Intel - EP4CE30F23C8N Datasheet

Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

Applications of Embedded - FPGAs

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

Details	
Product Status	Active
Number of LABs/CLBs	1803
Number of Logic Elements/Cells	28848
Total RAM Bits	608256
Number of I/O	328
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce30f23c8n

PLLs in Cyclone IV Devices

Cyclone IV GX devices offer two variations of PLLs: general purpose PLLs and multipurpose PLLs. Cyclone IV E devices only have the general purpose PLLs.

The general purpose PLLs are used for general-purpose applications in the FPGA fabric and periphery such as external memory interfaces. The multipurpose PLLs are used for clocking the transceiver blocks. When the multipurpose PLLs are not used for transceiver clocking, they can be used for general-purpose clocking.

For more details about the multipurpose PLLs used for transceiver clocking, refer to the *Cyclone IV Transceivers* chapter.

Cyclone IV GX devices contain up to eight general purpose PLLs and multipurpose PLLs while Cyclone IV E devices have up to four general purpose PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces.

For more information about the number of general purpose PLLs and multipurpose PLLs in each device density, refer to the *Cyclone IV Device Family Overview* chapter.

The general I/O pins cannot drive the PLL clock input pins.

Table 5–5 lists the features available in Cyclone IV GX PLLs.

Table 5–5. Cyclone IV GX PLL Features (Part 1 of 2)

Features	Availability									
	General Purpose PLLs				Multipurpose PLLs					
	PLL_1 *(1), (10)*	PLL_2 *(1), (10)*	PLL_3 *(2)*	PLL_4 *(3)*	PLL_1 *(4)*	PLL_2 *(4)*	PLL_5 *(1), (10)*	PLL_6 *(1), (10)*	PLL_7 *(1)*	PLL_8 *(1)*
C (output counters)	5									
M, N, C counter sizes	1 to 512 *(5)*									
Dedicated clock outputs	1 single-ended or 1 differential pair									
Clock input pins	12 single-ended or 6 differential pairs *(6)* and 4 differential pairs *(7)*									
Spread-spectrum input clock tracking	✓ *(8)*									
PLL cascading	Through GCLK									
Source-Synchronous Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
No Compensation Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Normal Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Zero Delay Buffer Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Deterministic Latency Compensation Mode	✓	✓	—	—	✓	✓	✓	✓	✓	✓
Phase shift resolution *(9)*	Down to 96 ps increments									
Programmable duty cycle	✓									
Output counter cascading	✓									

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

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

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

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

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

Equation 5–2. Coarse Resolution Phase Shift

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

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.

Table 5–10. Loop Filter Control of High Frequency Capacitor

LFC[1]	LFC[0]	Setting (Decimal)
0	0	0
0	1	1
1	1	3

Bypassing a PLL Counter

Bypassing a PLL counter results in a divide (N, C0 to C4 counters) factor of one.

Table 5–11 lists the settings for bypassing the counters in PLLs of Cyclone IV devices.

Table 5–11. PLL Counter Settings

PLL Scan Chain Bits [0..8] Settings									Description
LSB								MSB	
X	X	X	X	X	X	X	X	1 *(1)*	PLL counter bypassed
X	X	X	X	X	X	X	X	0 *(1)*	PLL counter not bypassed

Note to Table 5–11:

(1) Bypass bit.

To bypass any of the PLL counters, set the bypass bit to 1. The values on the other bits are then ignored.

Dynamic Phase Shifting

The dynamic phase shifting feature allows the output phase of individual PLL outputs to be dynamically adjusted relative to each other and the reference clock without sending serial data through the scan chain of the corresponding PLL. This feature simplifies the interface and allows you to quickly adjust t_{CO} delays by changing output clock phase shift in real time. This is achieved by incrementing or decrementing the VCO phase-tap selection to a given C counter or to the M counter. The phase is shifted by 1/8 the VCO frequency at a time. The output clocks are active during this phase reconfiguration process.

