5. Clock Networks and PLLs in the Cyclone III Device Family

CIII51006-4.1

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) with advanced features in the Cyclone® III device family (Cyclone III and Cyclone III LS devices).

This chapter includes the following sections:

Clock Networks

The Cyclone III device family provides up to 16 dedicated clock pins (`CLK[15..0]`) that can drive the global clocks (GCLKs). The Cyclone III device family supports four dedicated clock pins on each side of the device except EP3C5 and EP3C10 devices. EP3C5 and EP3C10 devices only support four dedicated clock pins on the left and right sides of the device.

For more information about the number of GCLK networks in each device density, refer to the *Cyclone III Device Family Overview* chapter.

GCLK Network

GCLKs drive throughout the entire device, feeding all device quadrants. All resources in the device (I/O elements, logic array blocks (LABs), dedicated multiplier blocks, and M9K memory blocks) can use GCLKs as clock sources. Use these clock network resources for control signals, such as clock enables and clears fed by an external pin. Internal logic can also drive GCLKs for internally generated GCLKs and asynchronous clears, clock enables, or other control signals with high fan-out.

Table 5–1 lists the connectivity of the clock sources to the GCLK networks.

Table 5–1. Cyclone III Device Family GCLK Network Connections (Part 1 of 2)

GCLK Network Clock Sources	GCLK Networks (1)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CLK0/DIFFCLK_0p	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
CLK1/DIFFCLK_0n	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
CLK2/DIFFCLK_1p	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
CLK3/DIFFCLK_1n	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
CLK4/DIFFCLK_2p	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
CLK5/DIFFCLK_2n	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
CLK6/DIFFCLK_3p	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—
CLK7/DIFFCLK_3n	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
CLK8/DIFFCLK_5n (2)	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—
CLK9/DIFFCLK_5p (2)	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
CLK10/DIFFCLK_4n (2)	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—
CLK11/DIFFCLK_4p (2)	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
CLK12/DIFFCLK_7n (2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓
CLK13/DIFFCLK_7p (2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—
CLK14/DIFFCLK_6n (2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓
CLK15/DIFFCLK_6p (2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL1_C0 (3)	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C1 (3)	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C2 (3)	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C3 (3)	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C4 (3)	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL2_C0 (3)	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
PLL2_C1 (3)	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—
PLL2_C2 (3)	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
PLL2_C3 (3)	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—
PLL2_C4 (3)	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
PLL3_C0	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
PLL3_C1	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—
PLL3_C2	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—
PLL3_C3	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—
PLL3_C4	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—
PLL4_C0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—

Table 5–1. Cyclone III Device Family GCLK Network Connections (Part 2 of 2)

GCLK Network Clock Sources	GCLK Networks (1)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
PLL4_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓
PLL4_C2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL4_C3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—
PLL4_C4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓
DPCLK0	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK1	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK7 (4) CDPCLK0, or CDPCLK7 (2), (5)	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK2 (4) CDPCLK1, or CDPCLK2 (2), (5)	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK5 (4) DPCLK7 (2)	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK4 (4) DPCLK6 (2)	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK6 (4) CDPCLK5, or CDPCLK6 (2), (5)	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK3 (4) CDPCLK4, or CDPCLK3 (2), (5)	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—
DPCLK8	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
DPCLK11	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
DPCLK9	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
DPCLK10	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—
DPCLK5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
DPCLK2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
DPCLK4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
DPCLK3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓

Notes to Table 5–1:

(1) EP3C5 and EP3C10 devices only have GCLK networks 0 to 9.
(2) These pins apply to all devices in the Cyclone III device family except EP3C5 and EP3C10 devices.
(3) EP3C5 and EP3C10 devices only have phase-locked loops (PLLs) 1 and 2.
(4) This pin applies only to EP3C5 and EP3C10 devices.
(5) Only one of the two CDPCLK pins can feed the clock control block. You can use the other pin as a regular I/O pin.

If you do not use dedicated clock pins to feed the GCLKs, you can use them as general-purpose input pins to feed the logic array. However, when using them as general-purpose input pins, they do not have support for an I/O register and must use LE-based registers in place of an I/O register.

For more information about how to connect the clock and PLL pins, refer to the *Cyclone III Device Family Pin Connection Guidelines* on the Altera® website.

Clock Control Block

The clock control block drives GCLKs. Clock control blocks are located on each side of the device, close to the dedicated clock input pins. GCLKs are optimized for minimum clock skew and delay.

Table 5–2 lists the sources that can feed the clock control block, which in turn feeds the GCLKs.

Table 5–2. Clock Control Block Inputs

Input	Description
Dedicated clock inputs	Dedicated clock input pins can drive clocks or global signals, such as synchronous and asynchronous clears, presets, or clock enables onto given GCLKs.
Dual-purpose clock (DPCLK and CDPCLK) I/O input	DPCLK and CDPCLK I/O pins are bidirectional dual function pins that are used for high fan-out control signals, such as protocol signals, TRDY and IRDY signals for PCI, via the GCLK. Clock control blocks that have inputs driven by dual-purpose clock I/O pins are not able to drive PLL inputs.
PLL outputs	PLL counter outputs can drive the GCLK.
Internal logic	You can drive the GCLK through logic array routing to enable internal logic elements (LEs) to drive a high fan-out, low-skew signal path. Clock control blocks that have inputs driven by internal logic are not able to drive PLL inputs.

In the Cyclone III device family, dedicated clock input pins, PLL counter outputs, dual-purpose clock I/O inputs, and internal logic can all feed the clock control block for each GCLK.

Normal I/O pins cannot drive the PLL input clock port.

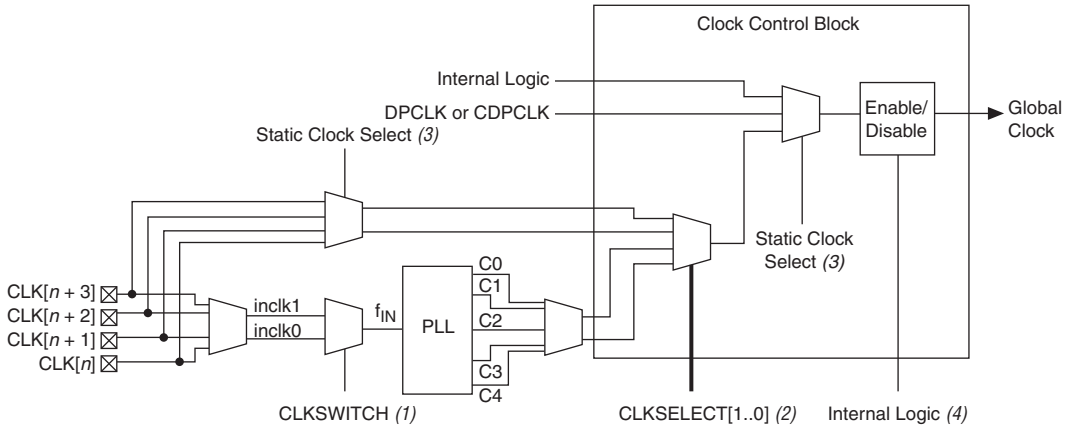
The output from the clock control block in turn feeds the corresponding GCLK. The GCLK can drive the PLL input if the clock control block inputs are outputs of another PLL or dedicated clock input pins. The clock control blocks are at the device periphery; there are a maximum of 20 clock control blocks available per Cyclone III device family.

The control block has two functions:

- Dynamic GCLK clock source selection (not applicable for DPCLK or CDPCLK and internal logic input)
- GCLK network power down (dynamic enable and disable)

Figure 5–1 shows the clock control block.

Figure 5–1. Clock Control Block



Notes to Figure 5–1:

(1) The `clkswitch` signal can either be set through the configuration file or dynamically set when using the manual PLL switchover feature. The output of the multiplexer is the input clock (f_{IN}) for the PLL.

(2) The `clkselect[1..0]` signals are fed by internal logic and is used to dynamically select the clock source for the GCLK when the device is in user mode.

(3) The static clock select signals are set in the configuration file. Therefore, dynamic control when the device is in user mode is not feasible.

(4) You can use internal logic to enable or disable the GCLK in user mode.

