6. Using PLLs in Cyclone Devices

C51006-1.5

Introduction

Cyclone® FPGAs offer phase locked loops (PLLs) and a global clock network for clock management solutions. Cyclone PLLs offer clock multiplication and division, phase shifting, programmable duty cycle, and external clock outputs, allowing system-level clock management and skew control. The Altera® Quartus® II software enables Cyclone PLLs and their features without using any external devices. This chapter explains how to design and enable Cyclone PLL features.

PLLs are commonly used to synchronize internal device clocks with an external clock, run internal clocks at higher frequencies than an external clock, minimize clock delay and clock skew, and reduce or adjust clock-to-out (t_{CO}) and set-up (t_{SU}) times.

Hardware Overview

Cyclone FPGAs contain up to two PLLs per device. Table 6–1 shows which PLLs are available for each Cyclone FPGA.

Table 6–1. Cyclone FPGA PLL Availability

Device	PLL1 *(1)*	PLL2 *(2)*
EP1C3	✓	—
EP1C4	✓	✓
EP1C6	✓	✓
EP1C12	✓	✓
EP1C20	✓	✓

Notes to Table 6–1:
(1) Located on the center left side of the device.
(2) Located on the center right side of the device.

Table 6–2 provides an overview of available Cyclone PLL features.

Table 6–2. Cyclone PLL Features

Feature	Description
Clock multiplication and division	*M*/(*N* × post-scale counter) *(1)*
Phase shift	Down to 125-ps increments *(2)*, *(3)*
Programmable duty cycle	✓
Number of internal clock outputs	Two per PLL
Number of external clock outputs *(4)*	One per PLL
Locked port can feed logic array	✓
PLL clock outputs can feed logic array	✓

Notes to Table 6–2:

(1) *M*, N, and post-scale counter values range from 1 to 32.

(2) The smallest phase shift is determined by the Voltage Control Oscillator (VCO) period divided by 8.

(3) For degree increments, Cyclone FPGAs can shift output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the multiplication/division ratio needed on the PLL clock output.

(4) The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not have support for a PLL LVDS input or an external clock output. The EP1C6 PLL2 in the 144-pin TQFP package does not support an external clock output.

Cyclone PLL Blocks

The main goal of a PLL is to synchronize the phase and frequency of an internal/external clock to an input reference clock. There are a number of components that comprise a PLL to achieve this phase alignment.

Cyclone PLLs align the rising edge of the reference input clock to a feedback clock using a phase-frequency detector (PFD). The falling edges are determined by the duty cycle specifications. The PFD produces an up or down signal that determines whether the VCO needs to operate at a higher or lower frequency. The PFD output is applied to the charge pump and loop filter, which produces a control voltage for setting the frequency of the VCO. If the PFD produces an up signal, then the VCO frequency increases, while a down signal causes the VCO frequency to decrease.

The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if it receives a down signal, current is drawn from the loop filter. The loop filter converts these up and down signals to a voltage that

is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage over-shoot, which minimizes the jitter on the VCO.

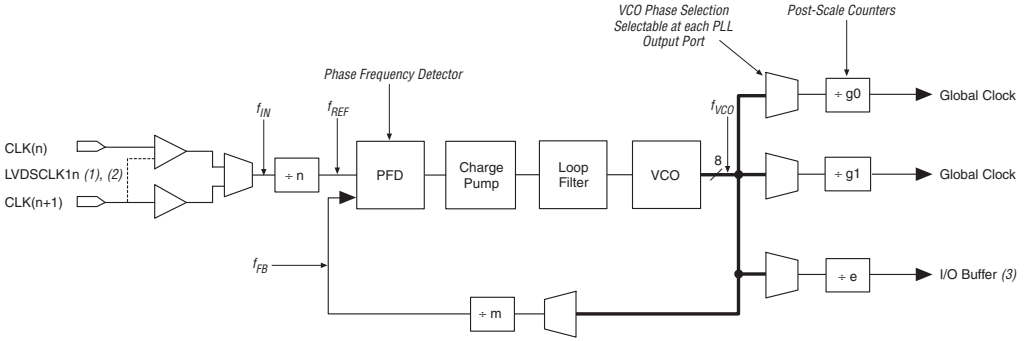
The voltage from the loop filter determines how fast the VCO operates. The VCO is implemented as a four-stage differential ring oscillator. A divide counter (*M*) is inserted in the feedback loop to increase the VCO frequency above the input reference frequency, making the VCO frequency (f_{VCO}) equal to *M* times the input reference clock (f_{REF}). The input reference clock (f_{REF}) to the PFD is equal to the input clock (f_{IN}) divided by the pre-scale counter (*N*). Therefore, the feedback clock (f_{FB}) that is applied to one input of the PFD is locked to the f_{REF} that is applied to the other input of the PFD.

The VCO output can feed up to three post-scale counters (G0, G1, and E). These post-scale counters allow a number of harmonically-related frequencies to be produced within the PLL.

Additionally, the PLL has internal delay elements to compensate for routing on the global clock networks and I/O buffers of the external clock output pins. These internal delays are fixed and not accessible to the user.

Figure 6–1 shows a block diagram of the major components of a Cyclone PLL.

Figure 6–1. Cyclone PLL



Notes to Figure 6–1:

(1) The EP1C3 device in the 100-pin TQFP package does not have support for a PLL LVDS input.

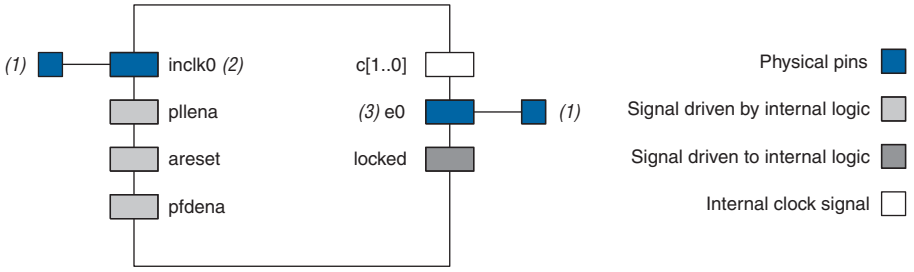
(2) If you are using the LVDS standard, then both `CLK` pins of that PLL are used. LVDS input is supported via the secondary function of the dedicated `CLK` pins. For PLL1, the `CLK0` pin's secondary function is `LVDSCLK1p` and the `CLK1` pin's secondary function is `LVDSCLK1n`. For PLL2, the `CLK2` pin's secondary function is `LVDSCLK2p` and the `CLK3` pin's secondary function is `LVDSCLK2n`.

(3) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.

Software Overview

Cyclone PLLs are enabled in the Quartus II software by using the `altpll` megafunction. Figure 6–2 shows the available ports (as they are named in the Quartus II `altpll` megafunction) of Cyclone PLLs and their sources and destinations. It is important to note that the `c[1..0]` and `e0` clock output ports from `altpll` are driven by the post-scale counters G0, G1, and E (not necessarily in that order). The G0 and G1 counters feed the internal global clock network on the `c0` and `c1` PLL outputs, and the E counter feeds the PLL external clock output pin on the `e0` PLL output.

Figure 6–2. Cyclone PLL Signals



Notes to Figure 6–2:

(1) You can assign these signals to either a single-ended I/O standard or LVDS.
(2) `Inclk0` must be driven by the dedicated clock input pin(s).
(3) `e0` drives the dual-purpose `PLL[2..1]_OUT` pins.

Tables 6–3 and 6–4 describe the Cyclone PLL input and output ports.

Table 6–3. PLL Input Signals

Port	Description	Source	Destination
inclk0	Clock input to PLL.	Dedicated clock input pin *(1)*	÷n counter
pllena *(2)*	pllena is an active-high signal that acts as a combined enable and reset signal for the PLL. You can use It for enabling or disabling one or two PLLs. When this signal is driven low, the PLL clock output ports are driven to GND and the PLL loses lock. Once this signal is driven high again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the pllena port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PLL control signal
areset	areset is an active-high signal that resets all PLL counters to their initial values. When this signal is driven high, the PLL resets its counters, clears the PLL outputs, and loses lock. Once this signal is driven low again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the areset port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PLL control signal
pfdena	pfdena is an active-high signal that enables or disables the up/down output signals from the PFD. When pfdena is driven low, the PFD is disabled, while the VCO continues to operate. The PLL clock outputs continue to toggle regardless of the input clock, but can experience some long-term drift. Because the output clock frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. You can drive the pfdena port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PFD

Notes to Table 6–3:

(1) The inclk0 port to the PLL must be driven by the dedicated clock input pin(s).

(2) There is no dedicated pllena pin for all PLLs, allowing you to choose either one pllena pin for both PLLs or each PLL can have its own pllena pin.

(3) Logic array source means that you can drive the port from internal logic or any general-purpose I/O pin.

Table 6–4. PLL Output Signals

Port	Description	Source	Destination
c[1..0]	PLL clock outputs driving the internal global clock network.	PLL post-scale counter G0 or G1	Global clock network *(1)*
e0 *(2)*	PLL clock output driving the single-ended or LVDS external clock output pin(s).	PLL post-scale counter E	PLL[2..1]_OUT pin(s) *(3)*
locked	Gives the status of the PLL lock. When the PLL is locked, this port drives logic high. When the PLL is out of lock, this port drives logic low. The locked port can pulse high and low during the PLL lock process.	PLL lock detect	Logic array *(4)*

Notes to Table 6–4:

(1) C[1..0] can also drive to any general-purpose I/O pin through the global clock network.

(2) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not have support for the external clock output PLL[2..1]_OUT.

(3) The PLL[2..1]_OUT pins are dual-purpose pins. If these pins are not required, they are available for use as general-purpose I/O pins.

(4) Logic array destination means that you can drive the port to internal logic or any general-purpose I/O pin.

In the Quartus II software, you define which internal clock output from the PLL (c0 or c1) should be compensated. This PLL clock output is phase-aligned with respect to the PLL input clock. For example, if c0 is specified as the compensation clock in normal mode, the compensation is based on the c0 routing on the global clock network.

Pins and Clock Network Connections

You must drive Cyclone PLLs by the dedicated clock input pins CLK[3..0]. Inverted clocks and internally generated clocks cannot drive the PLL. Table 6–5 shows which dedicated clock pin drives which PLL input clock port.

A single clock input pin cannot drive both PLLs, but a single clock input pin can feed both registers in the logic array, as well as the PLL `inclk` port.

Table 6–5. PLL Input Clock Sources

Clock Input Pins *(1)*	PLL1	PLL2 *(2)*
CLK0	✓	—
CLK1	✓	—
CLK2	—	✓
CLK3	—	✓

Notes to Table 6–5:
(1) If you are using the LVDS standard, then both `CLK` pins driving that PLL are used.
(2) The EP1C3 device only supports PLL1.

