Supply | 0.97V ~ 1.03V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 85°C (TJ) |
| Package / Case | 780-BGA |
| Supplier Device Package | 780-FBGA (29x29) |
| Purchase URL | https://www.e-xfl.com/product-detail/intel/ep4ce115f29c9ln |

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

**Figure 5–11. External Clock Outputs for PLLs**




**Notes to Figure 5–11:**

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

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

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

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

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

# Programmable Bandwidth

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

# Phase Shift Implementation

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

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

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

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

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

**Equation 5–1. Fine Resolution Phase Shift**

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

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

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

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

**Equation 5–2. Coarse Resolution Phase Shift**

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

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

# Document Revision History

Table 5–14 lists the revision history for this chapter.

**Table 5–14. Document Revision History**

| Date | Version | Changes |
|---|---|---|
| October 2012 | 2.4 | ■ Updated “Manual Override” and “PLL Cascading” sections.<br>■ Updated Figure 5–9. |
| November 2011 | 2.3 | ■ Updated the “Dynamic Phase Shifting” section.<br>■ Updated Figure 5–26. |
| December 2010 | 2.2 | ■ Updated for the Quartus II software version 10.1 release.<br>■ Updated Figure 5–3 and Figure 5–10.<br>■ Updated “GCLK Network Clock Source Generation”, “PLLs in Cyclone IV Devices”, and “Manual Override” sections.<br>■ Minor text edits. |
| July 2010 | 2.1 | ■ Updated Figure 5–2, Figure 5–3, Figure 5–4, and Figure 5–10.<br>■ Updated Table 5–1, Table 5–2, and Table 5–5.<br>■ Updated “Clock Feedback Modes” section. |
| February 2010 | 2.0 | ■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release.<br>■ Updated “Clock Networks” section.<br>■ Updated Table 5–1 and Table 5–2.<br>■ Added Table 5–3.<br>■ Updated Figure 5–2, Figure 5–3, and Figure 5–9.<br>■ Added Figure 5–4 and Figure 5–10. |
| November 2009 | 1.0 | Initial release. |

# 6. I/O Features in Cyclone IV Devices

CYIV-51006-2.7

This chapter describes the I/O and high speed I/O capabilities and features offered in Cyclone® IV devices.

The I/O capabilities of Cyclone IV devices are driven by the diversification of I/O standards in many low-cost applications, and the significant increase in required I/O performance. Altera's objective is to create a device that accommodates your key board design needs with ease and flexibility.

The I/O flexibility of Cyclone IV devices is increased from the previous generation low-cost FPGAs by allowing all I/O standards to be selected on all I/O banks. Improvements to on-chip termination (OCT) support and the addition of true differential buffers have eliminated the need for external resistors in many applications, such as display system interfaces.

High-speed differential I/O standards have become popular in high-speed interfaces because of their significant advantages over single-ended I/O standards. The Cyclone IV devices support LVDS, BLVDS, RSDS, mini-LVDS, and PPDS. The transceiver reference clocks and the existing general-purpose I/O (GPIO) clock input features also support the LVDS I/O standards.

The Quartus® II software completes the solution with powerful pin planning features that allow you to plan and optimize I/O system designs even before the design files are available.

This chapter includes the following sections:

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

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

### Address and Control/Command Pins

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

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

### Memory Clock Pins

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

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

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

## Cyclone IV Devices Memory Interfaces Features

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

### DDR Input Registers

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

# Section III. System Integration

This section includes the following chapters:

- Chapter 8, Configuration and Remote System Upgrades in Cyclone IV Devices
- Chapter 9, SEU Mitigation in Cyclone IV Devices
- Chapter 10, JTAG Boundary-Scan Testing for Cyclone IV Devices
- Chapter 11, Power Requirements for Cyclone IV Devices

## Revision History

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

**Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 2 of 2)**

| Device | | Data Size (bits) |
|---|---|---|
| Cyclone IV GX | EP4CGX15 | 3,805,568 |
| | EP4CGX22 | 7,600,040 |
| | EP4CGX30 | 7,600,040 |
| | | 22,010,888 (1) |
| | EP4CGX50 | 22,010,888 |
| | EP4CGX75 | 22,010,888 |
| | EP4CGX110 | 39,425,016 |
| | EP4CGX150 | 39,425,016 |

**Note to Table 8–2:**

(1) Only for the F484 package.

Use the data in Table 8–2 to estimate the file size before design compilation. Different configuration file formats, such as Hexadecimal (**.hex**) or Tabular Text File (**.ttf**) formats, have different file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you use compression, the file size varies after each compilation, because the compression ratio depends on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Software Settings* section in volume 2 of the *Configuration Handbook*.