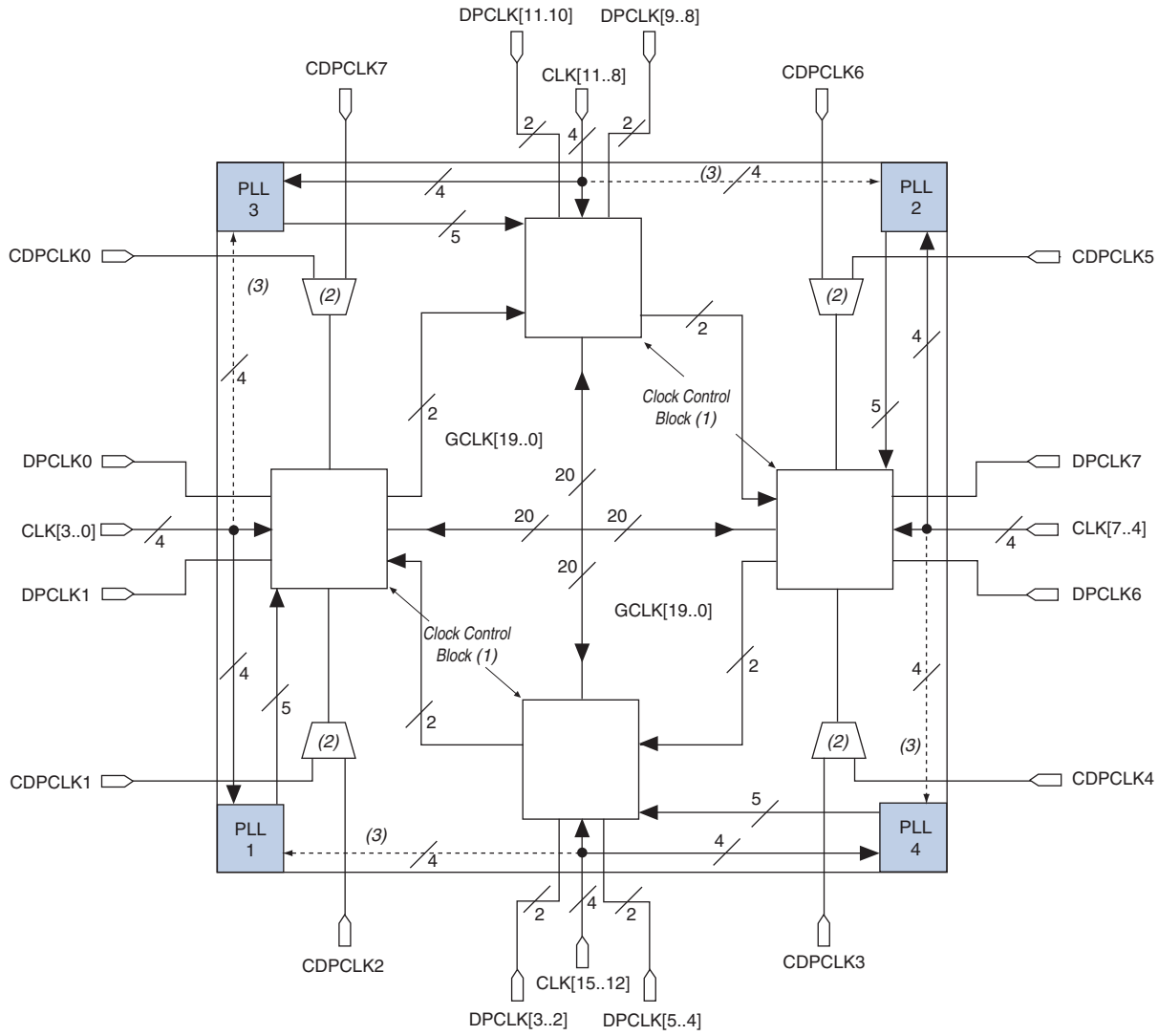
Each PLL generates five clock outputs through the `c[4..0]` counters. Two of these clocks can drive the GCLK through a clock control block, as shown in Figure 5–1.

For more information about how to use the clock control block in the Quartus® II software, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

GCLK Network Clock Source Generation

Figure 5–2 shows Cyclone III device family PLLs, clock inputs, and clock control block location for different device densities.

Figure 5–2. PLL, CLK[], DPCLK[], and Clock Control Block Locations in the Cyclone III Device Family *(1)*



Notes to Figure 5–2:

(1) There are five clock control blocks on each side.

(2) Only one of the corner CDPCLK pins in each corner can feed the clock control block at a time. You can use the other CDPCLK pins as general-purpose I/O pins.

(3) Remote clock pins can feed PLLs over dedicated clock paths. However, these paths are not fully compensated.

The inputs to the five clock control blocks on each side must be chosen from among the following clock sources:

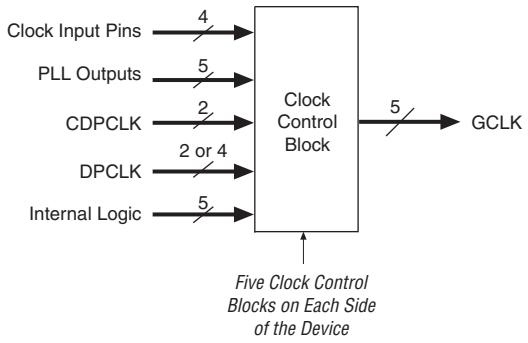
- Four clock input pins
- Five PLL counter outputs
- Two DPCLK pins and two CDPCLK pins from both the left and right sides, and four DPCLK pins and two CDPCLK pins from both the top and bottom
- Five signals from internal logic

From the clock sources listed above, only two clock input pins, two PLL clock outputs, one DPCLK or CDPCLK pin, and one source from internal logic can drive into any given clock control block, as shown in Figure 5–1 on page 5–5.

Out of these five inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

Figure 5–3 shows a simplified version of the five clock control blocks on each side of the Cyclone III device family periphery.

Figure 5–3. Clock Control Blocks on Each Side of the Cyclone III Device Family *(1)*



Note to Figure 5–3:

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

GCLK Network Power Down

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

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

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

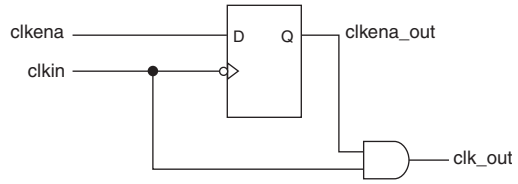
For more information, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

clkena Signals

The Cyclone III device family supports clkena signals at the GCLK network level. This allows you to gate-off the clock even when a PLL is used. Upon re-enabling the output clock, the PLL does not need a resynchronization or re-lock period because the circuit gates off the clock at the clock network level. In addition, the PLL can remain locked independent of the clkena signals because the loop-related counters are not affected.

Figure 5–4 shows how to implement the clkena signal.

Figure 5–4. clkena Implementation

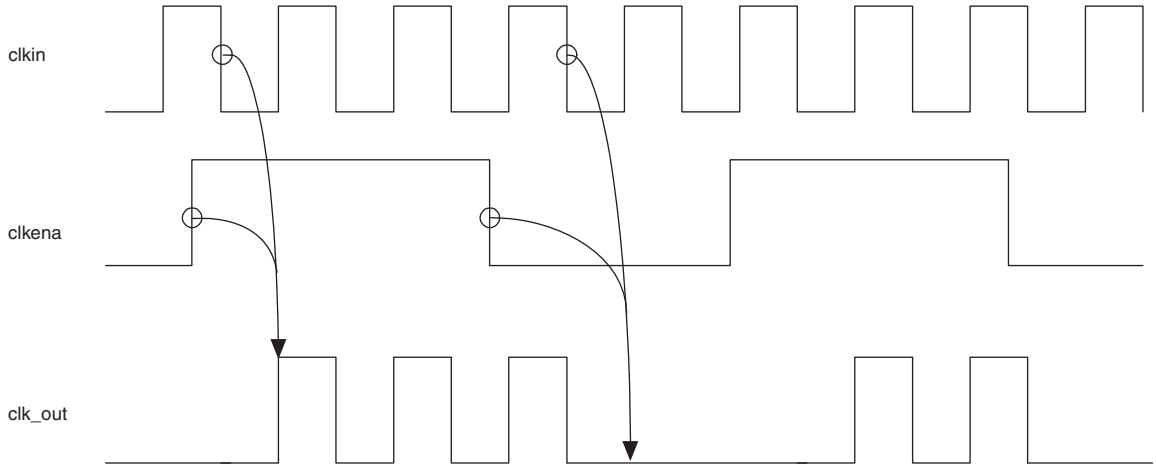


The clkena circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5–4.

Figure 5–5 shows the waveform example for a clock output enable. The clkena signal is sampled on the falling edge of the clock (clkin).

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

Figure 5–5. 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 deasserting 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.

PLLs in the Cyclone III Device Family

The Cyclone III device family offers up to four 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 PLLs in each device density, refer to the *Cyclone III Device Family Overview* chapter.

The Cyclone III device family PLLs have the same core analog structure.

Table 5–3 lists the features available in the Cyclone III device family PLLs.

Table 5–3. Cyclone III Device Family PLL Hardware Features

Hardware Features	Availability
C (output counters)	5
M, N, C counter sizes	1 to 512 (1)
Dedicated clock outputs	1 single-ended or 1 differential pair
Clock input pins	4 single-ended or 2 differential pairs
Spread-spectrum input clock tracking	✓ (2)
PLL cascading	Through GCLK
Compensation modes	Source-Synchronous Mode, No Compensation Mode, Normal Mode, and Zero Delay Buffer Mode
Phase shift resolution	Down to 96-ps increments (3)
Programmable duty cycle	✓
Output counter cascading	✓
Input clock switchover	✓
User mode reconfiguration	✓
Loss of lock detection	✓

Notes to Table 5–3:

(1) C counters range from 1 through 512 if the output clock uses a 50% duty cycle. For any output clocks using a non-50% duty cycle, the post-scale counters range from 1 through 256.

(2) Only applicable if the input clock jitter is in the input jitter tolerance specifications.