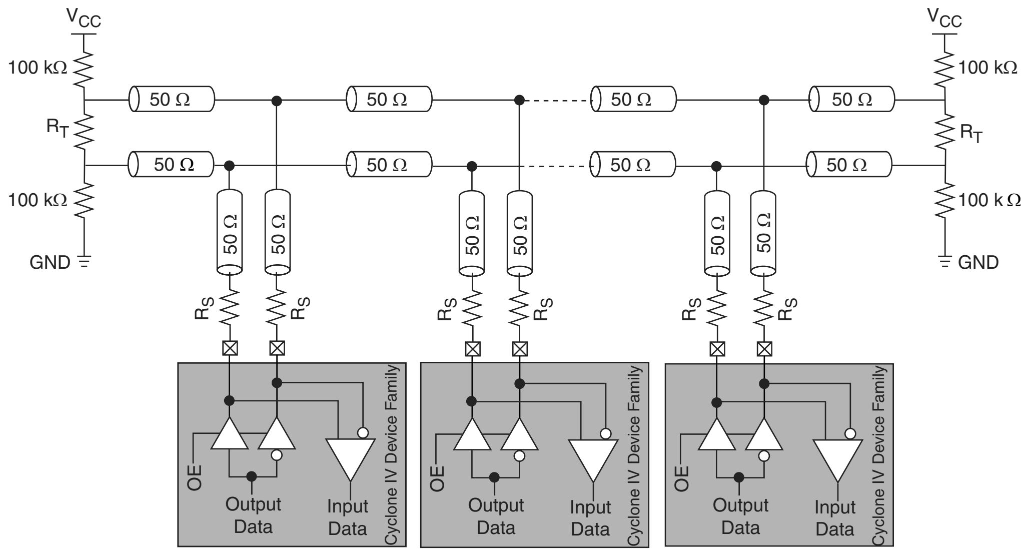
Table 5–12 lists the control signals that are used for dynamic phase shifting.

Table 5–12. Dynamic Phase Shifting Control Signals (Part 1 of 2)

Signal Name	Description	Source	Destination
`phasecounterselect[2..0]`	Counter Select. Three bits decoded to select either the M or one of the C counters for phase adjustment. One address map to select all C counters. This signal is registered in the PLL on the rising edge of `scanclk`.	Logic array or I/O pins	PLL reconfiguration circuit
`phaseupdown`	Selects dynamic phase shift direction; 1= UP, 0 = DOWN. Signal is registered in the PLL on the rising edge of `scanclk`.	Logic array or I/O pins	PLL reconfiguration circuit
`phasestep`	Logic high enables dynamic phase shifting.	Logic array or I/O pins	PLL reconfiguration circuit

Figure 6–14 shows a typical BLVDS topology with multiple transmitter and receiver pairs.

Figure 6–14. BLVDS Topology with Cyclone IV Devices Transmitters and Receivers



The BLVDS I/O standard is supported on the top, bottom, and right I/O banks of Cyclone IV devices. The BLVDS transmitter uses two single-ended output buffers with the second output buffer programmed as inverted, while the BLVDS receiver uses a true LVDS input buffer. The transmitter and receiver share the same pins. An output-enabled (`OE`) signal is required to tristate the output buffers when the LVDS input buffer receives a signal.

For more information, refer to the *Cyclone IV Device Datasheet* chapter.

Designing with BLVDS

The BLVDS bidirectional communication requires termination at both ends of the bus in BLVDS. The termination resistor (R_T) must match the bus differential impedance, which in turn depends on the loading on the bus. Increasing the load decreases the bus differential impedance. With termination at both ends of the bus, termination is not required between the two signals at the input buffer. A single series resistor (R_S) is required at the output buffer to match the output buffer impedance to the transmission line impedance. However, this series resistor affects the voltage swing at the input buffer. The maximum data rate achievable depends on many factors.

Altera recommends that you perform simulation using the IBIS model while considering factors such as bus loading, termination values, and output and input buffer location on the bus to ensure that the required performance is achieved.

For more information about BLVDS interface support in Altera devices, refer to *AN 522: Implementing Bus LVDS Interface in Supported Altera Device Families*.

Table 7–1 lists the number of `DQS` or `DQ` groups supported on each side of the Cyclone IV GX device.