## Configuration and JTAG Pin I/O Requirements

Cyclone IV devices are manufactured using the TSMC 60-nm low-k dielectric process. Although Cyclone IV devices use TSMC 2.5-V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5, 3.0, and 3.3-V configuration voltage standards by following specific requirements.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. When using a serial configuration device in an AS configuration scheme, you must connect a 25-Ω series resistor for the DATA[0] pin. When cascading the Cyclone IV device family in a multi-device configuration for AS, AP, FPP, and PS configuration schemes, you must connect the repeater buffers between the master and slave devices for the DATA and DCLK pins. When using the JTAG configuration scheme in a multi-device configuration, connect 25-Ω resistors on both ends of the TDO-TDI path if the TDO output driver is a non-Cyclone IV device.

The output resistance of the repeater buffers and the TDO path for all cases must fit the maximum overshoot equation shown in Equation 8–1.

**Equation 8–1. (1)**

$$0.8Z_O \leq R_E \leq 1.8Z_O$$

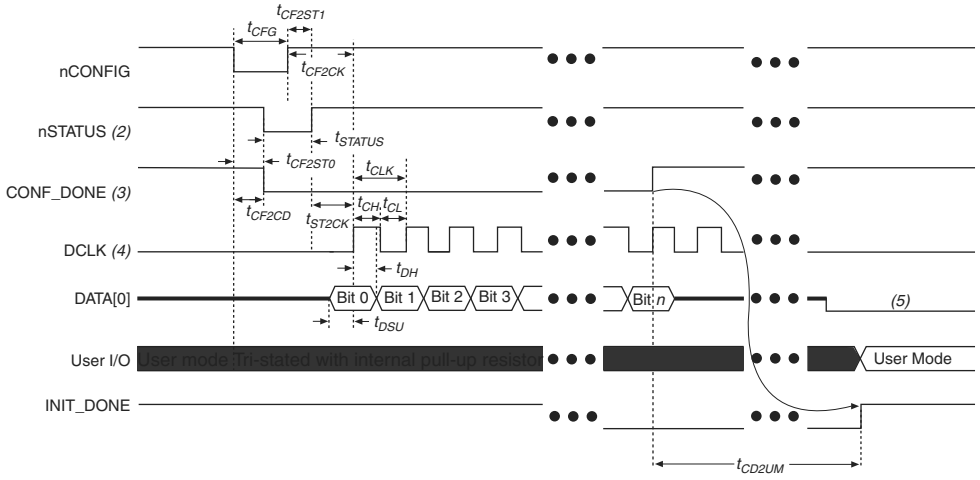
**Note to Equation 8–1:**

(1) $Z_O$ is the transmission line impedance and $R_E$ is the equivalent resistance of the output buffer.

## PS Configuration Timing

A PS configuration must meet the setup and hold timing parameters and the maximum clock frequency. When using a microprocessor or another intelligent host to control the PS interface, ensure that you meet these timing requirements. Figure 8–16 shows the timing waveform for PS configuration when using an external host device.

**Figure 8–16. PS Configuration Timing Waveform** *(1)*




**Notes to Figure 8–16:**

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic-high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) After power up, the Cyclone IV device holds nSTATUS low during POR delay.

(3) After power up, before and during configuration, CONF_DONE is low.

(4) In user mode, drive DCLK either high or low when using the PS configuration scheme, whichever is more convenient. When using the AS configuration scheme, DCLK is a Cyclone IV device output pin and must not be driven externally.

(5) Do not leave the DATA[0] pin floating after configuration. Drive the DATA[0] pin high or low, whichever is more convenient.

Table 8–12 lists the PS configuration timing parameters for Cyclone IV devices.

**Table 8–12. PS Configuration Timing Parameters For Cyclone IV Devices (Part 1 of 2)**

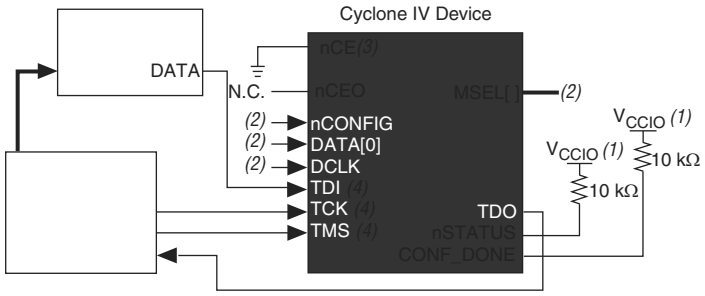
| Symbol | Parameter | Minimum | | Maximum | | Unit |
|---|---|---|---|---|---|---|
| | | Cyclone IV *(1)* | Cyclone IV E *(2)* | Cyclone IV *(1)* | Cyclone IV E *(2)* | |
| $t_{CF2CD}$ | nCONFIG low to CONF_DONE low | — | | 500 | | ns |
| $t_{CF2ST0}$ | nCONFIG low to nSTATUS low | — | | 500 | | ns |
| $t_{CFG}$ | nCONFIG low pulse width | 500 | | — | | ns |
| $t_{STATUS}$ | nSTATUS low pulse width | 45 | | 230 *(3)* | | µs |

JTAG configuration allows an unlimited number of Cyclone IV devices to be cascaded in a JTAG chain.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

Figure 8–27 shows JTAG configuration with a Cyclone IV device and a microprocessor.