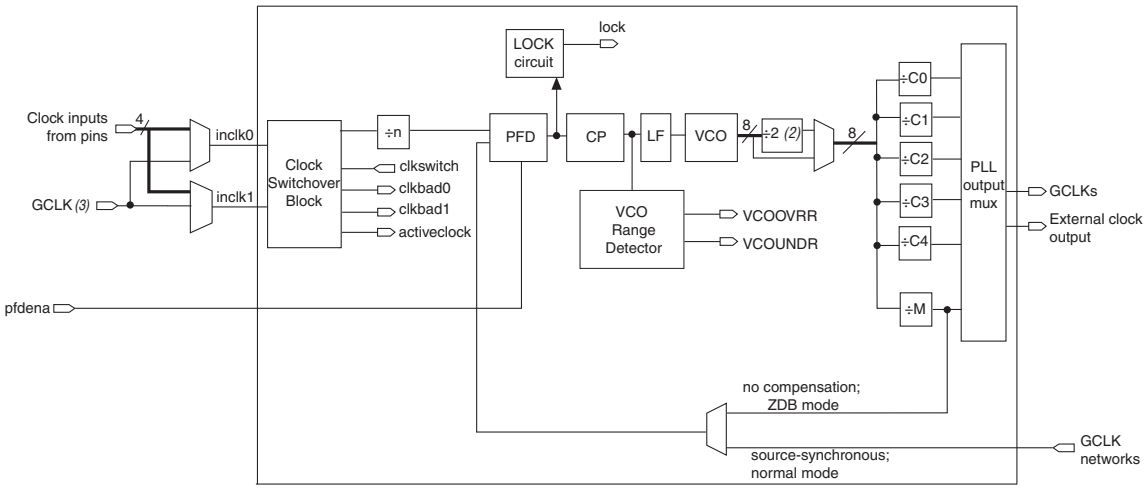
(3) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Cyclone III device family can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

Cyclone III Device Family PLL Hardware Overview

This section gives a hardware overview of the Cyclone III device family PLL.

Figure 5–6 shows a simplified block diagram of the major components of the PLL of the Cyclone III device family.

Figure 5–6. Cyclone III Device Family PLL Block Diagram *(1)*



Notes to Figure 5–6:

(1) Each clock source can come from any of the four clock pins located on the same side of the device as the PLL.

(2) This is the VCO post-scale counter K.

(3) This input port is fed by a pin-driven dedicated GCLK, or through a clock control block if the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK. An internally generated global signal cannot drive the PLL.

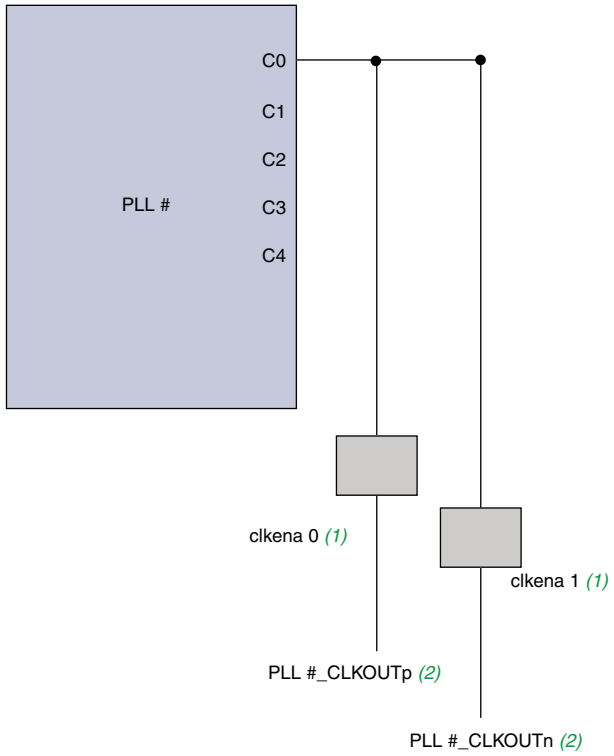
The VCO post-scale counter K is used to divide the supported VCO range by two. The VCO frequency reported by the Quartus II software in the PLL summary section of the compilation report takes into consideration the VCO post-scale counter value. Therefore, if the VCO post-scale counter has a value of 2, the frequency reported is lower than the f_{VCO} specification specified in the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

External Clock Outputs

Each PLL of the Cyclone III device family supports one single-ended clock output or one differential clock output. Only the C0 output counter can feed the dedicated external clock outputs, as shown in Figure 5–7, without going through the GCLK. Other output counters can feed other I/O pins through the GCLK.

Figure 5–7 shows the external clock outputs for PLLs.

Figure 5–7. External Clock Outputs for PLLs



Notes to Figure 5–7:

(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 or one differential clock output.

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 (in the top and bottom banks) as well as LVDS, LVPECL, differential HSTL, and differential SSTL.

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

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

Clock Feedback Modes

Cyclone III device family PLLs support up to four different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle.

Input and output delays are fully compensated by the PLL only when you are using the dedicated clock input pins associated with a given PLL as the clock sources. For example, when using PLL1 in normal mode, the clock delays from the input pin to the PLL and the PLL clock output-to-destination register are fully compensated, provided that the clock input pin is one of the following four pins:

- CLK0
- CLK1
- CLK2
- CLK3

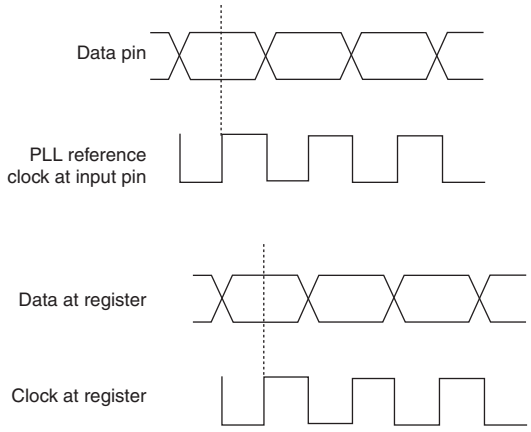
When driving the PLL using the GCLK network, the input and output delays may not be fully compensated in the Quartus II software.

Source-Synchronous Mode

If the data and clock arrive at the same time at the input pins, the phase relationship between the data and clock remains the same at the data and clock ports of any I/O element input register.

Figure 5–8 shows an example waveform of the data and clock in this mode. Use this mode for source-synchronous data transfers. Data and clock signals at the I/O element experience similar buffer delays as long as the same I/O standard is used.

Figure 5–8. Phase Relationship Between Data and Clock in Source-Synchronous Mode



Source-synchronous mode compensates for delay of the clock network used, including any difference in the delay between the following two paths:

- Data pin to I/O element register input
- Clock input pin to the PLL phase-frequency detector (PFD) input

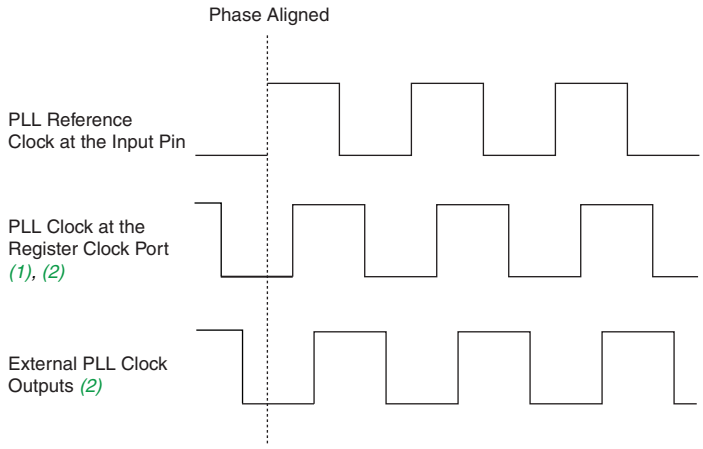
Set the input pin to the register delay chain in the I/O element to zero in the Quartus II software for all data pins clocked by a source-synchronous mode PLL. Also, all data pins must use the **PLL COMPENSATED logic** option in the Quartus II software.

No Compensation Mode

In no compensation mode, the PLL does not compensate for any clock networks. This provides better jitter performance because clock feedback into the PFD does not pass through as much circuitry. Both the PLL internal and external clock outputs are phase-shifted with respect to the PLL clock input.

Figure 5–9 shows a waveform example of the phase relationship of the PLL clock in this mode.

Figure 5–9. Phase Relationship Between PLL Clocks in No Compensation Mode



Notes to Figure 5–9:

(1) Internal clocks fed by the PLL are phase-aligned to each other.

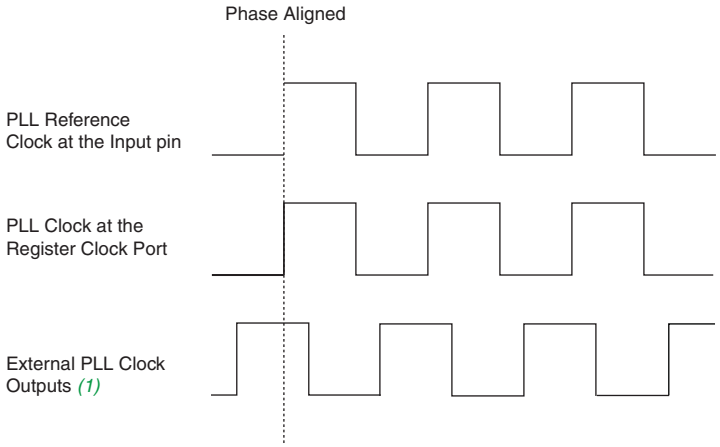
(2) The PLL clock outputs can lead or lag the PLL input clocks.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock output pin has a phase delay relative to the clock input pin if connected in this mode. The Quartus II software timing analyzer reports any phase difference between the two. In normal mode, the PLL fully compensates the delay introduced by the GCLK network.