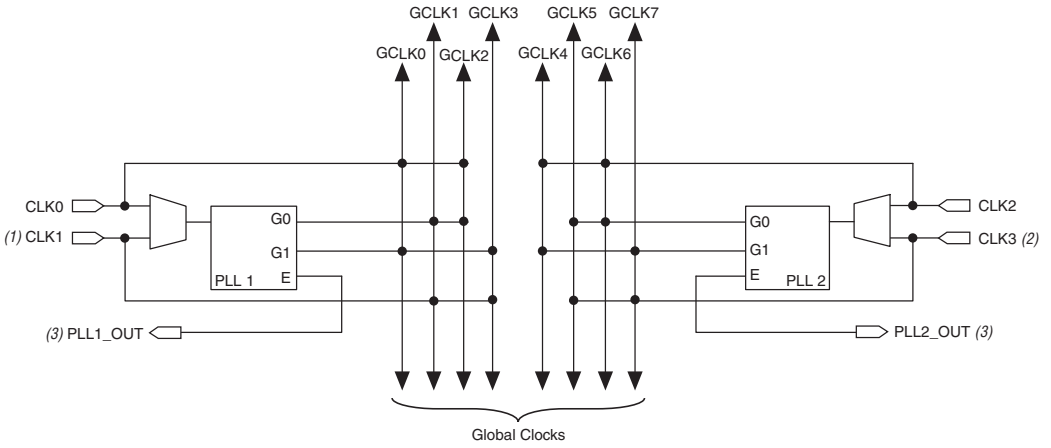
The `c[1..0]` and `e0` clock output ports from `altpll` are driven by the PLL post-scale counters G0, G1, and E (not necessarily in that order). The G0 and G1 counters feed the internal global clock network on the `c0` and `c1` PLL outputs, and the E counter feeds the PLL external clock output pin on the `e0` PLL output. Table 6–6 shows which global clock network can be driven by which PLL post-scale counter output.

Table 6–6. PLL Output Clock Destinations onto the Global Clock Network

PLL	Counter Output	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
PLL 1	G0	—	✓	✓	—	—	—	—	—
	G1	✓	—	—	✓	—	—	—	—
PLL2	G0	—	—	—	—	—	✓	✓	—
	G1	—	—	—	—	✓	—	—	✓

Figure 6–3 summarizes Tables 6–5 and 6–6 by showing the PLL input and output clock connections.

Figure 6–3. Cyclone PLL Clock Connections



Notes to Figure 6–3:

(1) PLL1 supports one single-ended or LVDS input via the `CLK0` and `CLK1` pins.

(2) PLL2 supports one single-ended or LVDS input via the `CLK2` and `CLK3` pins.

(3) `PLL1_OUT` and `PLL2_OUT` support single-ended or LVDS outputs. If the external clock output is not required, these pins are available as general-purpose I/O pins.

You can invert the clock outputs of the PLL at the logic array block (LAB) and at the input/output element (IOE) level.

Hardware Features

Cyclone PLLs have a number of advanced features available, including clock multiplication and division, phase shifting, programmable duty cycles, external clock outputs, and control signals.

Clock Multiplication and Division

Cyclone PLLs provide clock synthesis for PLL output ports using $M/(N \times \text{post-scale})$ scaling factors. There is one pre-scale divider (N) and one multiply counter (M) per PLL. N and post-scale counter values range from 1 to 32. The M counter ranges from 2 to 32. The input clock (f_{IN}) is divided by a pre-scale counter (N) to produce the input reference clock (f_{REF}) to the PFD. f_{REF} is then multiplied by the M feedback factor. The control loop drives the VCO frequency to match $f_{IN} \times (M/N)$. See the following equations:

$$f_{REF} = f_{IN}/N$$
$$f_{VCO} = f_{REF} \times M = f_{IN} \times (M/N)$$

Each output port has a unique post-scale counter to divide down the high-frequency VCO. There are three post-scale counters (G0, G1, and E) that range from 1 to 32. See the following equations:

$$f_{C0} = f_{VCO}/G0 = f_{IN} \times (M/(N \times G0))$$
$$f_{C1} = f_{VCO}/G1 = f_{IN} \times (M/(N \times G1))$$
$$f_{E} = f_{VCO}/E = f_{IN} \times (M/(N \times E))$$

`c0` and `c1` can use either post-scale counter, G0 or G1.

For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets the VCO frequency specifications. Then, the post-scale counters scale down the output frequency for each PLL clock output port. For example, if clock output frequencies required from one PLL are 33 and 66 MHz, the VCO is set to 330 MHz (the least common multiple in the VCO's range).

Phase Shifting

Cyclone PLLs have advanced clock shift capability to provide programmable phase shifting. You can enter the desired phase shift in the `altpll` MegaWizard® Plug-In Manager and the Quartus II software automatically sets and displays the closest phase shift achievable. You can enter the phase shift in degrees, or units of time, for each PLL clock output port. This feature is supported on all three PLL post-scale counters, G0, G1, and E and is supported for all available clock feedback modes.

Phase shifting is performed with respect to the PLL clock output that is compensated. For example, you have a 100 MHz input clock and request a × 1 multiplication with a +90° phase shift on `c0` and a × 1 multiplication with a +45° phase shift on `c1`. If you choose to compensate for the `c0` clock output, the PLL uses a zero phase-shifted `c0` clock as a reference point to produce the +90° phase shift on `c0`. Since `c0` is the compensated clock, it is phase-shifted +90° from the input clock. The `c1` clock also uses the zero phase-shifted `c0` reference to produce the +45° phase shift on `c1`.

For fine phase adjustment, each PLL clock output counter can choose a different phase of the VCO from up to eight phase taps. In addition, each clock output counter can use a unique initial count setting to achieve individual coarse phase shift selection, in steps of one VCO period. The Quartus II software can use this clock output counter, along with an initial setting on the post-scale counter, to achieve a phase shift range for the entire period of the output clock. You can phase shift the PLL clock output up to ±180°. The Quartus II software automatically sets the phase taps and counter settings according to the phase shift requested.

The resolution of the fine phase adjustment is dependent on the input frequency and the multiplication/division factors (i.e., it is a function of the VCO period), with the finest step being equal to an eighth (× 0.125) of the VCO period. The minimum phase shift is $1/(8 \times f_{VCO})$ or $N/(8 \times M \times f_{IN})$. In Cyclone FPGAs, the VCO ranges from 500 to 1,000 MHz. Therefore, phase shifting can be performed with a resolution range of 1/(8 × 1,000 MHz) to 1/(8 × 500 MHz), which is 125 to 250 ps in time units.

Because there are eight VCO phase taps, the maximum step size is 45°. Smaller steps are possible, depending on the multiplication and division ratio necessary on the output clock port. The equation to determine the precision of the phase shifting in degrees is 45° divided by the post-scale counter value. For example, if you have an input clock of 125 MHz with × 1, the post-scale counter G0 is 3. Therefore, the smallest phase shift step is (45°/3 = 15°) and possible phase-shift values would be multiples of 15°.

This type of phase shift provides the highest precision since it is the least sensitive to process, voltage and temperature variation.

Programmable Duty Cycle

The programmable duty cycle feature allows you to set the duty cycle of the PLL clock outputs. The duty cycle is the ratio of the clock output high/low time to the total clock cycle time, which is expressed as a percentage of high time. This feature is supported on all three PLL post-scale counters (G0, G1, and E).

The duty cycle is set by using a low- and high-time count setting for the post-scale counters. The Quartus II software uses the input frequency and target multiply/divide ratio to select the post-scale counter. The precision of the duty cycle is determined by the post-scale counter value chosen on a PLL clock output and is defined as 50% divided by the post-scale counter value. For example, if the post-scale counter value is 3, the allowed duty cycle precision would be 50% divided by 3 equaling 16.67%. Because the `altpll` megafunction does not accept non-integer values for the duty cycle values, the allowed duty cycles are 17, 33, 50, and 67%.

Due to hard limitations, you cannot achieve a duty cycle of 84% because you cannot achieve the closest value to 100% for a given counter value. However, you can achieve a duty cycle of 84% by choosing a 17% duty cycle and inverting the PLL clock output. For example, if the G0 counter is 10, increments of 5% are possible for duty cycle choices between 5 and 90%.

External Clock Output

Each PLL supports one single-ended or LVDS external clock output for general-purpose external clocks, or for source-synchronous transmitters. The output of the E counter drives the PLL external clock output (e0), which can only feed to the PLL[2..1]_OUT pins and not to internal logic. You can use PLL[2..1]_OUT in all three clock feedback modes.

The EP1C3 device in the 100-pin package, and the EP1C6 PLL2 in the 144-pin package, do not have support for an external clock output.

The PLL[2..1]_OUT pins are dual-purpose pins, meaning if the pins are not required by the PLL, they are available for use as general-purpose I/O pins. The I/O standards supported by the PLL[2..1]_OUT pins are listed in Table 6–7.

Table 6–7. Supported I/O Standards for Cyclone PLL Pins

I/O Standard	Inclk	PLL[2..1]_OUT *(1)*
LVTTL	✓	✓
LVCMOS	✓	✓
2.5-V	✓	✓
1.8-V	✓	✓
1.5-V	✓	✓
3.3-V PCI	✓	✓
LVDS *(2)*	✓	✓
SSTL-2 Class I	✓	✓
SSTL-2 Class II	✓	✓
SSTL-3 Class I	✓	✓
SSTL-3 Class II	✓	✓
Differential SSTL-2 Class II	—	✓

Notes to Table 6–7:

(1) The EP1C3 device in the 100-pin TQFP package and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.

(2) The EP1C3 device in the 100-pin TQFP package does not support an LVDS input.

Since the pllena and locked signal can be driven by or driven to general-purpose I/O pins, respectively, they support all Cyclone I/O standards.

The Cyclone external clock output pins (PLL[2..1]_OUT) do not have a separate V_{CC} and GND bank internal to the device. The PLL[2..1]_OUT pins share a V_{CCIO} bank with neighboring I/O pins. Only the I/O pins in the same bank have an effect on the PLL[2..1]_OUT pins. Therefore, to minimize jitter on the PLL[2..1]_OUT pins, I/O pins directly adjacent to these pins should be either inputs or they should not be used. For more information about board design guidelines, see "Jitter Considerations" on page 6–19.