**Figure 8–27. JTAG Configuration of a Single Device Using a Microprocessor**




**Notes to Figure 8–27:**

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain.

(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) You must connect the `nCE` pin to GND or driven low for successful JTAG configuration.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. Signals driving into `TDI`, `TMS`, and `TCK` must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

## Configuring Cyclone IV Devices with Jam STAPL

Jam™ STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard. The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam Player, visit the Altera website (www.altera.com).

## Configuring Cyclone IV Devices with the JRunner Software Driver

The JRunner software driver allows you to configure Cyclone IV devices through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The supported programming input file is in **.rbf** format. The JRunner software driver also requires a Chain Description File (**.cdf**) generated by the Quartus II software. The JRunner software driver is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS). You can customize the code to make it run on your embedded platform.

### Reconfiguration

After the configuration data is successfully written into the serial configuration device, the Cyclone IV device does not automatically start reconfiguration. The intelligent host issues the PULSE_NCONFIG JTAG instruction to initialize the reconfiguration process. During reconfiguration, the master device is reset and the SFL design no longer exists in the Cyclone IV device and the serial configuration device configures all the devices in the chain with the user design.

For more information about the SFL, refer to *AN 370: Using the Serial FlashLoader with Quartus II Software.*

## JTAG Instructions

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

### I/O Reconfiguration

Use the CONFIG_IO instruction to reconfigure the I/O configuration shift register (IOCSR) chain. 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. After the configuration is interrupted and JTAG testing is complete, you must reconfigure the part through the PULSE_NCONFIG JTAG instruction or by pulsing the nCONFIG pin low.

You can issue the CONFIG_IO instruction any time during user mode.

You must meet the following timing restrictions when using the CONFIG_IO instruction:

- The CONFIG_IO instruction cannot be issued when the nCONFIG pin is low
- You must observe a 230 µs minimum wait time after any of the following conditions:
  - nCONFIG pin goes high
  - Issuing the PULSE_NCONFIG instruction
  - Issuing the ACTIVE_ENGAGE instruction, before issuing the CONFIG_IO instruction
- You must wait 230 µs after power up, with the nCONFIG pin high before issuing the CONFIG_IO instruction (or wait for the nSTATUS pin to go high)

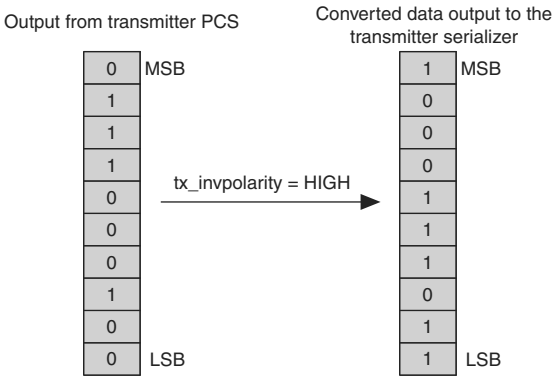
at time $n$ + 2 is encoded as a positive disparity code group. In the same example, the current running disparity at time $n$ + 5 indicates that the K28.5 in time $n$ + 6 should be encoded with a positive disparity. Because tx_forcedisp is high at time $n$ + 6, and tx_dispval is high, the K28.5 at time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

## Miscellaneous Transmitter PCS Features

The transmitter PCS supports the following additional features:

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

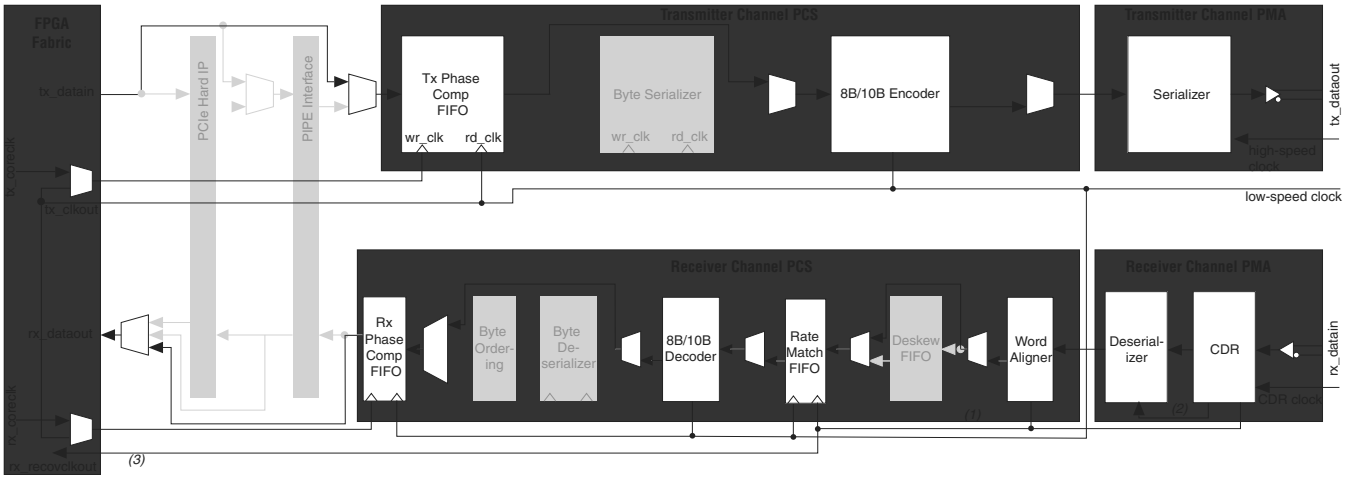
**Figure 1–9. Transmitter Polarity Inversion**




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

Figure 1–55 shows the transceiver channel datapath and clocking when configured in GIGE mode.

**Figure 1–55. Transceiver Channel Datapath and Clocking when Configured in GIGE Mode**




**Notes to Figure 1–55:**

(1) Low-speed recovered clock.

(2) High-speed recovered clock.

(3) Optional `rx_recovclkout` port from CDR low-speed recovered clock is available for applications such as Synchronous Ethernet.

**Table 2–3. Blocks Affected by Reset and Power-Down Signals (Part 2 of 2)**

| Transceiver Block | rx_digitalreset | rx_analogreset | tx_digitalreset | pll_areset | gxb_powerdown |
|---|---|---|---|---|---|
| Serializer | — | — | ✓ | — | ✓ |
| Transmitter Buffer | — | — | — | — | ✓ |
| Transmitter XAUI State Machine | — | — | ✓ | — | ✓ |
| Receiver Buffer | — | — | — | — | ✓ |
| Receiver CDR | — | ✓ | — | — | ✓ |
| Receiver Deserializer | — | — | — | — | ✓ |
| Receiver Word Aligner | ✓ | — | — | — | ✓ |
| Receiver Deskew FIFO | ✓ | — | — | — | ✓ |
| Receiver Clock Rate Compensation FIFO | ✓ | — | — | — | ✓ |
| Receiver 8B/10B Decoder | ✓ | — | — | — | ✓ |
| Receiver Byte Deserializer | ✓ | — | — | — | ✓ |
| Receiver Byte Ordering | ✓ | — | — | — | ✓ |
| Receiver Phase Compensation FIFO | ✓ | — | — | — | ✓ |
| Receiver XAUI State Machine | ✓ | — | — | — | ✓ |
| BIST Verifiers | ✓ | — | — | — | ✓ |

# Transceiver Reset Sequences

You can configure transceiver channels in Cyclone IV GX devices in various configurations. In all functional modes except XAUI functional mode, transceiver channels can be either bonded or non-bonded. In XAUI functional mode, transceiver channels must be bonded. In PCI Express® (PCIe®) functional mode, transceiver channels can be either bonded or non-bonded and need to follow a specific reset sequence.