Figure 5–10 shows a waveform example of the phase relationship of the PLL clocks in this mode.

Figure 5–10. Phase Relationship Between PLL Clocks in Normal Mode



Note to Figure 5–10:

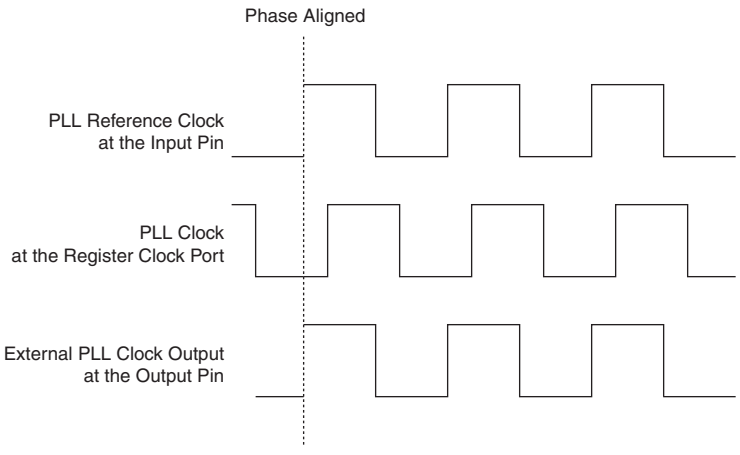
(1) The external clock output can lead or lag the PLL internal clock signals.

Zero Delay Buffer Mode

In zero delay buffer (ZDB) mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. When using this mode, use the same I/O standard on the input clock and output clocks to guarantee clock alignment at the input and output pins.

Figure 5–11 shows an example waveform of the phase relationship of the PLL clocks in ZDB mode.

Figure 5–11. Phase Relationship Between PLL Clocks in ZDB Mode



Hardware Features

Cyclone III device family PLLs support several features for general-purpose clock management. This section discusses clock multiplication and division implementation, phase shifting implementations, and programmable duty cycles.

Clock Multiplication and Division

Each Cyclone III device family PLL provides clock synthesis for PLL output ports using M/(N*post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, N, and is then multiplied by the M feedback factor. The control loop drives the VCO to match f_{IN} (M/N). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO value is the least common multiple of the output frequencies that meets its frequency specifications. For example, if output frequencies required from one PLL are 33 and 66 MHz, the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz in the VCO range). Then, the post-scale counters scale down the VCO frequency for each output port.

There is one pre-scale counter, N, and one multiply counter, M, per PLL, with a range of 1 to 512 for both M and N. The N counter does not use duty cycle control because the purpose of this counter is only to calculate frequency division. There are five generic post-scale counters per PLL that can feed GCLKs or external clock outputs. These post-scale counters range from 1 to 512 with a 50% duty cycle setting. The post-scale counters range from 1 to 256 with any non-50% duty cycle setting. The sum of the high/low count values chosen for a design selects the divide value for a given counter.

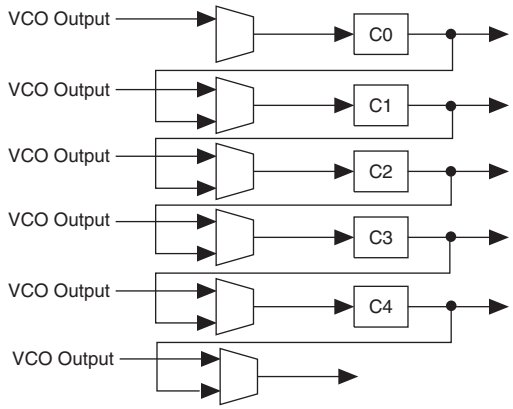
The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL megafunction.

Phase alignment between output counters are determined using the t_{PLL_PSERR} specification.

Post-Scale Counter Cascading

Cyclone III device family PLLs support post-scale counter cascading to create counters larger than 512. This is implemented by feeding the output of one C counter into the input of the next C counter, as shown in Figure 5–12.

Figure 5–12. Counter Cascading



When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings.

For example, if C0 = 4 and C1 = 2, the cascaded value is C0 × C1 = 8.

Post-scale counter cascading is automatically set by the Quartus II software in the configuration file. Post-scale counter cascading cannot be performed using the PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. You can achieve the duty cycle setting by a low and high time count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. For example, if the C0 counter is 10, steps of 5% are possible for duty cycle choices between 5 to 90%.

Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

PLL Control Signals

You can use the following three signals to observe and control the PLL operation and resynchronization.

pfdena

Use the `pfdena` signal to maintain the last locked frequency so that your system has time to store its current settings before shutting down. The `pfdena` signal controls the PFD output with a programmable gate. If you disable the PFD, the VCO operates at its last set value of control voltage and frequency with some long-term drift to a lower frequency.

areset

The `areset` signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When driven high, the PLL counters reset, clearing the PLL output and placing the PLL out of lock. The VCO is then set back to its nominal setting. When driven low again, the PLL resynchronizes to its input as it re-locks.

You must include the `areset` signal in your designs if one of the following conditions is true:

- PLL reconfiguration or clock switchover enabled in your design
- Phase relationships between the PLL input clock and output clocks must be maintained after a loss-of-lock condition

If the input clock to the PLL is toggling or unstable upon power up, assert the `areset` signal after the input clock is stable and within specifications.

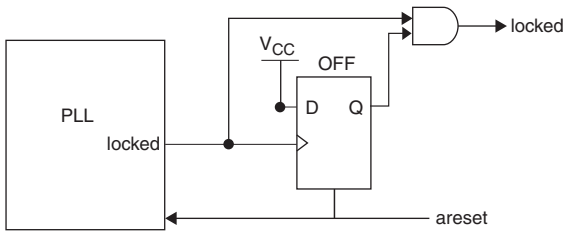
locked

The `locked` output indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the Quartus II MegaWizard™ Plug-in Manager.

Altera recommends that you use the `areset` and `locked` signals in your designs to control and observe the status of your PLL.

This implementation is illustrated in Figure 5–13.

Figure 5–13. Locked Signal Implementation



If you use the SignalTap® II tool to probe the `locked` signal before the D flip-flop, the `locked` signal goes low only when `areset` is deasserted. If the `areset` signal is not enabled, the extra logic is not implemented in the ALTPLL megafunction.

For more information about the PLL control signals, refer to the *Phase-Locked Loop (ALTPLL) Megafunction User Guide*.

Clock Switchover

The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application, such as a system that turns on the redundant clock if the previous clock stops running. Your design can automatically perform clock switchover when the clock is no longer toggling, or based on the user control signal, `clkswitch`.

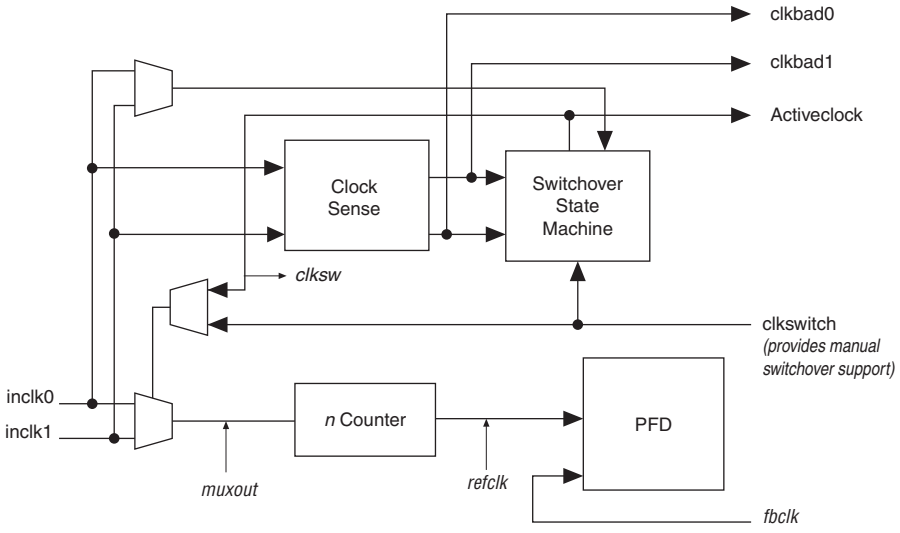
Automatic Clock Switchover

Cyclone III device family PLLs support a fully configurable clock switchover capability.

When the current reference clock is not present, the clock-sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad[0]`, `clkbad[1]`, and `activeclock`—from the PLL to implement a custom switchover circuit. You can select a clock source at the backup clock by connecting it to the `inclk1` port of the PLL in your design.

Figure 5–14 shows the block diagram of the switchover circuit built into the PLL.