Control Signals

There are four available control signals, pllena, areset, pfdena, and locked, in Cyclone PLLs that provide added PLL management.

pllena

The PLL enable signal, pllena, enables or disables the PLL. You can either enable/disable a single PLL (by connecting pllena port independently) or multiple PLLs (by connecting pllena ports together). The pllena signal is an active-high signal. When pllena is low, the PLL clock output ports are driven to logic low and the PLL loses lock. All PLL counters, including gated lock counter return to default state. When pllena goes high again, the PLL relocks and resynchronizes to the input clock. Therefore, pllena is an active-high signal. In Cyclone FPGAs, you can feed the pllena port from internal logic or any general-purpose I/O pin because there is no dedicated pllena pin. This feature offers added flexibility, since each PLL can have its own pllena control circuitry, or both PLLs can share the same pllena circuitry. The pllena signal is optional, and when it is not enabled in the software, the port is internally tied to VCC.

areset

The PLL areset signal is the reset or resynchronization input for each PLL. The areset signal should be asserted every time the PLL loses lock to guarantee correct phase relationship between the PLL input and output clocks. Users should include the areset signal in designs where phase relationship between input and output clocks need to be maintained after a loss of lock condition. The areset signal is an active high signal and, when driven high, the PLL counters reset, clearing the PLL output and causing the PLL to lose lock. The clock outputs of the PLL are driven to ground as long as areset is active. When areset transitions low, the PLL will resynchronize to its input clock as the PLL relocks. If the target VCO frequency is below this nominal frequency, the PLL clock output frequency will start at a higher value than desired during the lock process. In this case, Altera recommends monitoring the gated locked signal to ensure the PLL is fully in lock before enabling the clock outputs

from the PLL. Cyclone FPGAs can drive this PLL input signal from LEs or any general-purpose I/O pin. The areset signal is optional. When it is not enabled in the Quartus II software, the port is internally tied to GND.

pfdena

The pfdena signal controls the PFD output in the PLL with a programmable gate. If you disable the PFD by driving areset low, the VCO operates at its last set control voltage and frequency value with some long-term drift to a lower frequency. The VCO frequency can drift up to +/- 5% over 25 us. Even though the PLL clock outputs continue to toggle regardless of the input clock, the PLL could lose lock. The system continues running when the PLL goes out of lock, or if the input clock is disabled. Because the last locked output frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. By maintaining this frequency, the system has time to store its current settings before shutting down. If the pfdena signal goes high again, the PLL relocks and resynchronizes to the input clock. Therefore, the pfdena pin is an active-high signal. You can drive the pfdena input signal by any general-purpose I/O pin, or from internal logic. This signal is optional, and when it is not enabled in the software, the port is internally tied to VCC.

locked

When the locked output is at a logic-high level, this level indicates a stable PLL clock output in phase with the PLL reference input clock. Without any additional circuitry, the locked port may toggle as the PLL begins tracking the reference clock. The locked port of the PLL can feed any general-purpose I/O pin and/or internal logic. This locked signal is optional, but is useful in monitoring the PLL lock process.

Whenever the PLL loses lock for any reason (be it excessive inclk jitter, power supply noise, etc.), the PLL must be reset with the areset signal to guarantee correct phase relationship between the PLL output clocks. If the phase relationship between the input clock versus output clock, and between different output clocks from the PLL is not important in your design, the PLL need not be reset.

Clock Feedback Modes

Cyclone PLLs support three feedback modes: normal, zero delay buffer, and no compensation. Unlike other Altera device families, Cyclone PLLs do not have support for external feedback mode. All three supported

clock feedback modes allow for multiplication/division, phase shifting, and programmable duty cycle. The following sections give a brief description of each mode.

The phase relationship shown in Figure 6–4 through 6–6 are for the default phase shift setting of 0°. Changing the phase-shift setting will change the relationships.

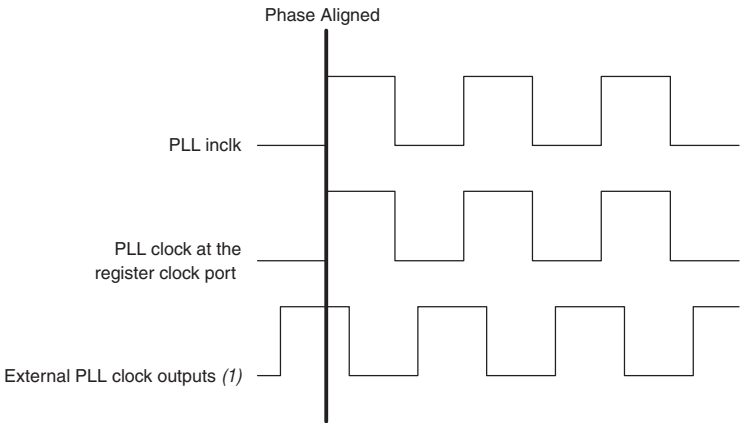
Normal Mode

In normal mode, the PLL phase aligns the input reference clock with the clock signal at the ports of the registers in the logic array or the IOE to compensate for the internal global clock network delay. In the `altpll` MegaWizard Plug-In Manager, you can define which internal clock output from the PLL (`c0` or `c1`) should be compensated.

If the external clock output (`PLL[2..1]_OUT`) is used in this mode, there will be a phase shift with respect to the clock input pin. Similarly, if you use the internal PLL clock outputs to drive general-purpose I/O pins, there will be a phase shift with respect to the clock input pin.

Figure 6–4 shows an example waveform of the PLL clocks' phase relationship in normal mode.

Figure 6–4. Phase Relationship Between PLL Clocks in Normal Mode

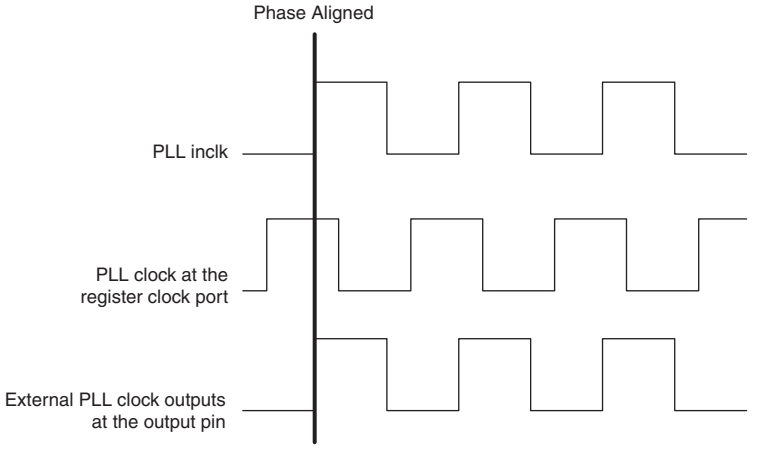


Note to Figure 6–4:
(1) The external clock output can lead or lag the PLL clock signals.

Zero Delay Buffer Mode

The clock signal on the PLL external clock output pin (PLL[2..1]_OUT) is phase-aligned with the PLL input clock pin for zero delay. If you use the c[1..0] ports to drive internal clock ports, there will be a phase shift with respect to the input clock pin. Figure 6–5 shows an example waveform of the PLL clocks' phase relationship in zero delay buffer mode.

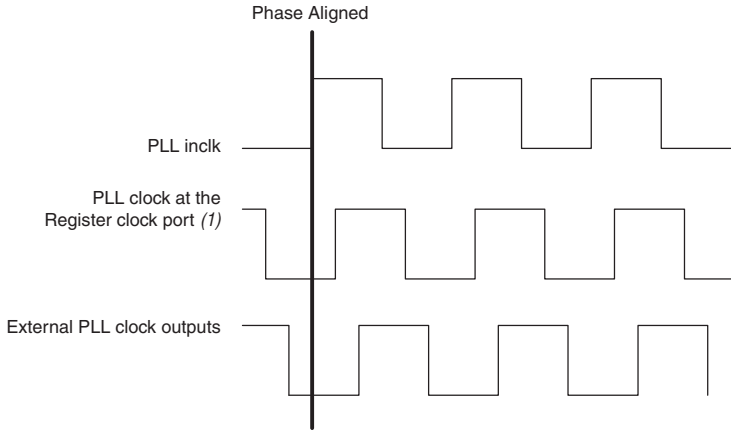
Figure 6–5. Phase Relationship Between PLL Clocks in Zero Delay Buffer Mode



No Compensation

In this mode, the PLL does not compensate for any clock networks, which leads to better jitter performance because the 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 6–6 shows an example waveform of the PLL clocks' phase relationship in no compensation mode.

Figure 6–6. Phase Relationship Between PLL Clocks in No Compensation Mode



Note to Figure 6–6:
(1) Internal clocks fed by the PLL are in phase alignment with each other.

Pins

Table 6–8 describes the Cyclone PLL-related physical pins and their functionality.

Table 6–8. Cyclone PLL Pins (Part 1 of 2)

Pin Name	Description
CLK0	Single-ended or LVDS p-pin that can drive the inclk0 port of PLL1.
CLK1 (1)	Single-ended or LVDS n-pin that can drive the inclk0 port of PLL1.
CLK2	Single-ended or LVDS p-pin that can drive the inclk0 port of PLL2.
CLK3 (1)	Single-ended or LVDS n-pin that can drive the inclk0 port of PLL2.
PLL1_OUTp (2) PLL1_OUTn (2)	Single-ended or LVDS pins driven by the e0 port from PLL1. If not used by the PLL, these are available as general-purpose I/O pins.
PLL2_OUTp (2) PLL2_OUTn (2)	Single-ended or LVDS pins driven by the e0 port from PLL2. If not used by the PLL, these are available as general-purpose I/O pins.
VCCA_PLL1 (3)	Analog power for PLL1. Even if the PLL is not used, you must connect this pin to 1.5 V.
GNDA_PLL1 (4)	Analog ground for PLL1. You can connect this pin to the GND plane on the board.
VCCA_PLL2 (3)	Analog power for PLL2. Even if the PLL is not used, you must connect this pin to 1.5 V.

Table 6–8. Cyclone PLL Pins (Part 2 of 2)	
Pin Name	**Description**
GNDA_PLL2 *(4)*	Analog ground for PLL2. You can connect this pin to the GND plane on the board.
GNDG_PLL1 *(5)*	Guard ring ground for PLL1. You can connect this pin to the GND plane on the board.
GNDG_PLL2 *(5)*	Guard ring ground for PLL2. You can connect this pin to the GND plane on the board.

Notes to Table 6–8:
(1) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.
(2) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.
(3) Refer to "Board Layout" on page 6–17 for filtering and other recommendations.
(4) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not have a separate GNDA_PLL pin. They are internally tied to GND.
(5) The Guard ring power (VCCG_PLL) is tied internally to V_{CCINT}.

Board Layout

Cyclone PLLs contain analog components that are embedded in a digital device. These analog components have separate power and ground pins to provide immunity against noise generated by the digital components. These separate VCC and GND pins are used to isolate circuitry and improve noise resistance.

V_{CCA} and GNDA

Each PLL has separate VCC and GND pairs for their analog circuitry. The analog circuit power and ground pin for each PLL is called VCCA_PLL# and GNDA_PLL# (# represents the PLL number). Even if the PLL is not used, the V_{CCA} power must be connected to a 1.5-V supply. The power connected to V_{CCA} must be isolated from the power to the rest of the Cyclone FPGA, or any other digital device on the board. The following sections describe three different methods for isolating V_{CCA}.