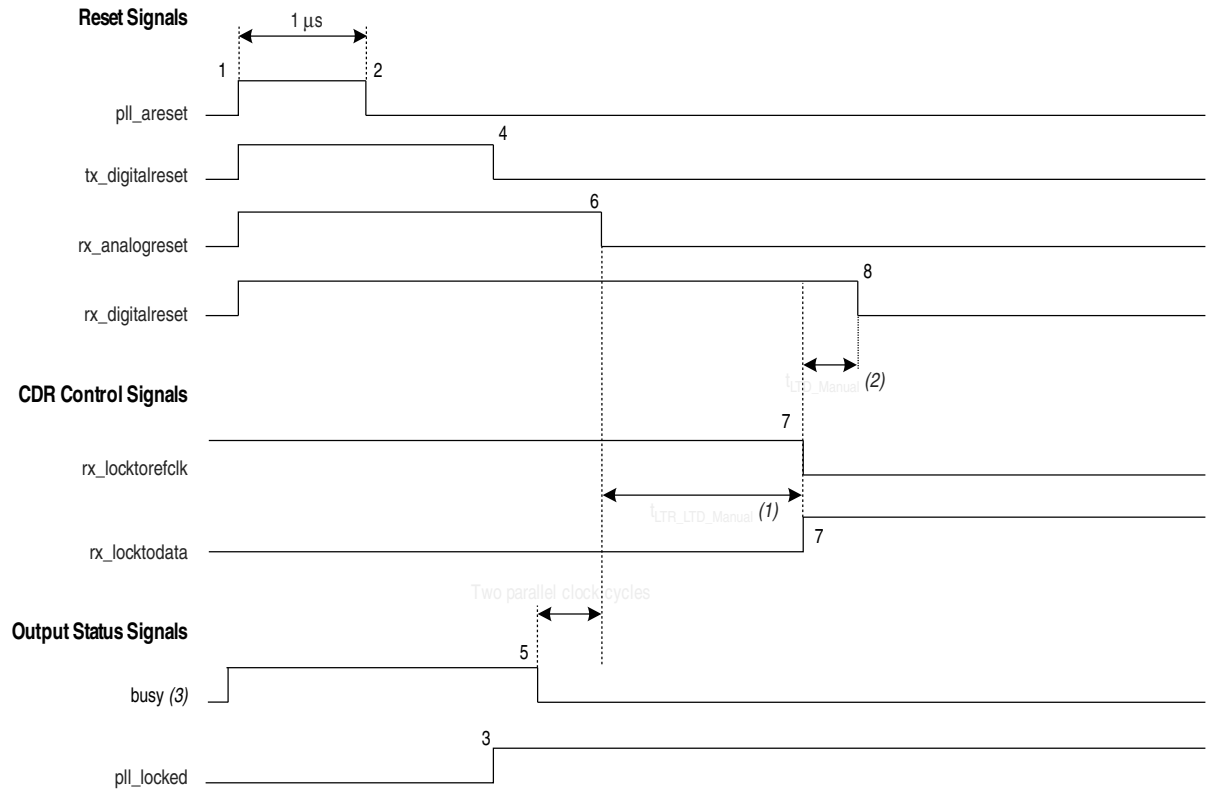
The two categories of reset sequences for Cyclone IV GX devices described in this chapter are:

- "All Supported Functional Modes Except the PCIe Functional Mode" on page 2–6—describes the reset sequences in bonded and non-bonded configurations.
- "PCIe Functional Mode" on page 2–17—describes the reset sequence for the initialization/compliance phase and the normal operation phase in PCIe functional modes.

### Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode

This configuration contains both a transmitter and receiver channel. If you create a **Receiver and Transmitter** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in manual lock mode, use the reset sequence shown in Figure 2–9.

**Figure 2–9. Sample Reset Sequence of Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode**




**Notes to Figure 2–9:**

(1) For $t_{LTR\_LTD\_Manual}$ duration, refer to the *Cyclone IV Device Datasheet* chapter.

(2) For $t_{LTD\_Manual}$ duration, refer to the *Cyclone IV Device Datasheet* chapter.

(3) 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 2–9, perform the following reset procedure for the receiver in manual 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`, `rx_digitalreset`, and `rx_locktorefclk` signals asserted and the `rx_locktodata` signal deasserted 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` (marker 4). For receiver operation, after deassertion of busy signal (marker 5), wait for two parallel clock cycles to deassert the `rx_analogreset` signal (marker 6). After `rx_analogreset` deassert, `rx_pll_locked` will assert.

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

| Dynamic Reconfiguration Supported Mode | Operational Mode | | | Quartus II Instances | | | .mif Requirements |
|---|---|---|---|---|---|---|---|
| | Transmitter Only | Receiver Only | Transmitter and Receiver Only | ALTGX | ALTGX_ RECONFIG | ALTPLL_ RECONFIG | |
| Channel Reconfiguration | | | | | | | |
| Channel Interface | ✓ | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| Data Rate Division in Receiver Channel | — | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| PLL Reconfiguration | ✓ | ✓ | ✓ | ✓ | — | ✓ | ✓ |

The following modes are available for dynamically reconfiguring the Cyclone IV transceivers:

- "PMA Controls Reconfiguration Mode" on page 3–13
- "Transceiver Channel Reconfiguration Mode" on page 3–21
  - Channel interface (**.mif** based)
  - Data rate division in receiver channel (**.mif** based)

The following sections describe each of these modes in detail.

The following modes are unsupported for dynamic reconfiguration:

- Dynamically enable/disable PRBS or BIST
- Switch between a receiver-only channel and a transmitter-only channel
- Switch between a ×1 mode to a bonded ×4 mode

## PMA Controls Reconfiguration Mode

You can dynamically reconfigure the following PMA controls for all supported transceiver configurations channels as configured in the ALTGX instances:

- Pre-emphasis settings
- Equalization settings (channel reconfiguration mode does not support equalization settings)
- DC gain settings
- $V_{OD}$ settings

You can use the analog reconfiguration feature to dynamically reconfigure the transceivers channels setting in either the transmitter or the receivers in the PMA blocks. You can update the PMA controls on-the-fly based on the desired input. You can perform both read and write transaction separately for this analog reconfiguration mode.