Table 7–1. Cyclone IV GX Device DQS and DQ Bus Mode Support for Each Side of the Device

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CGX15	169-pin FBGA	Right	1	0	0	0	—	—
		Top *(1)*	1	0	0	0	—	—
		Bottom *(2)*	1	0	0	0	—	—
EP4CGX22 EP4CGX30	169-pin FBGA	Right	1	0	0	0	—	—
		Top *(1)*	1	0	0	0	—	—
		Bottom *(2)*	1	0	0	0	—	—
	324-pin FBGA	Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	484-pin FBGA *(3)*	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
EP4CGX50 EP4CGX75	484-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	672-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
EP4CGX110 EP4CGX150	484-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	672-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	896-pin FBGA	Right	6	3	2	2	1	1
		Top	6	3	3	3	1	1
		Bottom	6	3	3	3	1	1

Notes to Table 7–1:

(1) Some of the DQ pins can be used as RUP and RDN pins. You cannot use these groups if you are using these pins as RUP and RDN pins for OCT calibration.

(2) Some of the DQ pins can be used as RUP pins while the DM pins can be used as RDN pins. You cannot use these groups if you are using the RUP and RDN pins for OCT calibration.

(3) Only available for EP4CGX30 device.

8. Configuration and Remote System Upgrades in Cyclone IV Devices

CYIV-51008-1.7

This chapter describes the configuration and remote system upgrades in Cyclone® IV devices. Cyclone IV (Cyclone IV GX and Cyclone IV E) devices use SRAM cells to store configuration data. You must download the configuration data to Cyclone IV devices each time the device powers up because SRAM memory is volatile.

Cyclone IV devices are configured using one of the following configuration schemes:

- Active serial (AS)
- Active parallel (AP) (supported in Cyclone IV E devices only)
- Passive serial (PS)
- Fast passive parallel (FPP) (not supported in EP4CGX15, EP4CGX22, and EP4CGX30 [except for the F484 package] devices)
- JTAG

Cyclone IV devices offer the following configuration features:

- Configuration data decompression (“Configuration Data Decompression” on page 8–2)
- Remote system upgrade (“Remote System Upgrade” on page 8–69)

System designers face difficult challenges, such as shortened design cycles, evolving standards, and system deployments in remote locations. Cyclone IV devices help overcome these challenges with inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduced time-to-market, and extended product life.

Configuration

This section describes Cyclone IV device configuration and includes the following topics:

- “Configuration Features” on page 8–2
- “Configuration Requirement” on page 8–3
- “Configuration Process” on page 8–6
- “Configuration Scheme” on page 8–8
- “AS Configuration (Serial Configuration Devices)” on page 8–10
- “AP Configuration (Supported Flash Memories)” on page 8–21
- “PS Configuration” on page 8–32

Configuration Scheme

A configuration scheme with different configuration voltage standards is selected by driving the MSEL pins either high or low, as shown in Table 8–3, Table 8–4, and Table 8–5.

☞ Hardwire the MSEL pins to V_{CCA} or GND without pull-up or pull-down resistors to avoid problems detecting an incorrect configuration scheme. Do not drive the MSEL pins with a microprocessor or another device.

Table 8–3. Configuration Schemes for Cyclone IV GX Devices (EP4CGX15, EP4CGX22, and EP4CGX30 [except for F484 Package])

Configuration Scheme	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) *(1)*
AS	1	0	1	Fast	3.3
	0	1	1	Fast	3.0, 2.5
	0	0	1	Standard	3.3
	0	1	0	Standard	3.0, 2.5
PS	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	0	Fast	1.8, 1.5
	0	0	0	Standard	3.3, 3.0, 2.5
JTAG-based configuration *(2)*	*(3)*	*(3)*	*(3)*	—	—

Notes to Table 8–3:

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

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

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

Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 1 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) *(1)*
AS	1	1	0	1	Fast	3.3
	1	0	1	1	Fast	3.0, 2.5
	1	0	0	1	Standard	3.3
	1	0	1	0	Standard	3.0, 2.5
PS	1	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	1	0	Fast	1.8, 1.5
	1	0	0	0	Standard	3.3, 3.0, 2.5
	0	0	0	0	Standard	1.8, 1.5
FPP	0	0	1	1	Fast	3.3, 3.0, 2.5
	0	1	0	0	Fast	1.8, 1.5
	0	0	0	1	Standard	3.3, 3.0, 2.5
	0	0	1	0	Standard	1.8, 1.5

Programming Serial Configuration Devices In-System with the JTAG Interface

Cyclone IV devices in a single- or multiple-device chain support in-system programming of a serial configuration device with the JTAG interface through the SFL design. The intelligent host or download cable of the board can use the four JTAG pins on the Cyclone IV device to program the serial configuration device in system, even if the host or download cable cannot access the configuration pins (`DCLK`, `DATA`, `ASDI`, and `nCS` pins).