Figure 5–14. Automatic Clock Switchover Circuit

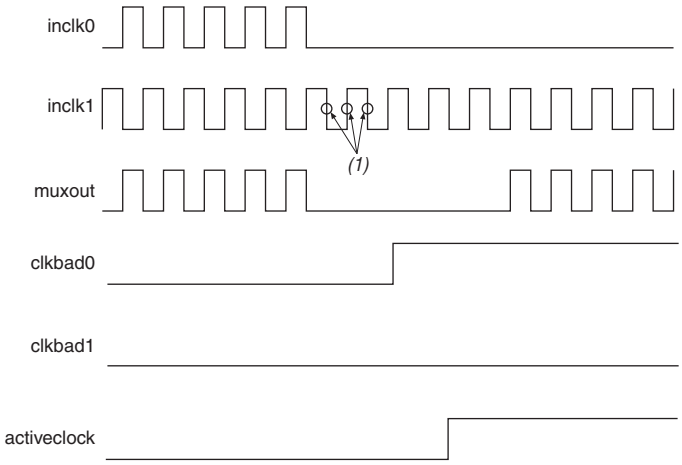


There are two ways to use the clock switchover feature:

- Use the switchover circuitry for switching from inclk0 to inclk1 running at the same frequency. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal that controls the multiplexer select input shown in Figure 5–14. In this case, inclk1 becomes the reference clock for the PLL. This automatic switchover can switch back and forth between the inclk0 and inclk1 clocks any number of times, when one of the two clocks fails and the other clock is available.
- Use the clkswitch input for user- or system-controlled switch conditions. This is possible for same-frequency switchover or to switch between inputs of different frequencies. For example, if inclk0 is 66 MHz and inclk1 is 200 MHz, you must control the switchover because the automatic clock-sense circuitry cannot monitor primary and secondary clock frequencies with a frequency difference of more than 20%. This feature is useful when clock sources can originate from multiple cards on the backplane, requiring a system-controlled switchover between frequencies of operation. Choose the secondary clock frequency so the VCO operates in the recommended frequency range. Also, set the M, N, and C counters accordingly to keep the VCO operating frequency in the recommended range.

Figure 5–15 shows a waveform example of the switchover feature when using automatic loss of clock detection. Here, the inclk0 signal remains low. After the inclk0 signal remains low for approximately two clock cycles, the clock-sense circuitry drives the clkbad[0] signal high. Also, because the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the clksw signal to switch to inclk1.

Figure 5–15. Automatic Switchover Upon Clock Loss Detection *(1)*



Note to Figure 5–15:

(1) Switchover is enabled on the falling edge of inclk0 or inclk1, depending on which clock is available. In this figure, switchover is enabled on the falling edge of inclk1.

Manual Override

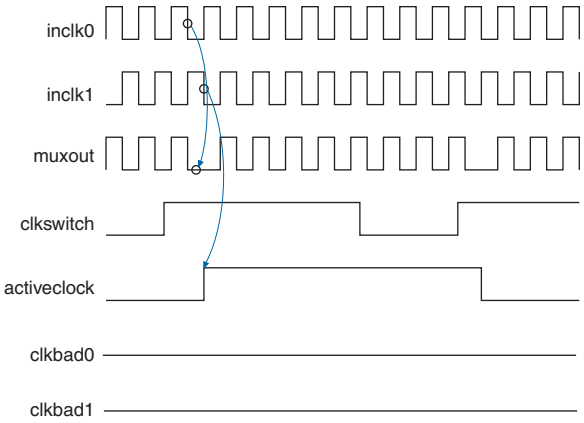
If you are using the automatic switchover, you must switch input clocks with the manual override feature with the clkswitch input.

Figure 5–16 shows an example of a waveform illustrating the switchover feature when controlled by clkswitch. In this case, both clock sources are functional and inclk0 is selected as the reference clock. A low-to-high transition of the clkswitch signal starts the switchover sequence. The clkswitch signal must be high for at least three clock cycles (at least three of the longer clock period if inclk0 and inclk1 have different frequencies). On the falling edge of inclk0, the reference clock of the counter, muxout, is gated off to prevent any clock glitching. On the falling edge of inclk1, the reference clock multiplexer switches from inclk0 to inclk1 as the PLL reference. On the falling edge of inclk1, the reference clock multiplexer switches from inclk0 to inclk1 as the PLL reference, and the activeclock signal changes to indicate which clock is currently feeding the PLL.

In this mode, the activeclock signal mirrors the clkswitch signal. As both blocks are still functional during the manual switch, neither clkbad signals go high. Because the switchover circuit is positive edge-sensitive, the falling edge of the clkswitch signal does not cause the circuit to switch back from inclk1 to inclk0. When the clkswitch signal goes high again, the process repeats. The clkswitch signal and the automatic switch only works depending on the availability of the clock that is switched to. If the clock is unavailable, the state machine waits until the clock is available.

If CLKSWITCH = 1, the automatic switchover function is overridden. While the clkswitch signal is high, further switchover action is blocked.

Figure 5–16. Clock Switchover Using the clkswitch Control *(1)*



Note to Figure 5–16:

(1) Both inclk0 and inclk1 must be running when the clkswitch signal goes high to start a manual clock switchover event.

Manual Clock Switchover

Cyclone III device family PLLs support manual switchover, in which the clkswitch signal controls whether inclk0 or inclk1 is the input clock to the PLL. The characteristics of a manual switchover is similar to the manual override feature in an automatic clock switchover, in which the switchover circuit is edge-sensitive. When the clkswitch signal goes high, the switchover sequence starts. The falling edge of the clkswitch signal does not cause the circuit to switch back to the previous input clock.

For more information about PLL software support in the Quartus II software, refer to the *Phase-Locked Loop (ALTPLL) Megafunction User Guide*.

Guidelines

Use the following guidelines to design with clock switchover in PLLs:

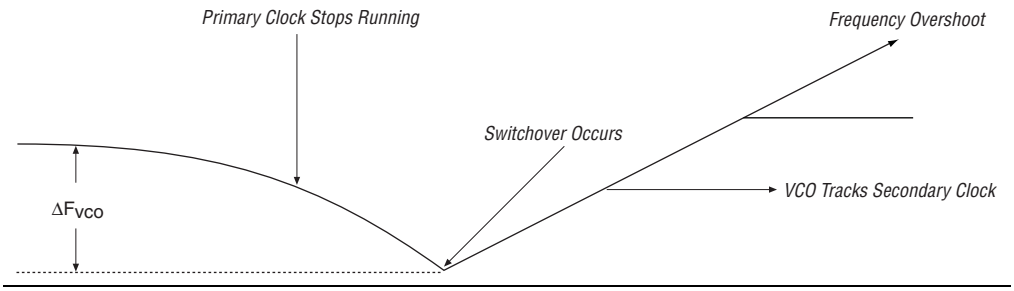
- Clock loss detection and automatic clock switchover requires that the inclk0 and inclk1 frequencies be within 20% of each other. Failing to meet this requirement causes the clkbad[0] and clkbad[1] signals to function improperly.
- When using manual clock switchover, the difference between inclk0 and inclk1 can be more than 20%. However, differences between the two clock sources (frequency, phase, or both) can cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between the input and output clocks.

Both inclk0 and inclk1 must be running when the clkswitch signal goes high to start the manual clock switchover event. Failing to meet this requirement causes the clock switchover to malfunction.

- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. When referencing input clock changes, the low-bandwidth PLL reacts slower than a high-bandwidth PLL. When the switchover happens, the low-bandwidth PLL propagates the stopping of the clock to the output slower than the high-bandwidth PLL. The low-bandwidth PLL filters out jitter on the reference clock. However, you must be aware that the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock is dependent on the PLL configuration.
- If the phase relationship between the input clock to the PLL and output clock from the PLL is important in your design, assert areset for 10 ns after performing a clock switchover. Wait for the locked signal (or gated lock) to go high before re-enabling the output clocks from the PLL.

- Figure 5–17 shows how the VCO frequency gradually decreases when the primary clock is lost and then increases as the VCO locks on to the secondary clock. After the VCO locks on to the secondary clock, some overshoot can occur (an over-frequency condition) in the VCO frequency.

Figure 5–17. VCO Switchover Operating Frequency



- Disable the system during switchover if the system is not tolerant to frequency variations during the PLL resynchronization period. You can use the `clkbad[0]` and `clkbad[1]` status signals to turn off the PFD (`pfdena = 0`) so the VCO maintains its last frequency. You can also use the switchover state machine to switch over to the secondary clock. Upon enabling the PFD, output clock enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. After the lock indication is stable, the system can re-enable the output clock or clocks.

Programmable Bandwidth

The PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. Cyclone III device family PLLs 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 the Cyclone III device family. 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 purely based on counter settings, which are independent of process, voltage, and temperature.