### Read Transaction

If you want to read the existing values from a specific channel connected to the ALTGX_RECONFIG instance, observe the corresponding byte positions of the PMA control output port after the read transaction is completed.

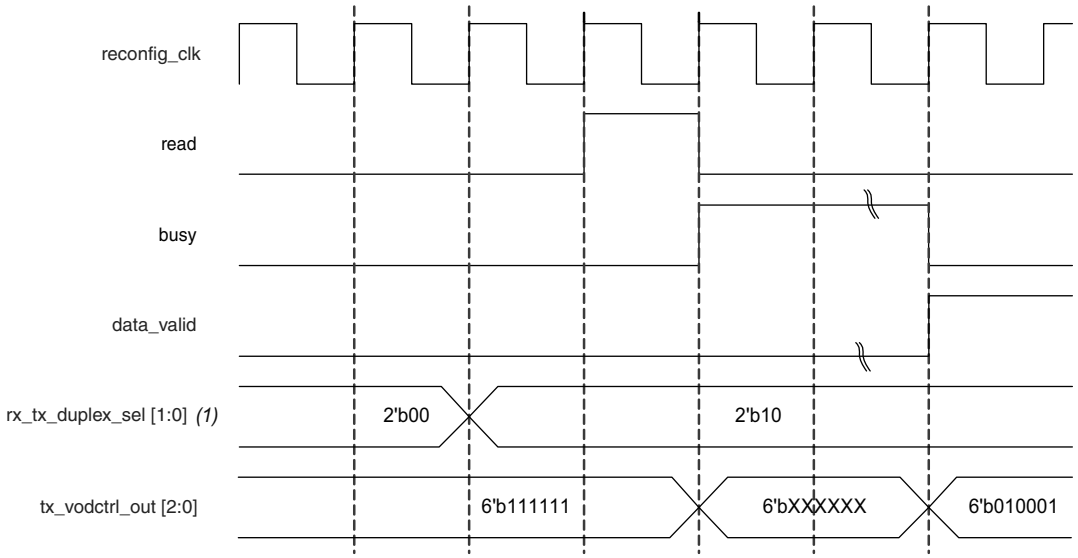
For example, if the number of channels controlled by the ALTGX_RECONFIG is two, the `tx_vodctrl_out` is 6 bits wide. The `tx_vodctrl_out[2:0]` signal corresponds to channel 1 and the `tx_vodctrl_out[5:3]` signal corresponds to channel 2.

To complete a read transaction to the $V_{OD}$ values of the second channel, perform the following steps:

1. Before you initiate a read transaction, set the `rx_tx_duplex_sel` port to **2'b10** so that only the transmit PMA controls are read from the transceiver channel.
2. Ensure that the `busy` signal is low before you start a read transaction.
3. Assert the read signal for one `reconfig_clk` clock cycle. This initiates the read transaction.
4. The `busy` output status signal is asserted high to indicate that the dynamic reconfiguration controller is busy reading the PMA control settings.
5. When the read transaction has completed, the `busy` signal goes low. The `data_valid` signal is asserted, indicating that the data available at the read control signal is valid.
6. To read the current $V_{OD}$ values in channel 2, observe the values in `tx_vodctrl_out[5:3]`.

In the waveform example shown in Figure 3–7, the transmit $V_{OD}$ settings written in channels 1 and 2 prior to the read transaction are 3'b001 and 3'b010, respectively.

**Figure 3–7. Read Transaction Waveform—Use the same control signal for all the channels Option Enabled**




**Note to Figure 3–7:**

(1) In this waveform example, you want to read from only the transmitter portion of all the channels.

☞ Simultaneous write and read transactions are not allowed.

## Control and Status Signals for Channel Reconfiguration

The various control and status signals involved in the Channel Reconfiguration mode are as follows. Refer to "Dynamic Reconfiguration Controller Port List" on page 3–4 for the descriptions of the control and status signals.

The following are the input control signals:

- `logical_channel_address[n..0]`
- `reset_reconfig_address`
- `reconfig_reset`
- `reconfig_mode_sel[2..0]`
- `write_all`