The SFL design is a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses their JTAG interface to access the EPCS JTAG Indirect Configuration Device Programming (**.jic**) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

In a multiple device chain, you must only configure the master device that controls the serial configuration device. Slave devices in the multiple device chain that are configured by the serial configuration device do not have to be configured when using this feature. To successfully use this feature, set the `MSEL` pins of the master device to select the AS configuration scheme (Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9). The serial configuration device in-system programming through the Cyclone IV device JTAG interface has three stages, which are described in the following sections:

- “Loading the SFL Design”
- “ISP of the Configuration Device” on page 8–56
- “Reconfiguration” on page 8–57

Loading the SFL Design

The SFL design is a design inside the Cyclone IV device that bridges the JTAG interface and AS interface with glue logic.

The intelligent host uses the JTAG interface to configure the master device with a SFL design. The SFL design allows the master device to control the access of four serial configuration device pins, also known as the Active Serial Memory Interface (ASMI) pins, through the JTAG interface. The ASMI pins are serial clock input (`DCLK`), serial data output (`DATA`), AS data input (`ASDI`), and active-low chip select (`nCS`) pins.

Use the ACTIVE_DISENGAGE instruction with the CONFIG_IO instruction to interrupt configuration. Table 8–16 lists the sequence of instructions to use for various CONFIG_IO usage scenarios.

Table 8–16. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows *(1)*

JTAG Instruction	Configuration Scheme and Current State of the Cyclone IV Device											
	Prior to User Mode (Interrupting Configuration)				User Mode				Power Up			
	PS	FPP	AS	AP	PS	FPP	AS	AP	PS	FPP	AS	AP
ACTIVE_DISENGAGE	O	O	O	O	O	O	O	O	—	—	—	—
CONFIG_IO	R	R	R	R	R	R	R	R	NA	NA	NA	NA
JTAG Boundary Scan Instructions (no JTAG_PROGRAM)	O	O	O	O	O	O	O	O	—	—	—	—
ACTIVE_ENGAGE			R *(2)*	R *(2)*			R *(2)*	R *(2)*	—	—	—	—
PULSE_NCONFIG	A	A	A *(3)*	A *(3)*	A	A	O	O	—	—	—	—
Pulse nCONFIG pin			A *(3)*	A *(3)*			O	O	—	—	—	—
JTAG TAP Reset	R	R	R	R	R	R	R	R	—	—	—	—

Notes to Table 8–16:

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

(2) Required if you use ACTIVE_DISENGAGE.

(3) Neither of the instruction is required if you use ACTIVE_ENGAGE.

The CONFIG_IO instruction does not hold nSTATUS low until reconfiguration. You must disengage the AS or AP configuration controller by issuing the ACTIVE_DISENGAGE and ACTIVE_ENGAGE instructions when active configuration is interrupted. You must issue the ACTIVE_DISENGAGE instruction alone or prior to the CONFIG_IO instruction if the JTAG_PROGRAM instruction is to be issued later (Table 8–17). This puts the active configuration controllers into the idle state. The active configuration controller is re-engaged after user mode is reached through JTAG programming (Table 8–17).

☞ While executing the CONFIG_IO instruction, all user I/Os are tri-stated.

If reconfiguration after interruption is performed using configuration modes (rather than using JTAG_PROGRAM), it is not necessary to issue the ACTIVE_DISENGAGE instruction prior to CONFIG_IO. You can start reconfiguration by either pulling nCONFIG low for at least 500 ns or issuing the PULSE_NCONFIG instruction. If the ACTIVE_DISENGAGE instruction was issued and the JTAG_PROGRAM instruction fails to enter user mode, you must issue the ACTIVE_ENGAGE instruction to reactivate the active configuration controller. Issuing the ACTIVE_ENGAGE instruction also triggers reconfiguration in configuration modes; therefore, it is not necessary to pull nCONFIG low or issue the PULSE_NCONFIG instruction.

EN_ACTIVE_CLK

The EN_ACTIVE_CLK instruction causes the CLKUSR pin signal to replace the internal oscillator as the clock source. When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. After this instruction is issued, other JTAG instructions can be issued while the CLKUSR pin signal remains as the clock source. The clock source is only reverted back to the internal oscillator by issuing the DIS_ACTIVE_CLK instruction or a POR.