You can phase shift the output clocks from the Cyclone III device family PLLs in either:

- Fine resolution using VCO phase taps, or
- 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

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

Note to Equation 5–1:

(1) 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 Φfine = 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

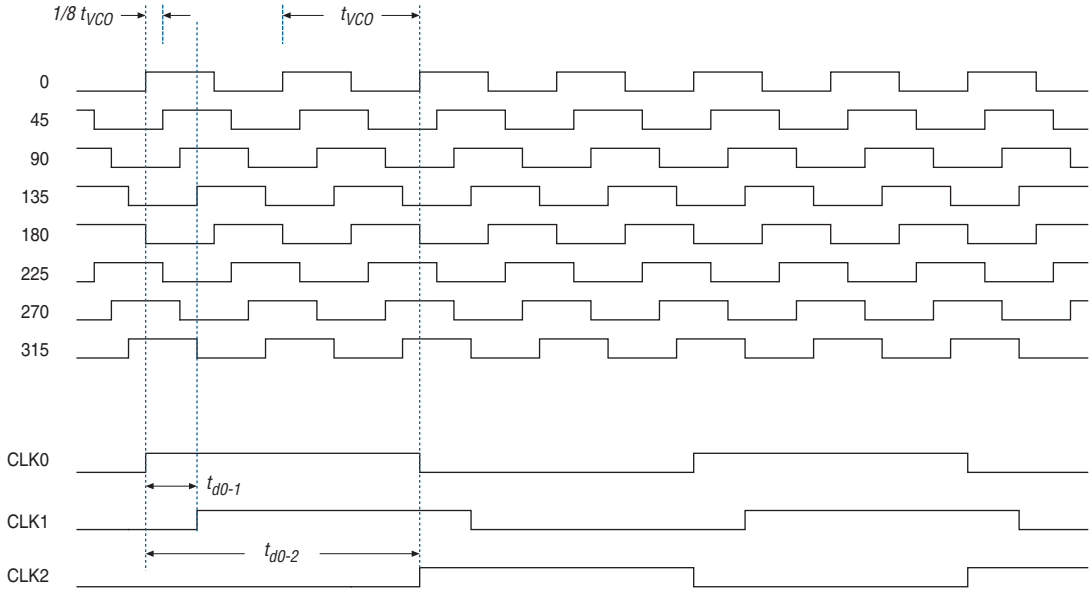
$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

Note to Equation 5–2:

(1) C is the count value set for the counter delay time—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.

Figure 5–18 shows an example of phase shift insertion using fine resolution through VCO phase taps method. The eight phases from the VCO are shown and labeled for reference. In this example, CLK0 is based on 0° phase from the VCO and has the C value for the counter set to one. The CLK1 signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. CLK1 is based on the 135° phase tap from the VCO and has the C value for the counter set to one. The CLK1 signal is also divided by four. In this case, the two clocks are offset by 3 Φ_{fine}. CLK2 is based on the 0° phase from the VCO but has the *C* value for the counter set to three. This creates a delay of two Φ_{coarse} (two complete VCO periods).

Figure 5–18. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use the coarse and fine phase shifts to implement clock delays in the Cyclone III device family.

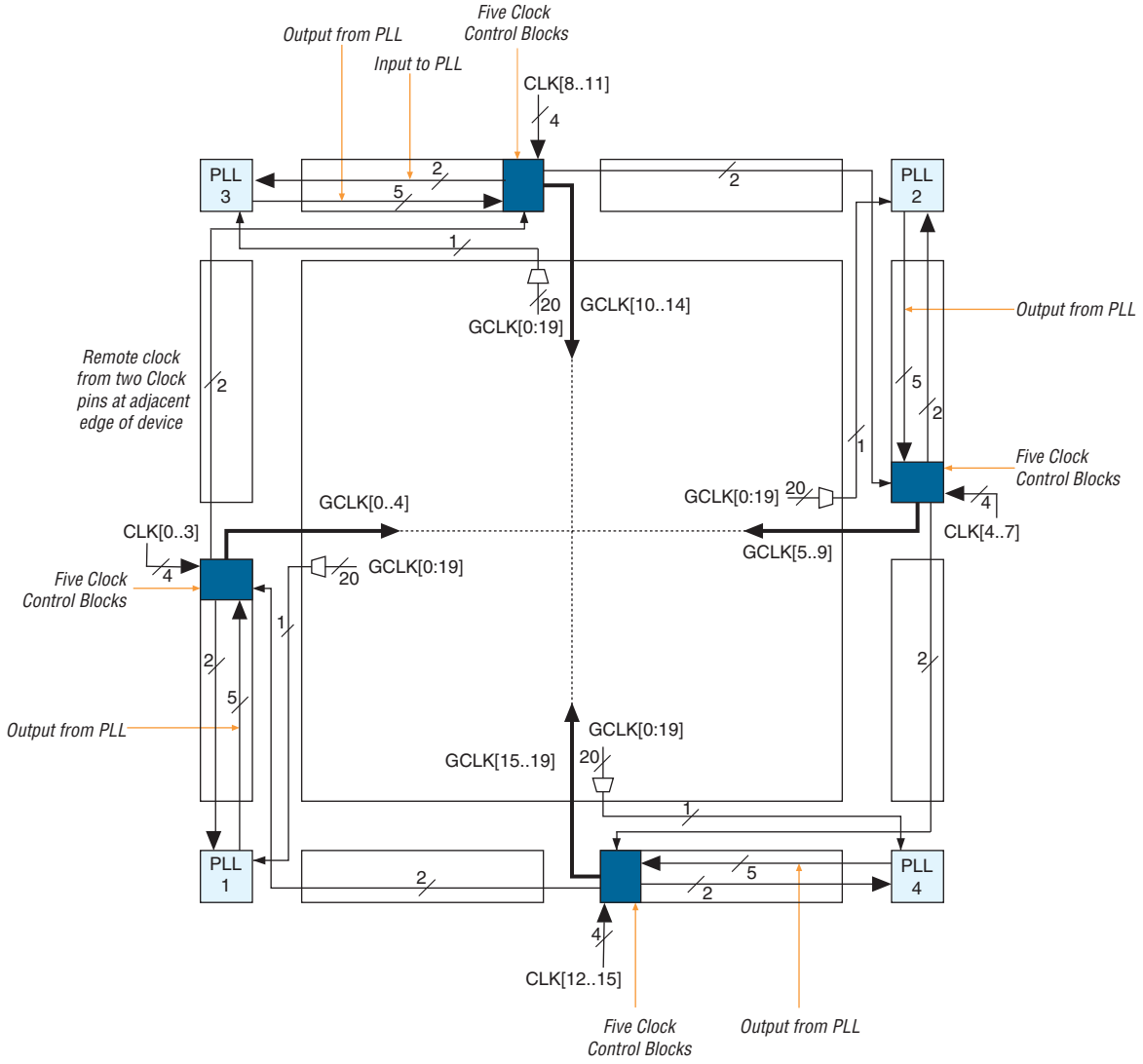
The Cyclone III device family supports dynamic phase shifting of VCO phase taps only. The phase shift is configurable for any number of times. Each phase shift takes about one scanclk cycle, allowing you to implement large phase shifts quickly.

PLL Cascading

Two PLLs are cascaded to each other through the clock network. If your design cascades PLLs, the source (upstream) PLL must have a low-bandwidth setting, while the destination (downstream) PLL must have a high-bandwidth setting.

Figure 5–19 shows using GCLK while cascading PLLs.

Figure 5–19. PLL Cascading Using GCLK



Consider the following guidelines when cascading PLLs:

- Set the primary PLL to low bandwidth to help filter jitter. Set the secondary PLL to high bandwidth to be able to track the jitter from the primary PLL. You can view the Quartus II software compilation report file to ensure the PLL bandwidth ranges do not overlap. If the bandwidth ranges overlap, jitter peaking can occur in the cascaded PLL scheme.

  You can get an estimate of the PLL deterministic jitter and static phase error (SPE) by using the TimeQuest Timing Analyzer in the Quartus II software. Use the SDC command "derive_clock_uncertainty" to direct TimeQuest to generate a report titled "PLLJ_PLLSPE_INFO.txt" in your project directory. Then, use "set_clock_uncertainty" commands to add jitter and SPE values to your clock constraints.

- Keep the secondary PLL in a reset state until the primary PLL has locked to ensure the phase settings are correct on the secondary PLL.
- You cannot connect either of the `inclk` ports of any PLLs in the cascaded scheme to clock outputs from PLLs in the cascaded scheme.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Cyclone III device family PLLs, you can reconfigure both counter settings and phase shift the PLL output clock in real time. You can also change the charge pump and loop filter components, which dynamically affects PLL bandwidth. You can use these PLL components to update the output clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

The ability to reconfigure the PLL in real time is useful in applications that might operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output clock phase dynamically. For instance, a system generating test patterns is required to generate and send patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring PLL components in real time allows you to switch between two such output frequencies in a few microseconds.