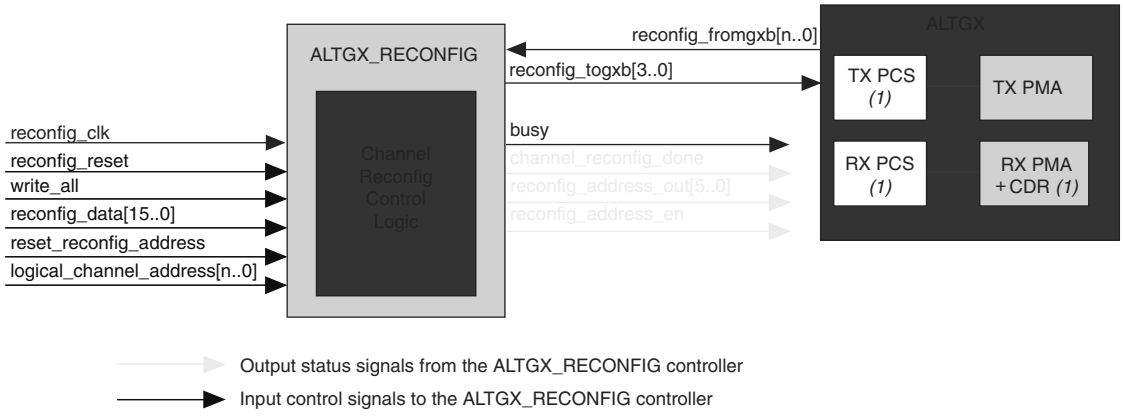
The following are output status signals:

- `reconfig_address_en`
- `reconfig_address_out[5..0]`
- `channel_reconfig_done`
- `busy`

The ALTGX_RECONFIG connection to the ALTGX instances when set in channel reconfiguration mode are as follows. For the port information, refer to "Dynamic Reconfiguration Controller Port List" on page 3–4.

Figure 3–10 shows the connection for channel reconfiguration mode.

**Figure 3–10. ALTGX and ALTGX_RECONFIG Connection for Channel Reconfiguration Mode**




**Note to Figure 3–10:**

(1) This block can be reconfigured in channel reconfiguration mode.

# Power Consumption

Use the following methods to estimate power for a design:

- the Excel-based EPE
- the Quartus® II PowerPlay power analyzer feature

The interactive Excel-based EPE is used prior to designing the device to get a magnitude estimate of the device power. The Quartus II PowerPlay power analyzer provides better quality estimates based on the specifics of the design after place-and-route is complete. The PowerPlay power analyzer can apply a combination of user-entered, simulation-derived, and estimated signal activities that, combined with detailed circuit models, can yield very accurate power estimates.

For more information about power estimation tools, refer to the *Early Power Estimator User Guide* and the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

# Switching Characteristics

This section provides performance characteristics of Cyclone IV core and periphery blocks for commercial grade devices.

These characteristics can be designated as Preliminary or Final.

- Preliminary characteristics are created using simulation results, process data, and other known parameters. The upper-right hand corner of these tables show the designation as “Preliminary”.
- Final numbers are based on actual silicon characterization and testing. The numbers reflect the actual performance of the device under worst-case silicon process, voltage, and junction temperature conditions. There are no designations on finalized tables.

Figure 1–4 shows the differential receiver input waveform.

**Figure 1–4. Receiver Input Waveform**

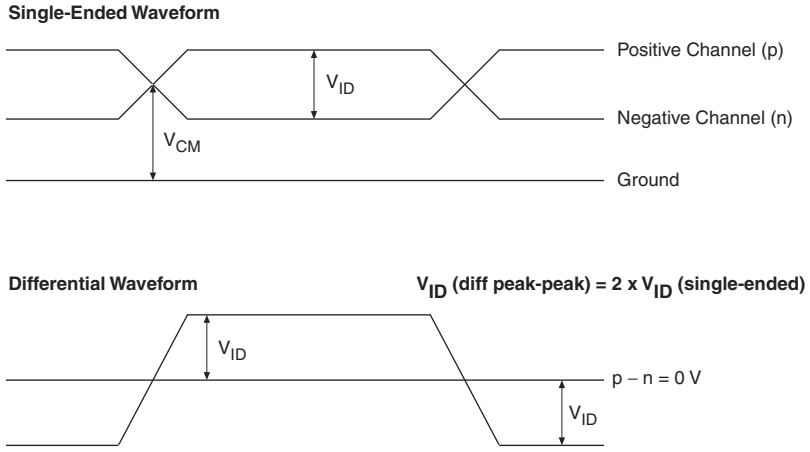



Figure 1–5 shows the transmitter output waveform.