DIS_ACTIVE_CLK

The DIS_ACTIVE_CLK instruction breaks the CLKUSR enable latch set by the EN_ACTIVE_CLK instruction and causes the clock source to revert back to the internal oscillator. After the DIS_ACTIVE_CLK instruction is issued, you must continue to clock the CLKUSR pin for 10 clock cycles.

Changing the Start Boot Address of the AP Flash

In the AP configuration scheme (for Cyclone IV E devices only), you can change the default configuration boot address of the parallel flash memory to any desired address using the APFC_BOOT_ADDR JTAG instruction.

APFC_BOOT_ADDR

The APFC_BOOT_ADDR instruction is for Cyclone IV E devices only and allows you to define a start boot address for the parallel flash memory in the AP configuration scheme.

This instruction shifts in a start boot address for the AP flash. When this instruction becomes the active instruction, the TDI and TDO pins are connected through a 22-bit active boot address shift register. The shifted-in boot address bits get loaded into the 22-bit AP boot address update register, which feeds into the AP controller. The content of the AP boot address update register can be captured and shifted-out of the active boot address shift register from TDO.

The boot address in the boot address shift register and update register are shifted to the right (in the LSB direction) by two bits versus the intended boot address. The reason for this is that the two LSB of the address are not accessible. When this boot address is fed into the AP controller, two 0s are attached in the end as LSB, thereby pushing the shifted-in boot address to the left by two bits, which become the actual AP boot address the AP controller gets.

If you have enabled the remote update feature, the APFC_BOOT_ADDR instruction sets the boot address for the factory configuration only.

☞ The APFC_BOOT_ADDR instruction is retained after reconfiguration while the system board is still powered on. However, you must reprogram the instruction whenever you restart the system board.

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that stores the configuration addresses, watchdog timer settings, and status information. Table 8–22 lists these registers.

Table 8–22. Remote System Upgrade Registers

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic. Write access is enabled in remote update mode for factory configurations to allow writing to the update register. Write access is disabled for all application configurations in remote update mode.
Control register	This register contains the current configuration address, the user watchdog timer settings, one option bit for checking early `CONF_DONE`, and one option bit for selecting the internal oscillator as the startup state machine clock. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is started, the contents of the update register are written into the control register.
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a read in a factory configuration, this register is read into the shift register.
Status register	This register is written by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

The control and status registers of the remote system upgrade are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer) or the `CLKUSR`. However, the shift and update registers of the remote system upgrade are clocked by the maximum frequency of 40-MHz user clock input (`RU_CLK`). There is no minimum frequency for `RU_CLK`.

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration address, the user watchdog timer settings, and option bits for a application configuration. In remote update mode for the AS configuration scheme, the control register address bits are set to all zeros (`24'b0`) at power up to load the AS factory configuration. In remote update mode for the AP configuration scheme, the control register address bits are set to `24'h010000` (`24'b1 0000 0000 0000 0000`) at power up to load the AP default factory configuration. However, for the AP configuration scheme, you can change the default factory configuration address to any desired address using the `APFC_BOOT_ADDR` JTAG instruction. Additionally, a factory configuration in remote update mode has write access to this register.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition, but before the factory configuration is loaded.

Table 8–26. Control Register Contents After an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting In Remote Update
nCONFIG reset	All bits are 0
nSTATUS error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
Wd time out	All bits are 0

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from indefinitely stalling the device. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Cyclone IV device.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The remote system upgrade circuitry appends 17'b1000 to form the 29-bits value for the watchdog timer. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator or CLKUSR (maximum frequency of 40 MHz).

Table 8–27 lists the operating range of the 10-MHz internal oscillator.

Table 8–27. 10-MHz Internal Oscillator Specifications

Minimum	Typical	Maximum	Unit
5	6.5	10	MHz

The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting RU_nRSTIMER. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (Wd) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

☞ To allow the remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the RU_nRSTIMER signal active for a minimum of 250 ns. This is equivalent to strobing the reset_timer input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

Errors during configuration are detected by the CRC engine. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

In some applications, it is necessary for a device to wake up very quickly to begin operation. Cyclone IV devices offer the Fast-On feature to support fast wake-up time applications. The MSEL pin settings determine the POR time (t_{POR}) of the device.