You can also use this feature to adjust clock-to-out (t_{CO}) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

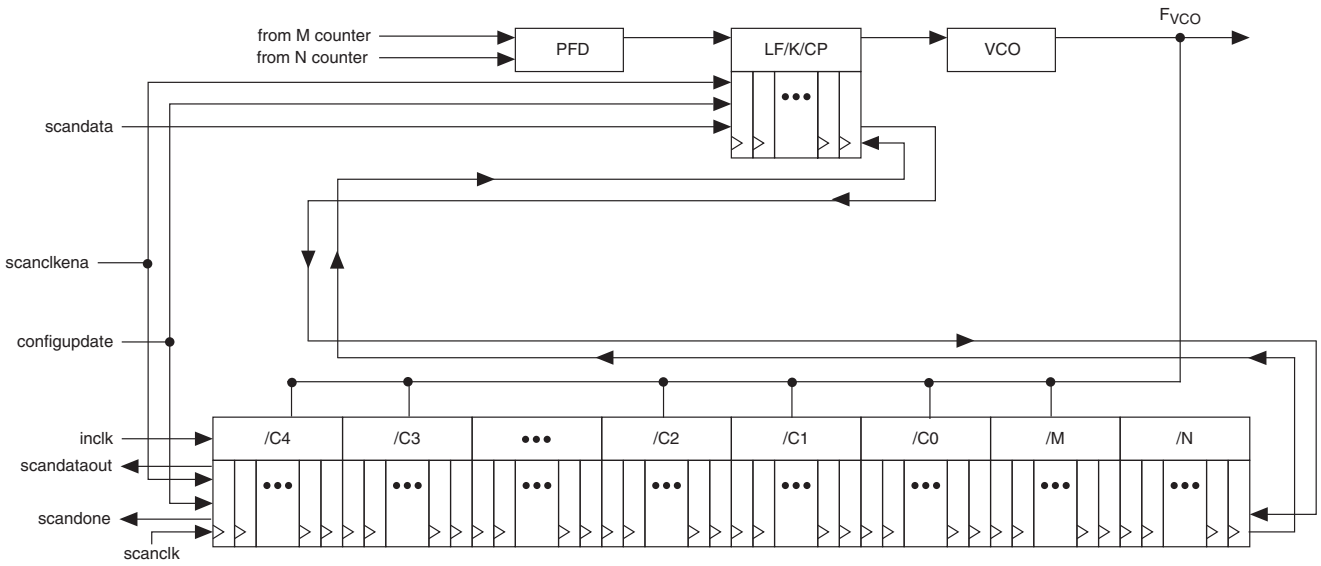
PLL Reconfiguration Hardware Implementation

The following PLL components are configurable in real time:

- Pre-scale counter (N)
- Feedback counter (M)
- Post-scale output counters (`C0-C4`)
- Dynamically adjust the charge pump current (I_{CP}) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth

Figure 5–20 shows how to adjust PLL counter settings dynamically by shifting their new settings into a serial shift register chain or scan chain. Serial data shifts to the scan chain via the `scandataport`, and shift registers are clocked by `scanclk`. The maximum `scanclk` frequency is 100 MHz. After shifting the last bit of data, asserting the `configupdate` signal for at least one `scanclk` clock cycle synchronously updates the PLL configuration bits with the data in the scan registers.

Figure 5–20. PLL Reconfiguration Scan Chain



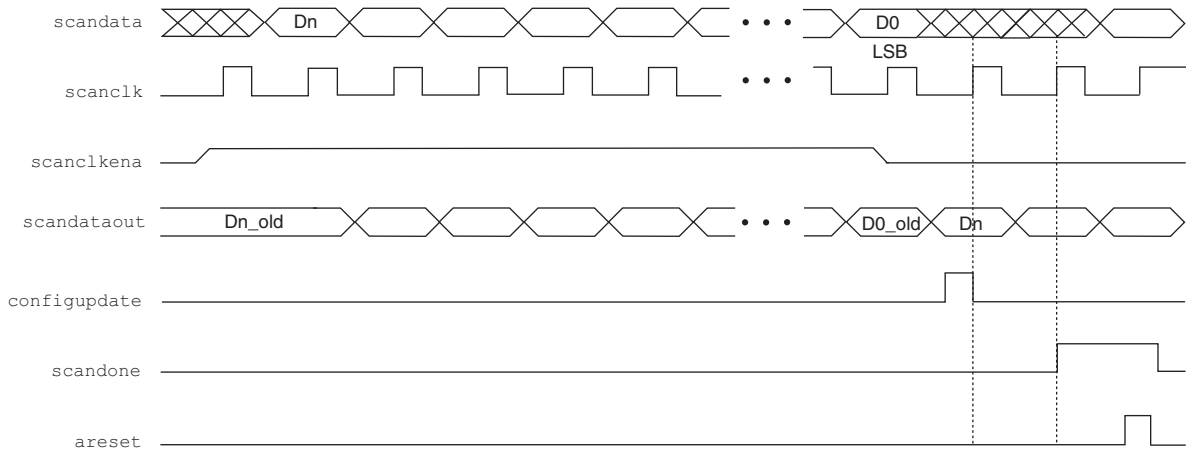
The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, not all counters update simultaneously.

To reconfigure the PLL counters, perform the following steps:

1. The `scanclkena` signal is asserted at least one `scanclk` cycle prior to shifting in the first bit of `scandata` (`Dn`).
2. Serial data (`scandata`) is shifted into the scan chain on the second rising edge of `scanclk`.
3. After all 144 bits have been scanned into the scan chain, the `scanclkena` signal is deasserted to prevent inadvertent shifting of bits in the scan chain.
4. The `configupdate` signal is asserted for one `scanclk` cycle to update the PLL counters with the contents of the scan chain.
5. The `scandone` signal goes high indicating that the PLL is being reconfigured. A falling edge indicates that the PLL counters have been updated with new settings.
6. Reset the PLL using the `areset` signal if you make any changes to the M, N, post-scale output C counters, or the I_{CP}, R, C settings.
7. You can repeat steps 1 through 5 to reconfigure the PLL any number of times.

Figure 5–21 shows a functional simulation of the PLL reconfiguration feature.

Figure 5–21. PLL Reconfiguration Scan Chain



When reconfiguring the counter clock frequency, the corresponding counter phase shift settings cannot be reconfigured using the same interface. You can reconfigure phase shifts in real time using the dynamic phase shift reconfiguration interface. If you reconfigure the counter frequency, but wish to keep the same non-zero phase shift setting (for example, 90°) on the clock output, you must reconfigure the phase shift after reconfiguring the counter clock frequency.

Post-Scale Counters (C0 to C4)

You can configure multiply or divide values and duty cycle of post-scale counters in real time. Each counter has an 8-bit high time setting and an 8-bit low time setting. The duty cycle is the ratio of output high or low time to the total cycle time, which is the sum of the two. Additionally, these counters have two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

When the `rbypass` bit is set to 1, it bypasses the counter, resulting in a divide by one. When this bit is set to 0, the PLL computes the effective division of the VCO output frequency based on the high and low time counters. For example, if the post-scale divide factor is 10, the high and low count values is set to 5 and 5 respectively, to achieve a 50–50% duty cycle. The PLL implements this duty cycle by transitioning the output clock from high-to-low on the rising edge of the VCO output clock. However, a 4 and 6 setting for the high and low count values, respectively, would produce an output clock with 40–60% duty cycle.

The `rselodd` bit indicates an odd divide factor for the VCO output frequency with a 50% duty cycle. For example, if the post-scale divide factor is three, the high and low time count values are 2 and 1, respectively, to achieve this division. This implies a 67%–33% duty cycle. If you need a 50%–50% duty cycle, you must set the `rselodd` control bit to 1 to achieve this duty cycle despite an odd division factor. The PLL implements this duty cycle by transitioning the output clock from high-to-low on a falling edge of the VCO output clock. When you set `rselodd` = 1, subtract 0.5 cycles from the high time and add 0.5 cycles to the low time.

For example:

- High time count = 2 cycles
- Low time count = 1 cycle
- `rselodd` = 1 effectively equals:
  - High time count = 1.5 cycles
  - Low time count = 1.5 cycles
  - Duty cycle = (1.5/3)% high time count and (1.5/3)% low time count

Scan Chain Description

Cyclone III device family PLLs have a 144-bit scan chain.

Table 5–4 lists the number of bits for each component of the PLL.

Table 5–4. Cyclone III Device Family PLL Reprogramming Bits

Block Name	Number of Bits		
	Counter	Other	Total
C4 (1)	16	2 (2)	18
C3	16	2 (2)	18
C2	16	2 (2)	18
C1	16	2 (2)	18
C0	16	2 (2)	18
M	16	2 (2)	18
N	16	2 (2)	18
Charge Pump	9	0	9
Loop Filter (3)	9	0	9
Total number of bits:			144

Notes to Table 5–4:

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

(2) These two control bits include `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

(3) MSB bit for loop filter is the last bit shifted into the scan chain.

Figure 5–22 shows the scan chain order of the PLL components.