Separate V_{CCA} Power Plane

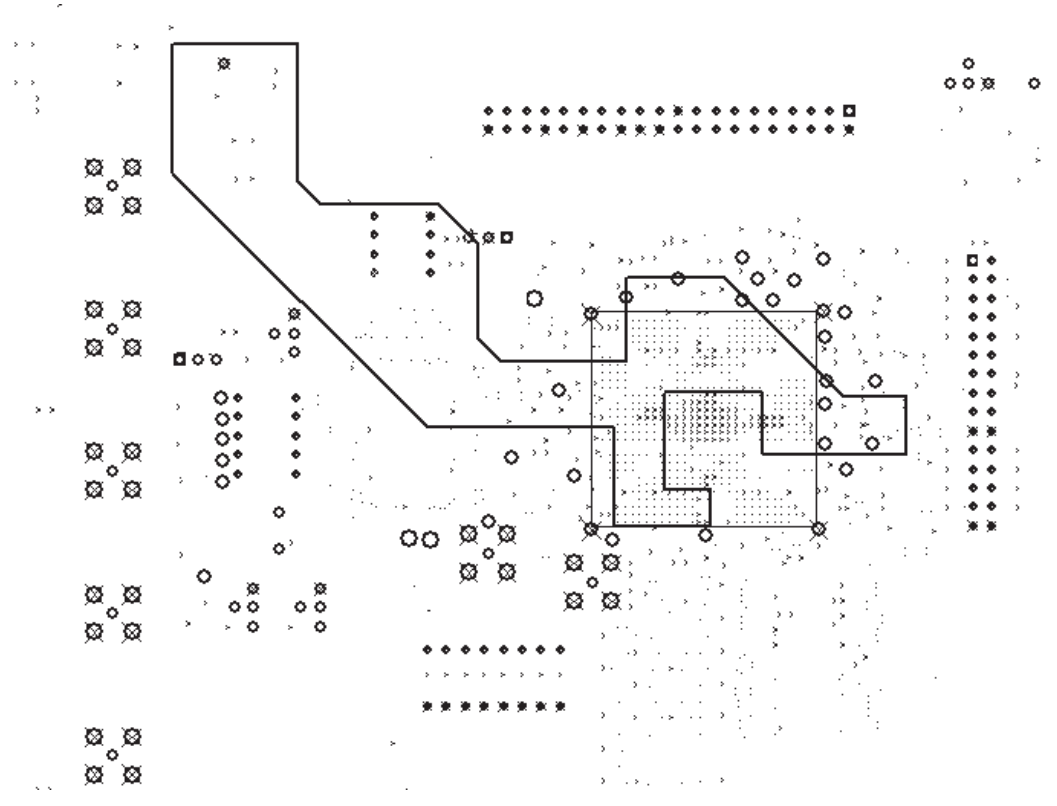
The designer of a mixed-signal system would have already partitioned the system into analog and digital sections, each with its own power planes on the board. In this case, you can connect V_{CCA} to the analog 1.5-V power plane.

Partitioned V_{CCA} Island within V_{CCINT} Plane

Most systems using Altera devices are fully digital, so there is not a separate analog power plane readily available on the board. Adding new planes to the board may be expensive. Therefore, you can create islands

for VCCA_PLL. The dielectric boundary that creates the island is approximately 25 mils thick. Figure 6–7 shows a partitioned plane within V_{CCINT} for V_{CCA}.

Figure 6–7. V_{CCINT} Plane Partitioned for V_{CCA} Island



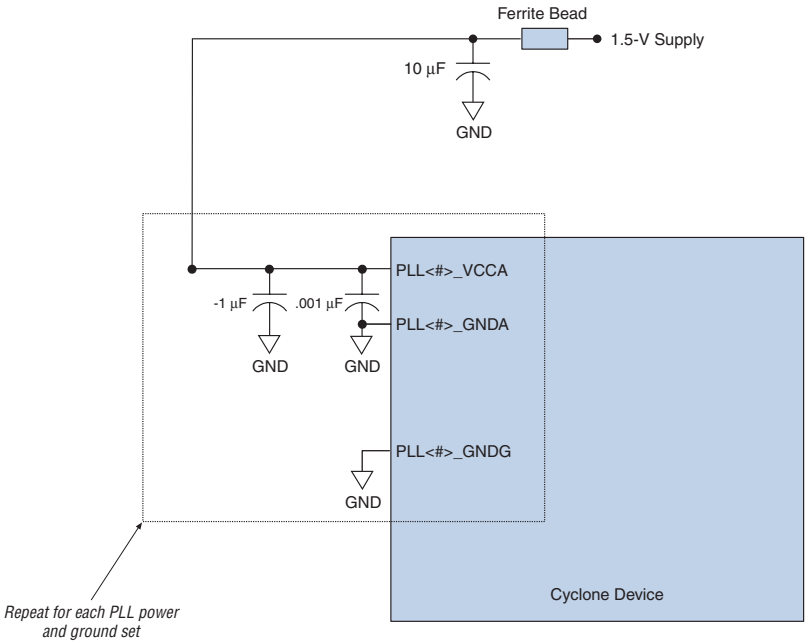
Thick V_{CCA} Traces

Due to board restraints, it may not be possible to partition a V_{CCA} island. Instead, run a thick trace from the power supply to each of the VCCA pins. The traces should be at least 20 mils thick.

In all cases, each VCCA pin must be filtered with a decoupling circuit shown in Figure 6–8. You must place a ferrite bead and a 10-µF tantalum parallel capacitor where the power enters the board. Choose a ferrite bead that exhibits high impedance at frequencies of 50 MHz or higher. Each VCCA pin must be decoupled with a 0.1-µF and a 0.001-µF parallel

combination of ceramic capacitors located as close as possible to the Cyclone FPGA. You can connect the GNDA pins directly to the same GND plane as the digital GND of the device.

Figure 6–8. PLL Power Schematic for Cyclone PLLs



For more information about board design guidelines, refer to *AN 75: High-Speed Board Designs*.

Jitter Considerations

If the input clocks have any low-frequency jitter (below the PLL bandwidth), the PLL attempts to track it, which increases the jitter seen at the PLL clock output. To minimize this effect, avoid placing noisy signals in the same V_{CCIO} bank as those that power the PLL clock input buffer. This is only important if the PLL input clock is assigned to 3.3-V or 2.5-V LVTTL or LVCMOS I/O standards. With these I/O standards, V_{CCIO}

powers the input clock buffer. Therefore, any noise on this V_{CCIO} supply can affect jitter performance. For all other I/O standards the input buffers are powered by V_{CCINT}.

Because Cyclone external clock output pins (PLL[2..1]_OUT) do not have a separate V_{CC} and GND bank, you should avoid placing noisy output signals directly next to these pins. Therefore, Altera recommends that PLL[2..1]_OUT neighboring I/O pins should be either inputs pins or not used at all. If noisy outputs are placed next to the PLL[2..1]_OUT pins, they could inject noise through ground bounce or V_{CC} sag and mutual pin inductance, which would result in worse jitter performance on the PLL[2..1]_OUT pins.

Additionally, you should take into consideration the number of simultaneously switching outputs within the same V_{CCIO} bank as the PLL[2..1]_OUT pins. Altera recommends that you switch as few outputs simultaneously in the same direction as possible in these V_{CCIO} banks. Also, if you have switching outputs in the same V_{CCIO} bank as the PLL[2..1]_OUT pins, Altera recommends that you use the low current strength and/or slow slew rate options on those output pins as they will help to improve the jitter performance.

Specifications

Refer to the *DC and Switching Characteristics* chapter of the *Cyclone Device Handbook* for Cyclone FPGA PLL specifications.

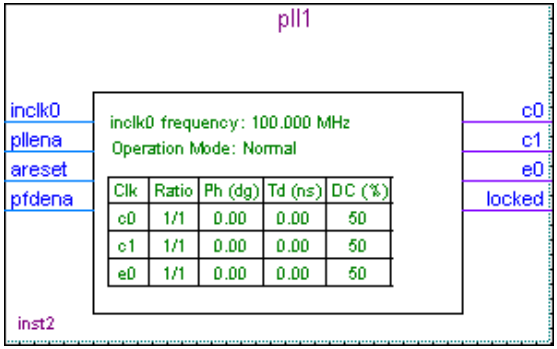
Software Support

Support for Cyclone PLLs is available in the Quartus II software by using the altpll megafunction. The following section describes how the altpll megafunction enables the various Cyclone PLL features and options. This section includes the megafunction symbol, the input and output ports, a description of the MegaWizard Plug-In Manager options, and example MegaWizard screen shots.

Quartus II altpll Megafunction

Figure 6–9 shows the altpll megafunction symbol in the Quartus II software.

Figure 6–9. altpll Megafunction Symbol Targeted for Cyclone FPGAs



Refer to Quartus II Help for the `altpll` megafunction AHDL functional prototypes (applicable to Verilog HDL), VHDL component declaration, and parameter descriptions.

altpll Input Ports

Table 6–9 shows the input ports of the altpll megafunction and describes their function.

Table 6–9. Input Ports of the altpll Megafunction

Port Name	Required	Description
inclk0 *(1)*	Yes	The input clock port that drives the PLL.
pllena *(2)*	No	pllena is an active-high signal, which acts as a combined enable and reset signal for the PLL. You can use it for enabling or disabling one or both PLLs. When this signal is driven low, the PLL clock output ports are driven to GND and the PLL loses lock. Once this signal is driven high again, the lock process begins and the PLL re-synchronizes to its input reference clock. The pllena port can be driven from internal logic or any general-purpose I/O pin.
areset *(2)*	No	areset is an active-high signal, which resets all PLL counters to their initial values. When this signal is driven high, the PLL resets its counters, clears the PLL outputs, and loses lock. Once this signal is driven low again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the areset port from internal logic or any general-purpose I/O pin.
pfdena *(2)*	No	pfdena is an active-high signal, which enables or disables the up/down output signals from the PFD. When pfdena is driven low, the PFD is disabled, while the VCO continues to operate. PLL clock outputs continue to toggle regardless of the input clock, but can experience some long-term drift. Because the output clock frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. You can drive the pfdena port from internal logic or any general-purpose I/O pin.

Notes to Table 6–9:
(1) The inclk0 port to the PLL must be driven by the dedicated clock input pin(s).
(2) See "Control Signals" on page 6–12 for further details.

altpll Output Ports

Table 6–10 shows the output ports of the altpll megafunction and describes their function.

Table 6–10. Output Ports of the altpll Megafunction

Port Name	Required	Description
c[1..0] *(1)*	No	Clock output of the PLL that drives the internal global clock network.
e0 *(1)*	No	Clock output that feeds the external clock output pins, PLL[2..1]_OUT.
locked *(2)*	No	Gives the status of the PLL lock. When the PLL is locked, this port drives logic high. When the PLL is out of lock, this port drives logic low. The locked port can pulse high and low during the PLL lock process.

Notes to Table 6–10:
(1) Either the internal or external clock output of the PLL must be selected.
(2) See "Control Signals" on page 6–12 for further details.