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

For more information about the POR specifications, refer to the *Cyclone IV Device Datasheet* chapter.

Document Revision History

Table 11–3 lists the revision history for this chapter.

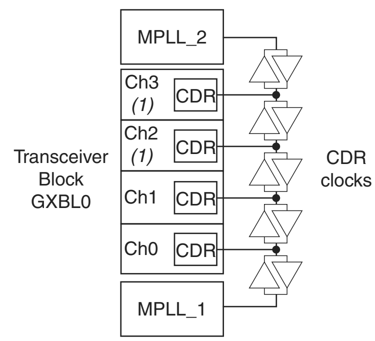
Table 11–3. Document Revision History

Date	Version	Changes
May 2013	1.3	Updated Note (4) in Table 11–1.
July 2010	1.2	■ Updated for the Quartus II software version 10.0 release. ■ Updated "I/O Pins Remain Tri-stated During Power-Up" section. ■ Updated Table 11–1.
February 2010	1.1	Updated Table 11–1 and Table 11–2 for the Quartus II software version 9.1 SP1 release.
November 2009	1.0	Initial release.

The CDR unit in each receiver channel gets the CDR clocks from one of the two multipurpose PLLs directly adjacent to the transceiver block. The CDR clocks distribution network is segmented by bidirectional tri-state buffers as shown in Figure 1–29 and Figure 1–30. This requires the CDR clocks from either one of the two multipurpose PLLs to drive a number of contiguous segmented paths to reach the intended receiver channel. Interleaving the CDR clocks from the two multipurpose PLLs is not supported.

For example, based on Figure 1–29, a combination of `MPLL_1` driving receiver channels 0, 1, and 3, while `MPLL_2` driving receiver channel 2 is not supported. In this case, only one multipurpose PLL can be used for the receiver channels.

Figure 1–29. CDR Clocking for Transceiver Channels in F324 and Smaller Packages



Note to Figure 1–29:

(1) Transceiver channels 2 and 3 are not available for devices in F169 and smaller packages.

Figure 1–30. CDR Clocking for Transceiver Channels in F484 and Larger Packages

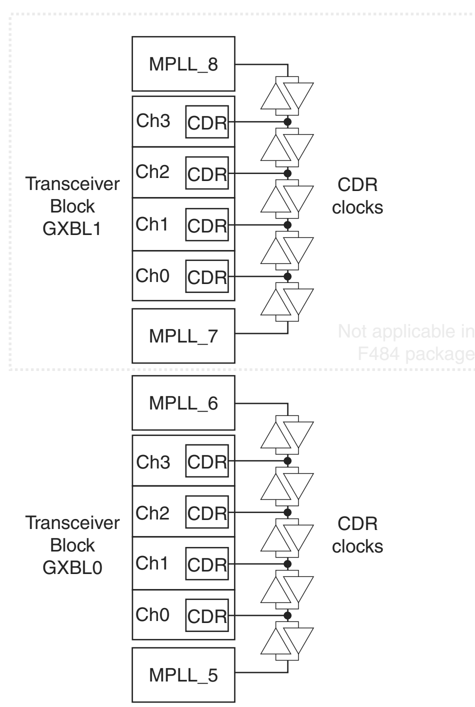
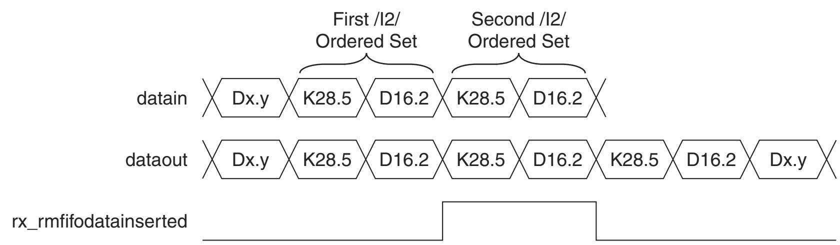


Figure 1–59 shows an example of rate match FIFO insertion in the case where one symbol must be inserted. Because the rate match FIFO can only insert /I2/ ordered sets, it inserts one /I2/ ordered set (two symbols inserted).