Figure 5–22. PLL Component Scan Chain Order

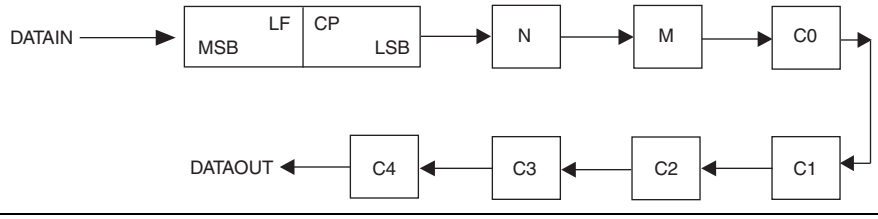
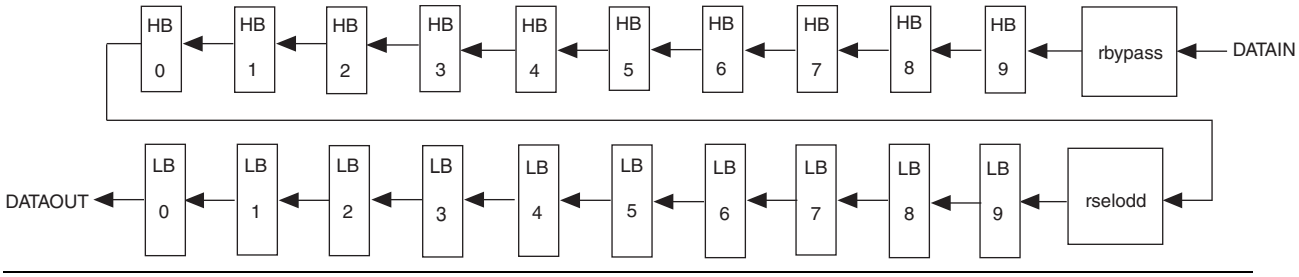


Figure 5–23 shows the scan chain bit order sequence for one PLL post-scale counter in Cyclone III device family PLLs.

Figure 5–23. Scan Chain Bit Order



For more information about the PLL scan chain, refer to the *Implementing PLL Reconfiguration in Cyclone III Devices* application note.

Charge Pump and Loop Filter

You can reconfigure the charge pump and loop filter settings to update the PLL bandwidth in real time. Table 5–5 through Table 5–7 list the possible settings for charge pump (ICP), loop filter resistor (R), and capacitor (C) values for Cyclone III device family PLLs.

Table 5–5. Charge Pump Bit Control

CP[2]	CP[1]	CP[0]	Setting (Decimal)
0	0	0	0
0	0	1	1
0	1	1	3
1	1	1	7

Table 5–6. Loop Filter Resistor Value Control

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	Setting (Decimal)
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

Table 5–7. Loop Filter Control of High Frequency Capacitor

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

Bypassing PLL Counter

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

Table 5–8 lists the settings for bypassing the counters in Cyclone III device family PLLs.

Table 5–8. 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 5–8:

(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 5–9 lists the control signals that are used for dynamic phase shifting.

Table 5–9. 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

Table 5–9. Dynamic Phase Shifting Control Signals (Part 2 of 2)

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

Table 5–10 lists the PLL counter selection based on the corresponding PHASECOUNTERSELECT setting.

Table 5–10. Phase Counter Select Mapping

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

To perform one dynamic phase shift step, you must perform the following procedures:

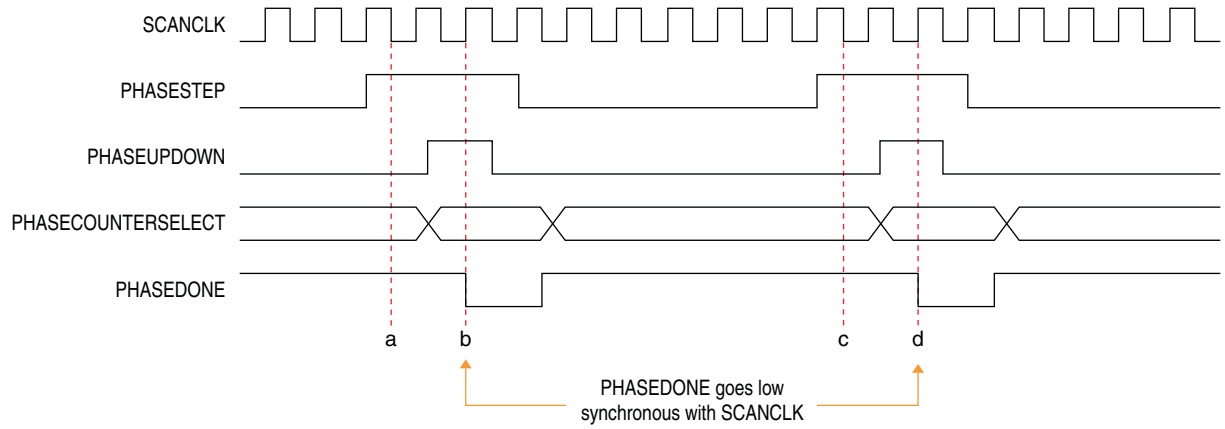
1. Set PHASEUPDOWN and PHASECOUNTERSELECT as required.
2. Assert PHASESTEP for at least two SCANCLK cycles. Each PHASESTEP pulse allows one phase shift.
3. Deassert PHASESTEP after PHASEDONE goes low.
4. Wait for PHASEDONE to go high.
5. Repeat steps 1 through 4 as many times as required to perform multiple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

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

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

Figure 5–24 shows the dynamic phase shifting waveform.

Figure 5–24. Timing Diagram for Dynamic Phase Shift



The PHASESTEP signal is latched on the negative edge of SCANCLK (a,c) and must remain asserted for at least two SCANCLK cycles. Deassert PHASESTEP after PHASEDONE goes low. On the second SCANCLK rising edge (b,d) after PHASESTEP is latched, the values of PHASEUPDOWN and PHASECOUNTERSELECT are latched and the PLL starts dynamic phase-shifting for the specified counters, and in the indicated direction. PHASEDONE is deasserted synchronous to SCANCLK at the second rising edge (b,d) and remains low until the PLL finishes dynamic phase-shifting. Depending on the VCO and SCANCLK frequencies, PHASEDONE low time may be greater than or less than one SCANCLK cycle.

You can perform another dynamic phase-shift after the PHASEDONE signal goes from low to high. Each PHASESTEP pulse enables one phase shift. PHASESTEP pulses must be at least one SCANCLK cycle apart.

For information about the ALTPLL_RECONFIG MegaWizard Plug-In Manager, refer to the *Phase-Locked Loop Reconfiguration (ALTPLL_RECONFIG) Megafunction* user guide.

Spread-Spectrum Clocking

The Cyclone III device family can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of the PLL. Cyclone III device family PLLs can track a spread-spectrum input clock as long as it is in the input jitter tolerance specifications and the modulation frequency of the input clock is below the PLL bandwidth, which is specified in the fitter report. The Cyclone III device family cannot generate spread-spectrum signals internally.

PLL Specifications

For information about PLL specifications, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Document Revision History

Table 5–11 lists the revision history for this document.

Table 5–11. Document Revision History (Part 1 of 2)

Date	Version	Changes
July 2012	4.1	Updated Figure 5–2.
November 2011	4.0	■ Minor edits to Equation 5–1 and Equation 5–2. ■ Updated Table 5–5. ■ Updated Figure 5–6, Figure 5–13, Figure 5–19, and Figure 5–24. ■ Updated "Clock Control Block" on page 5–4, "Manual Override" on page 5–20, "PLL Cascading" on page 5–24, and "Dynamic Phase Shifting" on page 5–31. ■ Minor text edits.
December 2009	3.2	Minor changes to the text.
July 2009	3.1	Made minor correction to the part number.
June 2009	3.0	■ Updated to include Cyclone III LS information. ■ Updated chapter part number. ■ Updated "Clock Networks" on page 5–1. ■ Updated Table 5–1 on page 5–2, Table 5–3 on page 5–9. ■ Updated "PLLs in the Cyclone III Device Family" on page 5–9. ■ Updated "PLL Reconfiguration Hardware Implementation" on page 5–25. ■ Updated "Spread-Spectrum Clocking" on page 5–32.
October 2008	2.1	■ Updated the "Dynamic Phase Shifting" and "Introduction" sections. ■ Updated Figure 5–2, Figure 5–8, and Figure 5–24. ■ Updated chapter to new template.
May 2008	2.0	■ Updated Figure 5–2 and added (Note 3). ■ Updated "clkena Signals" section. ■ Updated Figure 5–8 and added (Note 3). ■ Updated "PLL Control Signals" section. ■ Updated "PLL Cascading" section. ■ Updated "Cyclone III PLL Hardware Overview" section. ■ Updated Table 5–6, Table 5–3, Table 5–7. ■ Updated Figure 5–14. ■ Updated "PLL Cascading" section. ■ Updated "Clock Multiplication and Division" section. ■ Updated Step 6–32 in "PLL Reconfiguration Hardware Implementation" section. ■ Updated "Spread-Spectrum Clocking" section. ■ Updated Figure 5–29. ■ Updated "VCCD and GND" section. ■ Added "Power Consumption" section.
September 2007	1.2	■ Updated "Board Layout" section and removed Figure 5-30.

Table 5–11. Document Revision History (Part 2 of 2)

Date	Version	Changes
July 2007	1.1	■ Updated document with EP3C120 information. ■ Updated Table 5–1 and Table 5–4 with EP3C120 information. ■ Updated "Clock Control Block" section. ■ Updated locked signal information in "PLL Control Signals" section and added Figure 5–16. ■ Updated "Manual Override" section, updated "Manual Clock Switchover" section. ■ Added new "Programmable Bandwidth" section with Figure 5–21 and Figure 5–22. ■ Replaced Figure 5-30 with correct diagram. ■ Added chapter TOC and "Referenced Documents" section.
March 2007	1.0	Initial release.