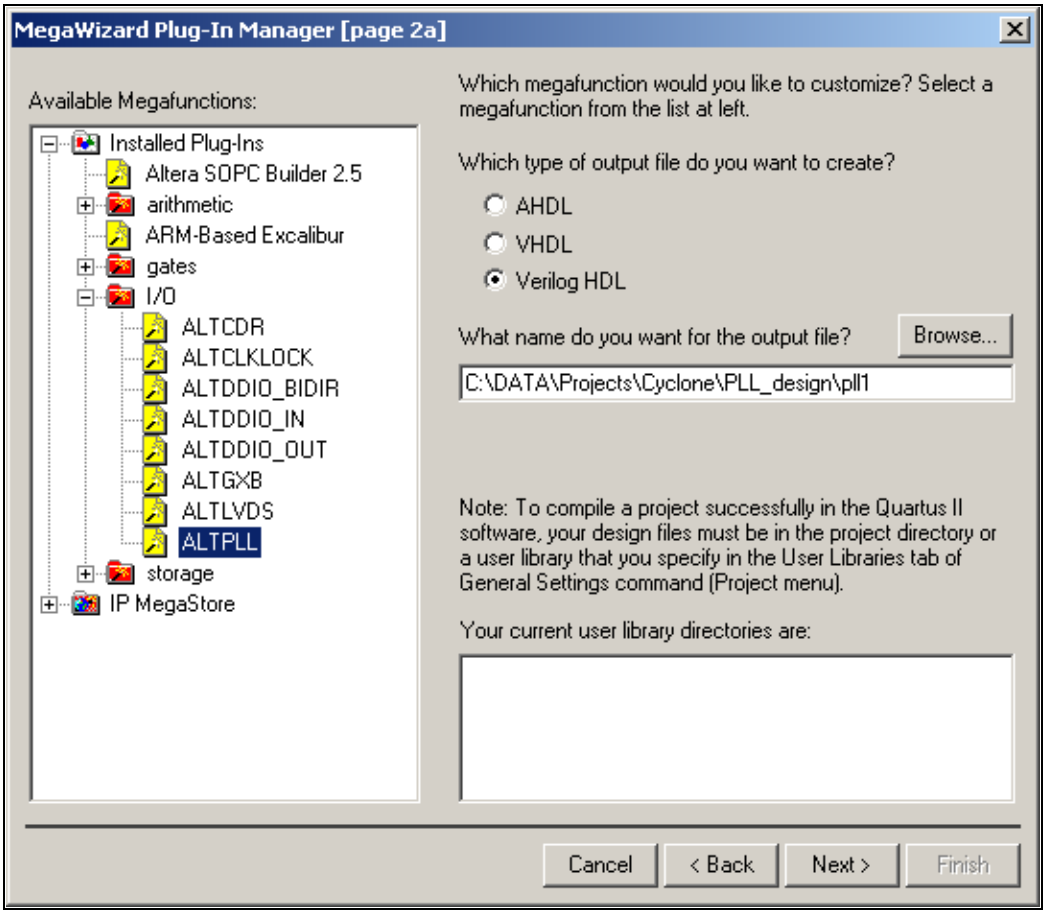
MegaWizard Customization

You can use the MegaWizard Plug-In Manager to set the altpll megafunction options for each PLL instance in your design.

If you instantiate the altpll megafunction without using the MegaWizard Plug-In Manager, search for "altpll" in the Quartus II Help for a list of the altpll parameters.

In the MegaWizard Plug-In Manager, select the altpll megafunction in the I/O directory from the **Available Megafunctions** dialog box (see Figure 6–10). The altclklock megafunction is also available from the Quartus II software for backward compatibility, but instantiates the new altpll megafunction when targeting Cyclone FPGAs.

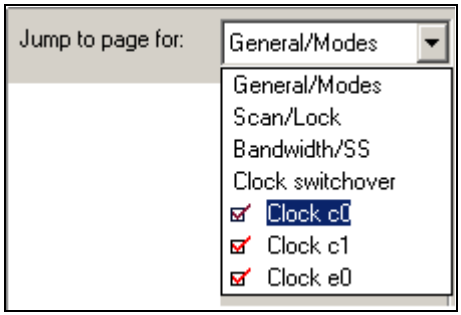
Figure 6–10. altpll Megafunction Selection in the MegaWizard Plug-In Manager



The `altpll` MegaWizard Plug-In Manager has separate pages that apply to Cyclone PLLs. The MegaWizard will gray-out options that are unavailable in Cyclone PLLs. During compilation, theQuartus II Compiler verifies the `altpll` parameters selected against the available PLLs, and any PLL or input clock location assignments.

At the top right-hand corner of each page of the `altpll` MegaWizard Plug-In Manager, there is a **jump to page** drop-down list (see Figure 6–11). This drop-down list allows you to jump to any particular `altpll` MegaWizard page and set those options.

Figure 6–11. Jump to Page Drop-Down List in the altpll MegaWizard Plug-In



MegaWizard Page Description

This section describes the options available on the `altpll` MegaWizard pages. Each of the MegaWizard pages are shown. Tables 6–11 through 6–13 describe the features or settings on that page that apply to Cyclone PLLs. Use these tables, along with the hardware descriptions of the PLL features, to determine appropriate settings for your PLL instance.

You can use the **General/Modes** (Page 1) of the `altpll` MegaWizard Plug-In Manager for selecting the target device family, clock input frequency, general control signal selection, and clock feedback operation mode (see Figure 6–12 and Table 6–11).

Figure 6–12. altpll MegaWizard Plug-In Manager (Page 1)

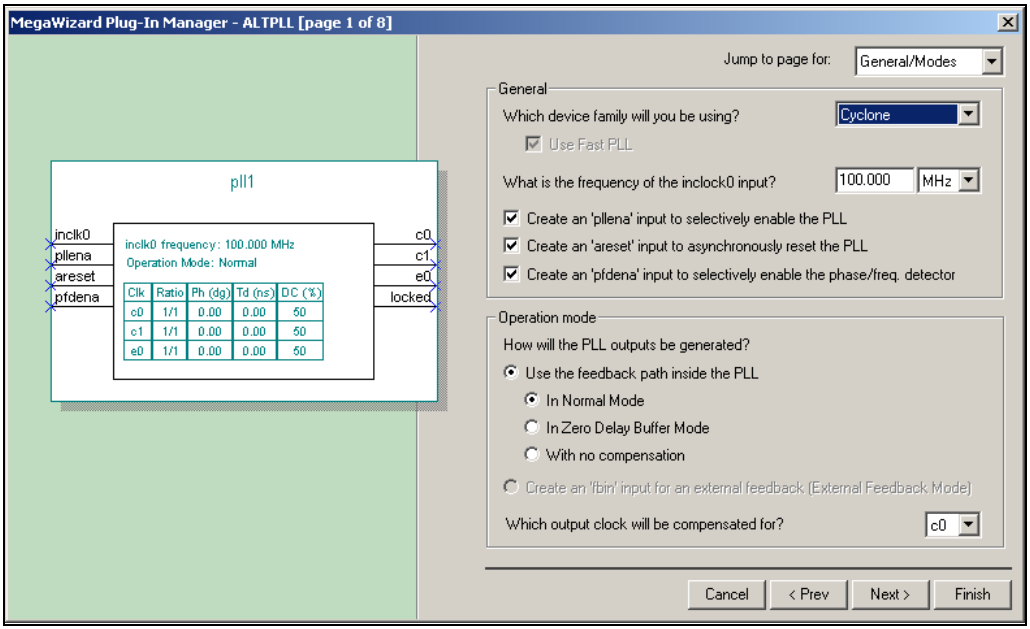


Table 6–11. altpll MegaWizard Plug-In Options Page 1 (Part 1 of 2)

Function	Description
Which device family will you be using?	This chapter explains all `altpll` options that apply when Cyclone is the target device family selected.
What is the frequency of the `inclock0` input	The frequency for the PLL input clock, `inclock0`.
Create an `pllena` input to selectively enable the PLL	Creates a `pllena` port for this PLL instance. See Table 6–9 for `pllena` port description.
Create an `areset` input to asynchronously reset the PLL	Creates a `areset` port for this PLL instance. See Table 6–9 for `areset` port description.
Create an `pfdena` input to selectively enable the PFD	Creates a `pfdena` port for this PLL instance. See Table 6–9 for `pfdena` port description.

Table 6–11. altpll MegaWizard Plug-In Options Page 1 (Part 2 of 2)

Function	Description
Use the feedback path inside the PLL	This option sets the OPERATION_MODE parameter to either normal, zero delay buffer, or no compensation mode. In normal mode, the PLL feedback path comes from a global clock network, which minimizes the clock delay to registers for that specific PLL clock output. You can specify which PLL output is compensated for by using the COMPENSATE_CLOCK parameter. In zero delay buffer mode, the PLL feedback path is confined to the dedicated PLL external output pin. The clock signal driven off-chip on the PLL_OUT pin is phase aligned with the PLL clock input for a minimized delay between clock input and external clock output. If the PLL is also used to drive the internal clock network, a corresponding phase shift of that clock network results. In no compensation mode, the PLL feedback path is confined to the PLL loop; it does not come from the global clock network or an external source. There is no clock network compensation, but this mode minimizes jitter on clocks. This mode may lead to positive hold times on IOE registers; you can use manual phase shifting to compensate for positive hold times. For more information, see "Clock Feedback Modes" on page 6–13.
Which output clock will be compensated?	Indicates which output port of the PLL is compensated. For normal mode, you can select c0 or c1.

You can use **Scan/Lock** (Page 2) for selecting the locked output port (see Figure 6–13 and Table 6–12).

Figure 6–13. altpll MegaWizard Plug-In Manager (Page 2)