Figure 1–59. Example of Rate Match FIFO Insertion in GIGE Mode



☞ 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 two 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.

Serial RapidIO Mode

Serial RapidIO mode provides the non-bonded (×1) transceiver channel datapath configuration for SRIO protocol implementation. The Cyclone IV GX transceiver provides the PMA and the following PCS functions:

- 8B/10B encoding and decoding
- lane synchronization state machine

☞ Cyclone IV GX transceivers do not have built-in support for some PCS functions such as pseudo-random idle sequence generation and lane alignment in ×4 bonded channel configuration. If required, you must implement these functions in a user logics or external circuits.

The RapidIO Trade Association defines a high-performance, packet-switched interconnect standard to pass data and control information between microprocessors, digital signals, communications, network processes, system memories, and peripheral devices. The SRIO physical layer specification defines serial protocol running at 1.25 Gbps, 2.5 Gbps, and 3.125 Gbps in either single-lane (×1) or bonded four-lane (×4) at each line rate. Cyclone IV GX transceivers support single-lane (×1) configuration at all three line rates. Four ×1 channels configured in Serial RapidIO mode can be instantiated to achieve one non-bonded ×4 SRIO link. When implementing four ×1 SRIO channels, the receivers do not have lane alignment or deskew capability.

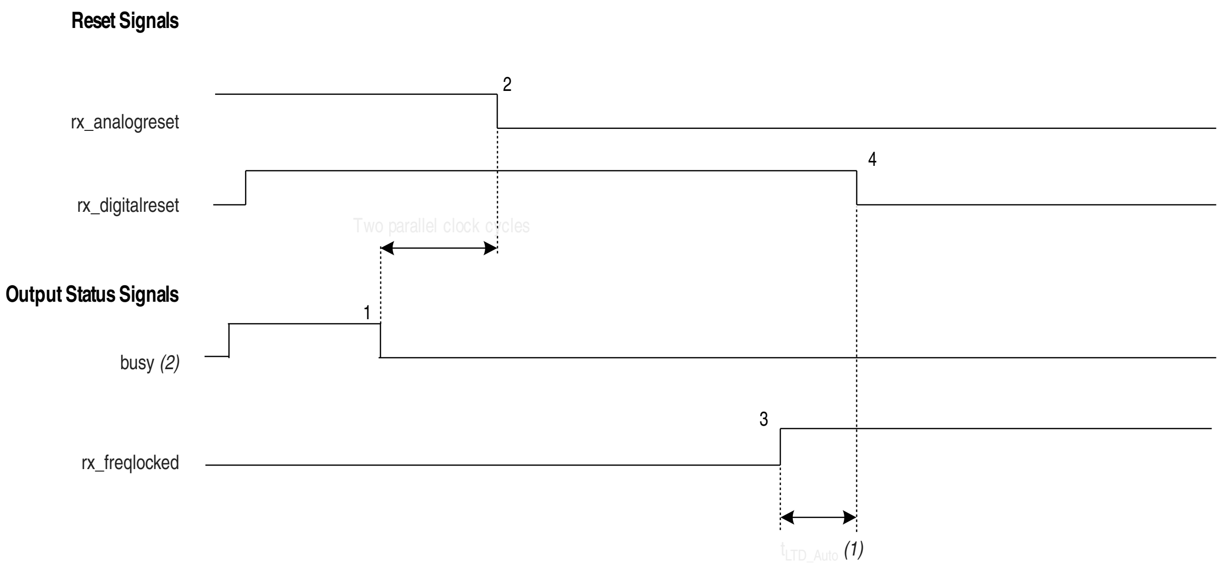
Transmitter Only Channel

This configuration contains only a transmitter channel. If you create a **Transmitter Only** instance in the ALTGX MegaWizard Plug-In Manager, use the same reset sequence shown in Figure 2–3 on page 2–7.

Receiver Only Channel—Receiver CDR in Automatic Lock Mode

This configuration contains only a receiver channel. If you create a **Receiver Only** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in automatic lock mode, use the reset sequence shown in Figure 2–6.

Figure 2–6. Sample Reset Sequence of Receiver Only Channel—Receiver CDR in Automatic Lock Mode



Notes to Figure 2–6:

(1) For t_{LTD_Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.