**Figure 1–5. Transmitter Output Waveform**

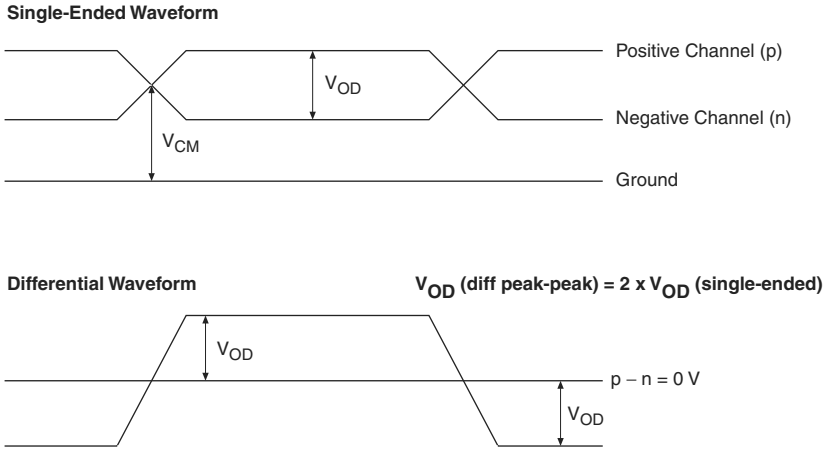



Table 1–22 lists the typical $V_{OD}$ for Tx term that equals 100 Ω.

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

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

**Note to Table 1–22:**

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

**Table 1–31. RSDS Transmitter Timing Specifications for Cyclone IV Devices (1), (2), (4) (Part 2 of 2)**

| Symbol | Modes | C6 | | | C7, I7 | | | C8, A7 | | | C8L, I8L | | | C9L | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | |
| $t_{LOCK}$ (3) | — | — | — | 1 | — | — | 1 | — | — | 1 | — | — | 1 | — | — | 1 | ms |

**Notes to Table 1–31:**

(1) Applicable for true RSDS and emulated RSDS_E_3R transmitter.

(2) Cyclone IV E devices—true RSDS transmitter is only supported at the output pin of Row I/O Banks 1, 2, 5, and 6. Emulated RSDS transmitter is supported at the output pin of all I/O Banks.
Cyclone IV GX devices—true RSDS transmitter is only supported at the output pin of Row I/O Banks 5 and 6. Emulated RSDS transmitter is supported at the output pin of I/O Banks 3, 4, 5, 6, 7, 8, and 9.

(3) $t_{LOCK}$ is the time required for the PLL to lock from the end-of-device configuration.

(4) Cyclone IV E 1.0 V core voltage devices only support C8L, C9L, and I8L speed grades. Cyclone IV E 1.2 V core voltage devices only support C6, C7, C8, I7, and A7 speed grades. Cyclone IV GX devices only support C6, C7, C8, and I7 speed grades.

**Table 1–32. Emulated RSDS_E_1R Transmitter Timing Specifications for Cyclone IV Devices (1), (3) (Part 1 of 2)**

| Symbol | Modes | C6 | | | C7, I7 | | | C8, A7 | | | C8L, I8L | | | C9L | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | |
| $f_{HSCLK}$ (input clock frequency) | ×10 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 72.5 | MHz |
| | ×8 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 72.5 | MHz |
| | ×7 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 72.5 | MHz |
| | ×4 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 72.5 | MHz |
| | ×2 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 85 | 5 | — | 72.5 | MHz |
| | ×1 | 5 | — | 170 | 5 | — | 170 | 5 | — | 170 | 5 | — | 170 | 5 | — | 145 | MHz |
| Device operation in Mbps | ×10 | 100 | — | 170 | 100 | — | 170 | 100 | — | 170 | 100 | — | 170 | 100 | — | 145 | Mbps |
| | ×8 | 80 | — | 170 | 80 | — | 170 | 80 | — | 170 | 80 | — | 170 | 80 | — | 145 | Mbps |
| | ×7 | 70 | — | 170 | 70 | — | 170 | 70 | — | 170 | 70 | — | 170 | 70 | — | 145 | Mbps |
| | ×4 | 40 | — | 170 | 40 | — | 170 | 40 | — | 170 | 40 | — | 170 | 40 | — | 145 | Mbps |
| | ×2 | 20 | — | 170 | 20 | — | 170 | 20 | — | 170 | 20 | — | 170 | 20 | — | 145 | Mbps |
| | ×1 | 10 | — | 170 | 10 | — | 170 | 10 | — | 170 | 10 | — | 170 | 10 | — | 145 | Mbps |
| $t_{DUTY}$ | — | 45 | — | 55 | 45 | — | 55 | 45 | — | 55 | 45 | — | 55 | 45 | — | 55 | % |
| TCCS | — | — | — | 200 | — | — | 200 | — | — | 200 | — | — | 200 | — | — | 200 | ps |
| Output jitter (peak to peak) | — | — | — | 500 | — | — | 500 | — | — | 550 | — | — | 600 | — | — | 700 | ps |
| $t_{RISE}$ | 20 – 80%, $C_{LOAD}$ = 5 pF | — | 500 | — | — | 500 | — | — | 500 | — | — | 500 | — | — | 500 | — | ps |
| $t_{FALL}$ | 20 – 80%, $C_{LOAD}$ = 5 pF | — | 500 | — | — | 500 | — | — | 500 | — | — | 500 | — | — | 500 | — | ps |