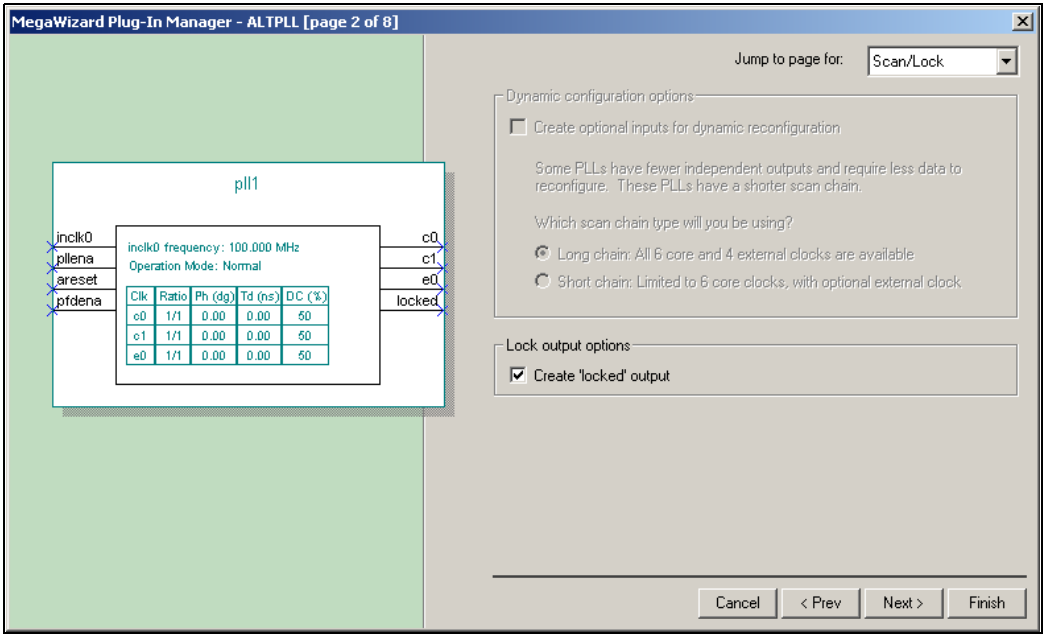
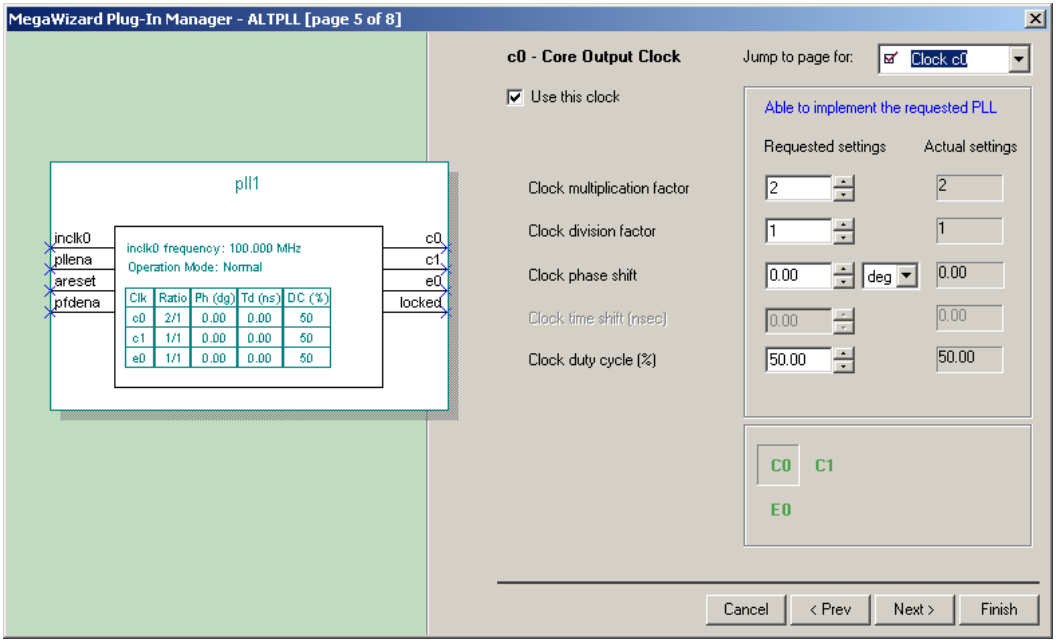


Table 6–12. altpll MegaWizard Plug-In Options Page 2

Function	Description
Create "locked" output	Creates a locked output port to indicate PLL lock. See `locked` port description in Table 6–10.

The options on the next two pages of the MegaWizard Plug-In Manager, (Pages 3 to 4, titled **Bandwidth/SS** and **Clock Switchover**) are not supported in Cyclone FPGAs.

Figure 6–14. altpll MegaWizard Plug-In Manager Pages 5 of 8



The last 3 pages of the MegaWizard Plug-In Manager (Pages 5 to 7) allow you to set the multiplication/division factors, phase shift, and duty cycle for each PLL output port (see Figure 6–14 and Table 6–13).

Each page represents the settings for one PLL clock output port. Table 6–13 describes the options for Pages 5 to 8.

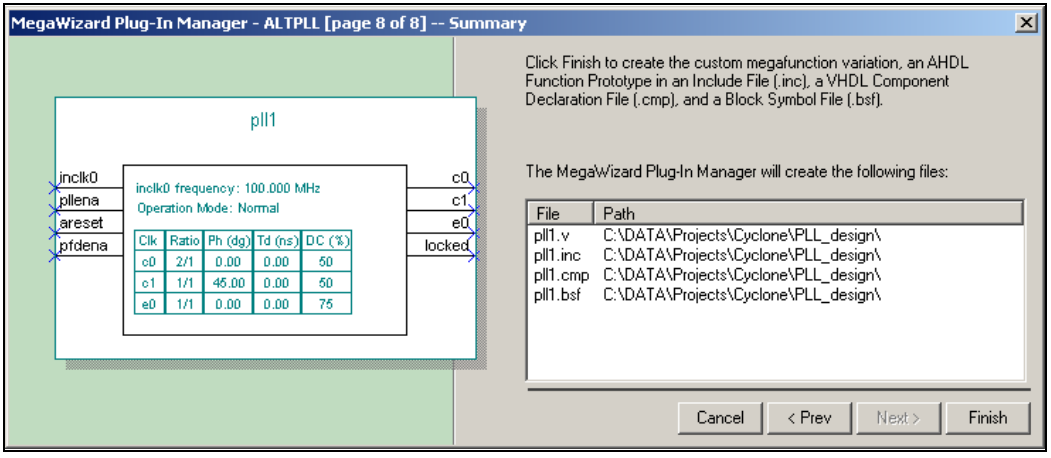
Table 6–13. altpll MegaWizard Plug-In Options Pages 5 of 8

Function	Description
Clock multiplication factor (ratio)	Specifies the clock multiplication for this PLL output. The multiplication factor cannot be greater than 32.
Clock division factor (ratio)	Specifies the clock division for this PLL output.
Clock phase shift (Ph)	Sets the programmable phase shift for the clock output with respect to the PLL clock output that is compensated. The equation to determine the precision of the phase shifting in degrees is (45° divided by the post-scale counter value). Therefore, the maximum step size is 45°, and smaller steps are possible, depending on the multiplication/division ratio necessary on the clock output port. For example, if you have an input clock of 125 MHz with ×1, the post-scale counter G0 is 3. Therefore, the smallest phase shift step is 15°, and additional phase shifting is in 15° increments. The up/down buttons cycle through the possible phase shift settings with the default *M* and post-scale dividers that the MegaWizard Pug-In Manager has chosen for your target frequency and multiplication/division ratio. It is possible to get other granularities of phase shifts if you manually enter a number into the phase shift field. For example, you can override the MegaWizard-chosen values and manually enter 7.5°. The MegaWizard Plug-In Manager verifies this is possible by using *M* = 6 and G0 = 6. The MegaWizard Plug-In Manager tries to achieve the closest phase shift possible. For example, if you enter 10°, the MegaWizard Plug-In Manager verifies that 9° is possible by using *M* = 5 and G0 = 5. For more information, see "Phase Shifting" on page 6–9.
Clock duty cycle (DC)	Specifies the clock duty cycle of the PLL clock output. The equation to determine the precision of the duty cycle is (50% divided by the post-scale counter value). For example, if post-scale counter G0 is 3, the allowed duty cycles are 50% divided by 3, equaling 16.67%. Because the `altpll` megafunction does not accept non-integer values for the duty cycle values, the allowed duty cycles are 17, 33, 50, and 67%. Due to hard limitations, a duty cycle of 84% cannot be achieved because the closest value to 100% cannot be achieved for a given counter value. However, you can achieve a duty cycle of 84% by choosing a 17% duty cycle and inverting the PLL clock output. Use the up/down buttons to cycle through all possible settings. For more information, see "Programmable Duty Cycle" on page 6–10.

Page 8 is the summary page and tells you what files the MegaWizard Plug-In Manager will create (see Figure 6–15).

You can click **Finish** at anytime while in the MegaWizard Plug-In Manager to update the files.

Figure 6–15. altpll MegaWizard Plug-In Manager Page 8

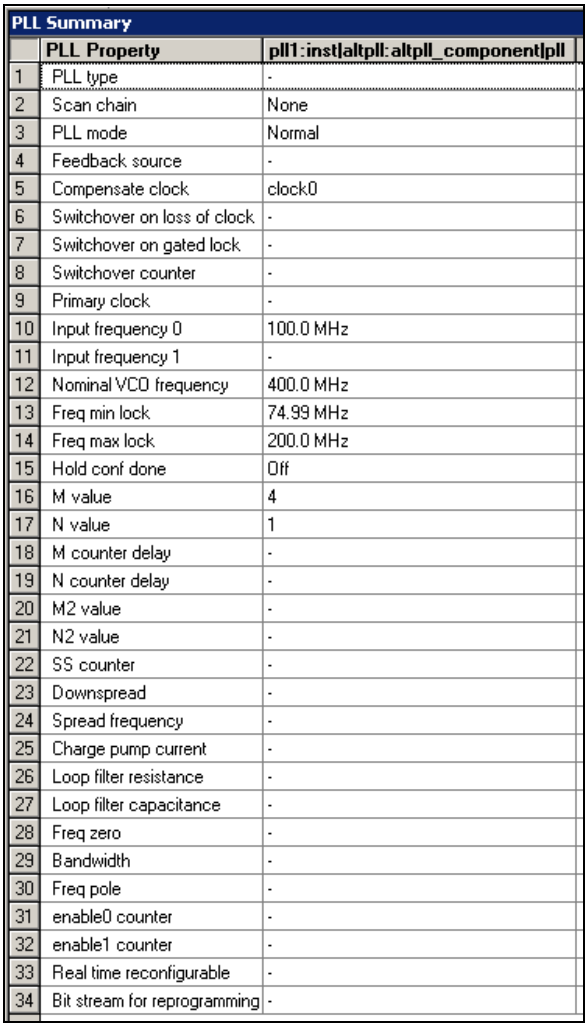


Compilation Report

During compilation, an information message displays whether the requested multiplication/division factors, and/or phase shift, and/or duty cycle were achieved. If you enter an invalid multiplication/division ratio, compilation fails, and the Quartus II software displays an error message. If you enter an invalid phase shift or duty cycle value, the compilation proceeds, and you will receive an information message displaying the best alternative values chosen by the Quartus II software.

The **Resource Section** of the compilation report provides two PLL reports: the **PLL Summary** and the **PLL Usage** reports. The **PLL Summary** provides information on each PLL's parameters (see Figure 6–16). The **PLL Summary** is column-based in the report file, where each column represents a different PLL instance. Table 6–14 lists and explains the parameters shown in the **PLL Summary** report. PLL properties not listed in Table 6–14 do not apply to Cyclone PLLs.

Figure 6–16. PLL Summary Report



PLL Summary

	PLL Property	pll1:inst|altpll:altpll_component|pll
1	PLL type	-
2	Scan chain	None
3	PLL mode	Normal
4	Feedback source	-
5	Compensate clock	clock0
6	Switchover on loss of clock	-
7	Switchover on gated lock	-
8	Switchover counter	-
9	Primary clock	-
10	Input frequency 0	100.0 MHz
11	Input frequency 1	-
12	Nominal VCO frequency	400.0 MHz
13	Freq min lock	74.99 MHz
14	Freq max lock	200.0 MHz
15	Hold conf done	Off
16	M value	4
17	N value	1
18	M counter delay	-
19	N counter delay	-
20	M2 value	-
21	N2 value	-
22	SS counter	-
23	Downspread	-
24	Spread frequency	-
25	Charge pump current	-
26	Loop filter resistance	-
27	Loop filter capacitance	-
28	Freq zero	-
29	Bandwidth	-
30	Freq pole	-
31	enable0 counter	-
32	enable1 counter	-
33	Real time reconfigurable	-
34	Bit stream for reprogramming	-

Table 6–14. PLL Summary in Compilation Report File (Part 1 of 2)

PLL Property	Description
PLL mode	Clock feedback mode
Compensate clock	Indicates which PLL clock output (`clock0`, `clock1`, or `extclock0`) port is compensated
Input frequency 0	Clock input frequency for `inclk0`

Table 6–14. PLL Summary in Compilation Report File (Part 2 of 2)	
PLL Property	**Description**
Nominal VCO frequency	Shows the VCO frequency; $f_{VCO} = f_{IN} \times M/N$
Freq min lock	Shows the minimum PLL input clock frequency for which the current combination of *M*/*N* still provides a valid VCO lock
Freq max lock	Shows the maximum PLL input clock frequency for which the current combination of *M*/*N* still provides a valid VCO lock
M value	*M* counter value
N value	*N* counter value

The **PLL Usage** report shows the breakdown information for each PLL clock output (see Figure 6–17). This report is categorized by PLL clock output ports, such that each row represents a different PLL clock output used in your design. Table 6–15 lists and explains the parameters shown in the **PLL Usage** report file in a row format. PLL parameters not listed in Table 6–15 do not apply to Cyclone PLLs.