(2) The `busy` signal is asserted and deasserted only during initial power up when offset cancellation occurs. In subsequent reset sequences, the `busy` signal is asserted and deasserted only if there is a read or write operation to the ALTGX_RECONFIG megafunction.

As shown in Figure 2–6, perform the following reset procedure for the receiver in CDR automatic lock mode:

1. After power up, wait for the `busy` signal to be deasserted.
2. Keep the `rx_digitalreset` and `rx_analogreset` signals asserted during this time period.
3. After the busy signal is deasserted, wait for another two parallel clock cycles, then deassert the `rx_analogreset` signal.
4. Wait for the `rx_freqlocked` signal to go high.
5. When `rx_freqlocked` goes high (marker 3), from that point onwards, wait for at least t_{LTD_Auto}, then de-assert the `rx_digitalreset` signal (marker 4). At this point, the receiver is ready to receive data.

Table 1–20. Differential I/O Standard Specifications for Cyclone IV Devices *(1)* **(Part 2 of 2)**

I/O Standard	V_{CCIO} (V)			V_{ID} (mV)		V_{IcM} (V) *(2)*			V_{OD} (mV) *(3)*			V_{OS} (V) *(3)*		
	Min	**Typ**	**Max**	**Min**	**Max**	**Min**	**Condition**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
LVDS (Column I/Os)	2.375	2.5	2.625	100	—	0.05	$D_{MAX} \leq$ 500 Mbps	1.80	247	—	600	1.125	1.25	1.375
						0.55	500 Mbps $\leq D_{MAX} \leq$ 700 Mbps	1.80						
						1.05	D_{MAX} > 700 Mbps	1.55						
BLVDS (Row I/Os) *(4)*	2.375	2.5	2.625	100	—	—	—	—	—	—	—	—	—	—
BLVDS (Column I/Os) *(4)*	2.375	2.5	2.625	100	—	—	—	—	—	—	—	—	—	—
mini-LVDS (Row I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	300	—	600	1.0	1.2	1.4
mini-LVDS (Column I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	300	—	600	1.0	1.2	1.4
RSDS® (Row I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.5
RSDS (Column I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.5
PPDS (Row I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.4
PPDS (Column I/Os) *(5)*	2.375	2.5	2.625	—	—	—	—	—	100	200	600	0.5	1.2	1.4

Notes to Table 1–20:

(1) For an explanation of terms used in Table 1–20, refer to “Glossary” on page 1–37.

(2) V_{IN} range: 0 V $\leq V_{IN} \leq$ 1.85 V.

(3) R_L range: 90 $\leq R_L \leq$ 110 Ω .

(4) There are no fixed V_{IN}, V_{OD}, and V_{OS} specifications for BLVDS. They depend on the system topology.

(5) The Mini-LVDS, RSDS, and PPDS standards are only supported at the output pins.

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

Table 1–44 and Table 1–45 list the IOE programmable delay for Cyclone IV GX devices.

Table 1–44. IOE Programmable Delay on Column Pins for Cyclone IV GX Devices (1), (2)

Parameter	Paths Affected	Number of Settings	Min Offset	Max Offset						Unit
				Fast Corner		Slow Corner				
				C6	I7	C6	C7	C8	I7	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	1.313	1.209	2.184	2.336	2.451	2.387	ns
Input delay from pin to input register	Pad to I/O input register	8	0	1.312	1.208	2.200	2.399	2.554	2.446	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.438	0.404	0.751	0.825	0.886	0.839	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.713	0.682	1.228	1.41	1.566	1.424	ns

Notes to Table 1–44:

(1) The incremental values for the settings are generally linear. For exact values of each setting, use the latest version of the Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software.

Table 1–45. IOE Programmable Delay on Row Pins for Cyclone IV GX Devices (1), (2)

Parameter	Paths Affected	Number of Settings	Min Offset	Max Offset						Unit
				Fast Corner		Slow Corner				
				C6	I7	C6	C7	C8	I7	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	1.314	1.210	2.209	2.398	2.526	2.443	ns
Input delay from pin to input register	Pad to I/O input register	8	0	1.313	1.208	2.205	2.406	2.563	2.450	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.461	0.421	0.789	0.869	0.933	0.884	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.712	0.682	1.225	1.407	1.562	1.421	ns

Notes to Table 1–45:

(1) The incremental values for the settings are generally linear. For exact values of each setting, use the latest version of Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software