Figure 6–17. PLL Usage Report

PLL Usage

	Name	Output Clock	Mult	Div	Output Frequency	Phase Shift	D	Duty Cycle	Counter	C	Counter Value	High / Low	Initial	VCO Tap
1	pll1:inst|altpll:altpll_component|_clk0	clock0	2	1	200.0 MHz	0 (0 ps)	0..	50/50	G1	-	2	1/1 Even	1	0
2	pll1:inst|altpll:altpll_component|_clk1	clock1	1	1	100.0 MHz	45 (1250 ps)	0..	50/50	G0	-	4	2/2 Even	1	4
3	pll1:inst|altpll:altpll_component|_extclk0	extclock0	1	1	100.0 MHz	0 (0 ps)	0..	75/25	E0	-	4	3/1 Even	1	0

Timing Analysis

Table 6–15 shows the usage in the compilation report file.

Table 6–15. PLL Usage in Compilation Report File (Part 1 of 2)	
PLL Parameter	**Description**
`Name`	Indicates the PLL instance name and clock output reported.
`Output Clock`	Indicates the PLL clock output (`clock0`, `clock1`, or `extclock0`) for which the parameter information in this row applies. This is the clock port specified in the MegaWizard Plug-In Manager (`c0`, `c1`, `e0`).
`Mult`	Overall multiplication ratio.
`Div`	Overall division ratio.
`Output Frequency`	Output frequency for this output clock.
`Phase Shift`	Achieved phase shift in degrees and units of time (can differ from user-entered value).

Table 6–15. PLL Usage in Compilation Report File (Part 2 of 2)

PLL Parameter	Description
Duty Cycle	Duty cycle for this clock output.
Counter	Post-scale counter used for this clock output, which counter (G0, G1, E0) feeds the clock output.
Counter Value	Value of post-scale counter.
High/Low	High- and low-time counts that make up the counter value. The ratio of high- and low-counts is directly proportional to the duty cycle.
Initial	Initial value for this post-scale counter (achieves the coarse granularity for phase shifting). Specifies the initial number of VCO cycles before starting the counter.
VCO Tap	VCO tap ranges from 0 to 7 (achieves fine granularity for phase shift in units of 1/8 of the VCO period).

The register-to-register timing for each PLL clock output that drives the logic array is reported with slack. The timing analysis section of the report file provides slack information in a clock requirement line for each PLL clock output.

You can derive f_{MAX} numbers from the slack reporting. The microparameters t_{CO}, t_{SU}, and the path delay are given for a `List Path` command on the Actual Maximum P2P timing in the Slack Report window. You can add and invert these to find the f_{MAX} for that path. See the following equation:

$$f_{MAX} = 1/(<\textit{register to register delay}> - <\textit{clock skew delay}> + <\textit{micro setup delay}> + <\textit{micro clock to output delay}>)$$

During timing analysis for Cyclone designs using PLLs, the project clock settings override the PLL input clock frequency and duty cycle settings. It is important to note the following:

- A warning during compilation reports that the project clock settings override the PLL clock settings.
- The project clock setting overrides the PLL clock settings for timing-driven compilation. When you compile a design with timing-driven compilation turned on, you are overconstraining the design so that the fitter can give you a better f_{MAX} performance. For example, if the PLL is set to output a 150 MHz clock, you can set a project clock setting for 170 MHz so that the fitter tries to achieve a design performance of 170 MHz.

- The Compiler checks the lock frequency range of the PLL. If the frequency specified in the project clock settings is outside the lock frequency range, the PLL clock settings will not be overridden.
- Overriding the PLL clock settings only changes the timing requirements; it does not change the overall multiplication/division and phase delay on each clock output of the PLL. The MegaWizard Plug-In Manager does not use the project clock settings to determine the `altpll` parameters.
- Performing a timing analysis without recompiling your design does not change the programming files. You must recompile your design to update the programming files.
- A Default Required f_{MAX} setting does not override the PLL clock settings. Only individual clock settings will override the PLL clock settings.

This capability is useful when you have configured a Cyclone device and want to see if your timing requirements are met when you feed the PLL a different input clock than what is specified for the PLL parameters. Therefore, this feature allows you to overwrite the PLL input clock frequency settings for timing analysis, meaning you do not have to re-synthesize or re-fit your design. The following procedure allows you to override the PLL input frequency setting and re-generate timing analysis.

1. Choose **Timing Settings** (Project menu).
2. Click on the **Clock Settings** tab.
3. Under **Specify circuit frequency as**, select **Settings for individual clock signals**.
4. Click **New**.
5. In the **New Clock Settings** dialog box, type a <name> for the new clock settings in the **Clock settings** box.
6. If you want to specify timing requirements for an absolute clock, follow these steps:
    a. Under **Relationship to other clock settings**, select **Independent of other clock settings**.
    b. In the **Required fMAX** box, type the required frequency (f_{MAX}) of the clock signal and select a time unit from the list.

c. In the **Duty Cycle** list, specify the required duty cycle for the clock.

Cyclone PLLs accept input clocks with 40 to 60% duty cycle.

d. If you want to include external delays to and from device pins in the f_{MAX} calculations, turn on **Include external delays to and from device pins in fMAX calculations**.

e. Click **OK**.

7. Click **OK** to close the Timing Settings window.
8. Open the **Assignment Organizer** dialog box (Tools menu).
9. Click on the **By Node** tab.
10. Under *Mode*, select **Edit specific entity & node settings for**.
11. If necessary, copy a specific PLL input clock pin name to the **Name** box using the **Node Finder** dialog box.
12. Under **Assignment Categories**, click the + icon next to **Timing**.
13. Click on **Click here to add a new assignment**.
14. Under **Assignment**, select **Clock Settings** in the **Name** list, and select the <name> of the clock settings you created in step 5.
15. Under **Stored in assignments for**, select **This instance only**, **This instance in all occurrences of its parent entity**, or **Other**.
16. Click **Add**.
17. Click **OK** or **Apply**.
18. Select **Start Timing Analysis (**Processing Menu).

Simulation

The altpll megafunction supports behavioral and timing simulation in both the Quartus II software and supported third-party simulation tools. You can simulate all digital aspects of the PLL, but none of the analog aspects. Simulation supports all control signals and clock outputs. Table 6–16 explains the simulation support for altpll.

Table 6–16. altpll Simulation Support for Cyclone FPGAs

Feature	Simulation Support
pllena	The pllena signal is modeled. When this signal is driven low, the PLL loses lock and the PLL clock outputs are driven to logic low.
areset	The areset signal is modeled. When this signal is driven high, the PLL loses lock and the PLL clock outputs are driven to logic low. Frequency over-shoot on the PLL clock outputs is not modeled.
pfdena	The pfdena control signal is modeled. When this signal is driven low, the PLL's locked output is undefined and the PLL clock outputs continue to toggle at their last set frequency. The finite frequency long-term drift of the VCO is not modeled.
locked	The locked signal is modeled for a high-bandwidth condition only. The PLL locks or relocks within 2 to 10 cycles during simulation, and does not necessarily reflect the real lock time.
Frequency input change	If the input frequency of the PLL is changed in simulation, the model checks that $f_{IN} \times (M/N)$ is within the VCO frequency range and loses lock if outside the VCO operating range.
Jitter	Jitter is not modeled in simulation.

You can use the altpll behavioral model to simulate the Cyclone PLLs. The Cyclone behavioral model instantiation must follow the same guidelines and restrictions as the design entry. The altpll behavioral and timing models do not simulate jitter, lock time, or VCO drift.

The behavioral models for altpll reside in the **\quartus\eda\sim_lib directory**. **ALTERA_MF.VHD** contains the VHDL behavioral models and can be used for Cyclone designs that instantiate altpll. **ALTERA_MF.v** contains the Verilog HDL behavioral models. The behavioral model does not perform parameter error checking, and you must specify only valid values.

You must set the resolution of the simulator to units of pico seconds (ps) to simulate the model successfully. A larger resolution rounds off the calculations, providing incorrect results.

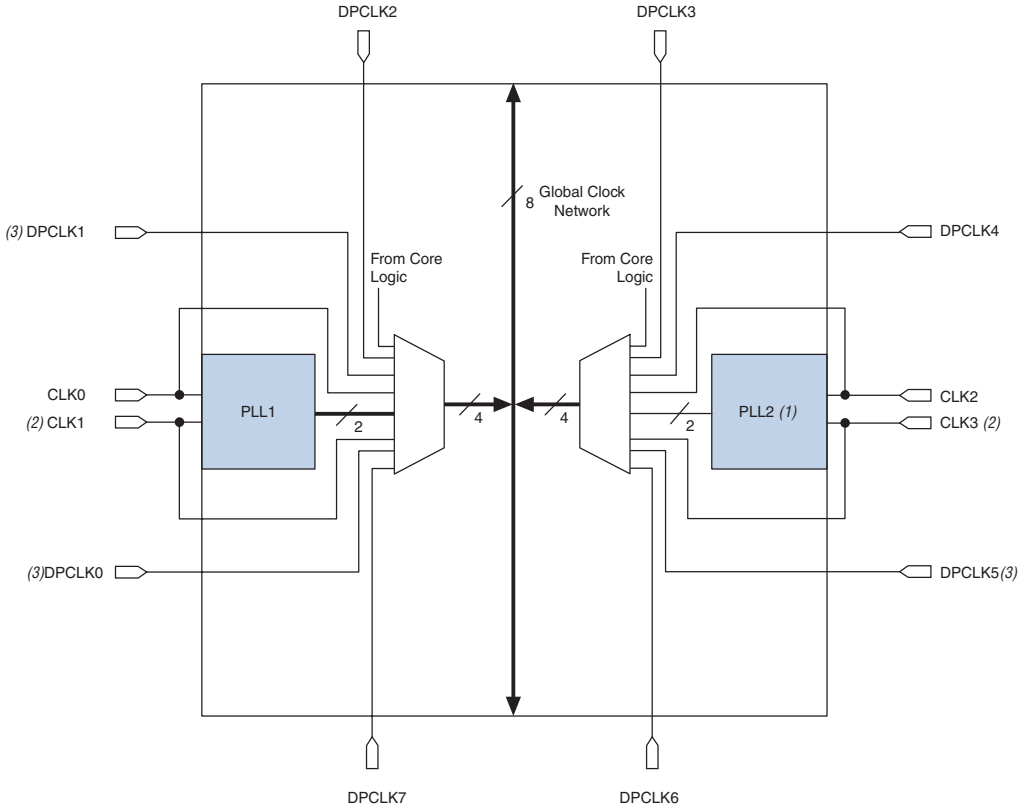
Global Clock Network

Cyclone FPGAs have eight global clock networks. The four dedicated clock input pins (CLK[3..0]), eight dual-purpose clock pins (DPCLK[7..0]), and PLL clock outputs can drive the global clock networks. In addition, internal logic for internally-generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout can drive the global clock networks.

The eight global clock lines that comprise the global clock network drive throughout the entire device. You can use the global clock network as clock sources for all device resources, including IOEs, logic elements (LEs), and memory blocks. You can also use global clock resources for control signals, such as clock enables and synchronous or asynchronous clears fed from external pins.

Figure 6–18 shows the global clock network resources.

Figure 6–18. Global Clock Generation



Notes to Figure 6–18:

(1) The EP1C3 device contains PLL1 only.

(2) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins `CLK1` and `CLK3`.

(3) The EP1C3 device in the 100-pin TQFP package has five `DPCLK` pins (`DPCLK2`, `DPCLK3`, `DPCLK4`, `DPCLK6`, and `DPCLK7`). For more information, see "“Dual-Purpose Clock I/O Pins” on page 6–40.

Dedicated Clock Input Pins

Cyclone FPGAs have up to four dedicated clock input pins (CLK[3..0], two on the left and right side of the device. You can use the CLK[3..0] pins to drive the PLLs, or directly drive them onto the global clock network. Table 6–17 shows which clock pins drive which global clock network.

Table 6–17. Dedicated Clock Input Pin Connections to Global Clock Network

Clock Input Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
CLK0	✓	—	✓	—	—	—	—	—
CLK1 *(1)*	—	✓	—	✓	—	—	—	—
CLK2	—	—	—	—	✓	—	✓	—
CLK3 *(1)*	—	—	—	—	—	✓	—	✓

Note to Table 6–17:
(1) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.

Dual-Purpose Clock I/O Pins

Cyclone FPGAs can have up to eight dual-purpose clock pins, DPCLK[7..0] (two on each side of the device). These dual-purpose pins can connect to the global clock network. You can use the DPCLK[7..0] pins for high fanout control signals, such as asynchronous clears, presets, clock enables, or protocol control signals (e.g., TRDY and IRDY for PCI, or DQS signals for external memory interfaces). These pins are also available as general-purpose I/O pins, meaning they can be inputs, outputs, or bidirectional pins. Table 6–18 shows which dual-purpose clock pins drive which global clock network in Cyclone FPGAs.

Table 6–18. Dual-Purpose Clock I/O Connections to the Global Clock Network (Part 1 of 2)

Dual-Purpose Clock Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
DPCLK0 *(1)*	—	—	—	✓	—	—	—	—
DPCLK1 *(1)*	—	—	✓	—	—	—	—	—
DPCLK2	✓	—	—	—	—	—	—	—
DPCLK3	—	—	—	—	✓	—	—	—
DPCLK4	—	—	—	—	—	—	✓	—

Table 6–18. Dual-Purpose Clock I/O Connections to the Global Clock Network (Part 2 of 2)

Dual-Purpose Clock Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
DPCLK5 *(1)*	—	—	—	—	—	—	—	✓
DPCLK6	—	—	—	—	—	✓	—	—
DPCLK7	—	✓	—	—	—	—	—	—

Note to Table 6–18:
(1) The EP1C3 device in the 100-pin TQFP package does not have the DPCLK0, DPCLK1, or DPCLK5 pins.

Combined Sources

Table 6–19 shows which combined sources drive which global clock network.

Table 6–19. Global Clock Network Sources (Part 1 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
PLL Counter Outputs	PLL1 G0	—	✓	✓	—	—	—	—	—
	PLL1 G1	✓	—	—	✓	—	—	—	—
	PLL2 G0 *(1)*	—	—	—	—	—	✓	✓	—
	PLL2 G1 *(1)*	—	—	—	—	✓	—	—	✓
Dedicated Clock Input Pins	CLK0	✓	—	✓	—	—	—	—	—
	CLK1 *(2)*	—	✓	—	✓	—	—	—	—
	CLK2	—	—	—	—	✓	—	✓	—
	CLK3 *(2)*	—	—	—	—	—	✓	—	✓

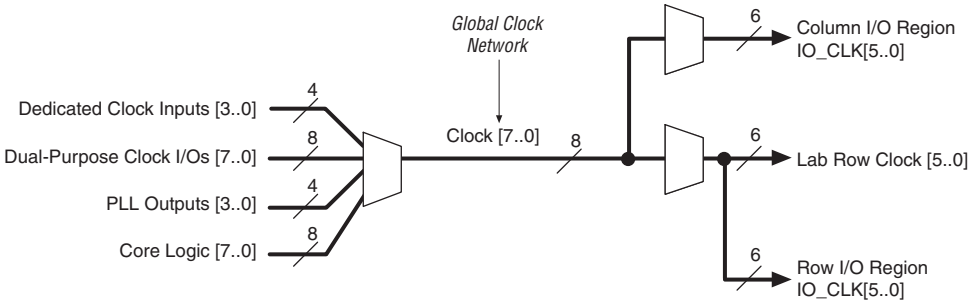
Table 6–19. Global Clock Network Sources (Part 2 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
Dual-Purpose Clock Pins	DPCLK0	—	—	—	✓	—	—	—	—
	DPCLK1 *(3)*	—	—	✓	—	—	—	—	—
	DPCLK2	✓	—	—	—	—	—	—	—
	DPCLK3	—	—	—	—	✓	—	—	—
	DPCLK4	—	—	—	—	—	—	✓	—
	DPCLK5	—	—	—	—	—	—	—	✓
	DPCLK6	—	—	—	—	—	✓	—	—
	DPCLK7	—	✓	—	—	—	—	—	—

Notes to Table 6–19:
(1) The EP1C3 device only has PLL1.
(2) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.
(3) The EP1C3 device does not have DPCLK1.

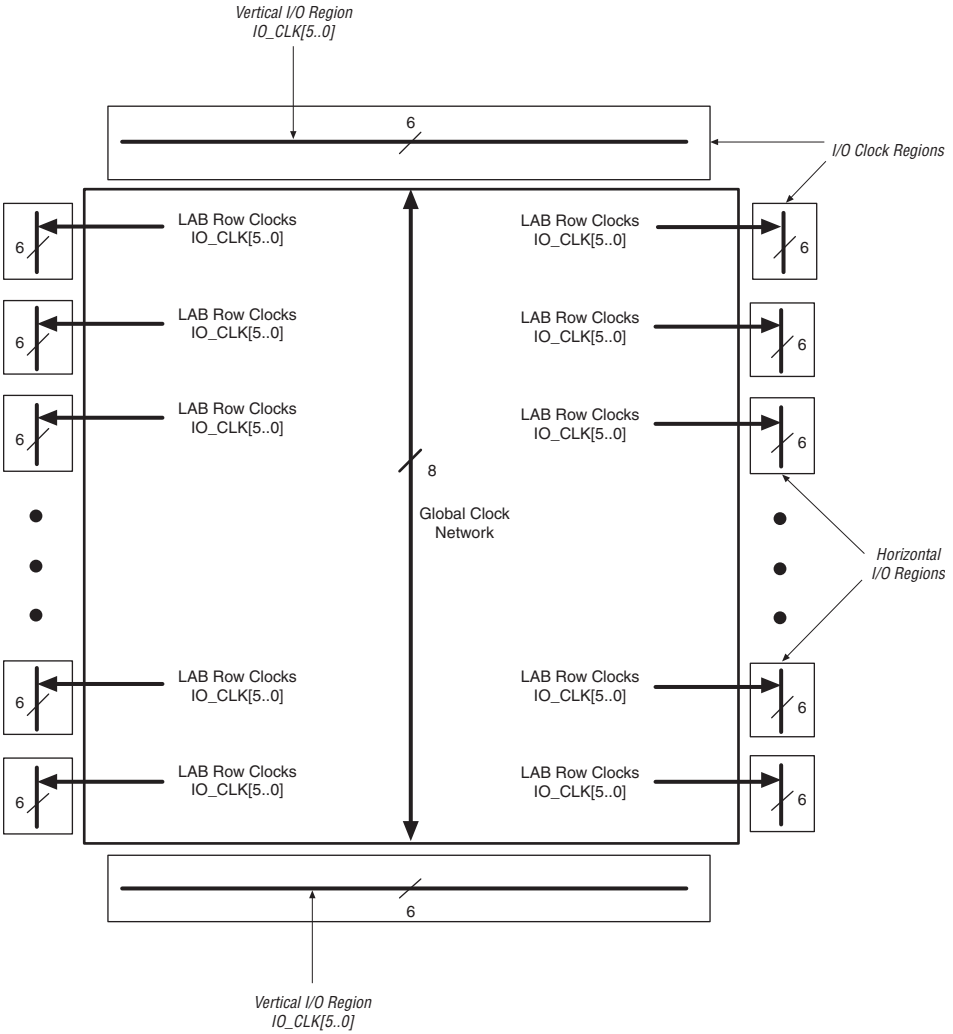
In the Cyclone FPGA, there are eight distinct dedicated global clock networks. Multiplexers are used with these clocks to form six-bit buses to drive LAB row clocks, column IOE clocks, or row IOE clocks (see Figure 6–19). Another multiplexer is used at the LAB level to select two of the six row clocks to feed the LE registers within the LAB.

Figure 6–19. Global Clock Network Multiplexers



IOE clocks have horizontal (row) and vertical (column) block regions that are clocked by six I/O clock signals chosen from the eight global clock resources. Figure 6–20 shows the I/O clock regions.

Figure 6–20. I/O Clock Regions



Conclusion

Cyclone PLLs provide significant features such as $M/(N \times \text{post-scale})$ multiplication/division, phase shift, and programmable duty cycle for your cost-sensitive clock synthesis applications. The reduction in clock delay, and the elimination of clock skew within the device, improves design speed. Cyclone PLL features simplify board design by running the internal logic of the device at a faster rate than the input clock frequency.

Referenced Documents

This chapter references the following documents:

- *AN 75: High-Speed Board Designs*
- *DC and Switching Characteristics* chapter of the *Cyclone Device Handbook*

Document Revision History

Table 6–20 shows the revision history for this chapter.

Table 6–20. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.5	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.4	• Added document revision history. • Updated information about `plena` signal in "Control Signals" section. • Updated "Zero Delay Buffer Mode" section. • Updated Figure 6–5.	—
August 2005 v1.3	Minor updates.	—
October 2003 v1.2	Updated phase shift information.	—
July 2003 v1.1	Updated input and output frequency specifications.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—