Phase-Locked Loops (ALTPLL)

Megafunction User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Document Version: 7.0
Document Date: December 2008

Contents

1. About this Megafunction

Device Family Support

The Phase-Locked Loops megafunction (also known as ALTPLL) supports the following target Altera® device families:

- Stratix® series
- Cyclone® series
- HardCopy® series

Introduction

As design complexities increase, the use of vendor-specific intellectual property (IP) blocks has become a common design methodology. Altera provides parameterizable megafunctions that are optimized for Altera device architectures. Using megafunctions instead of coding your own logic saves valuable design time. The Altera-provided functions offer more efficient logic synthesis and device implementation. You can scale the size of the megafunction by setting various parameters.

Features

The ALTPLL megafunction configures the phase-locked loops (PLLs) in the Stratix and Cyclone series of devices. Table 1–1 shows the key features of the ALTPLL megafunction. Not all features are supported by each device family. Refer to the device handbook of the device you are using for details on which features are supported.

Table 1–1. ALTPLL Megafunction Features (1 of 3)

Feature	Port/Parameter	Description
PLL enable input	`pllena`	This option adds an active high enable signal to the PLL. When the PLL is disabled, the PLL does not output clock signals.
Asynchronous reset	`areset`	This option adds an asynchronous reset to the PLL. The active high input resets the PLL when enabled.

Table 1–1. ALTPLL Megafunction Features (2 of 3)

Feature	Port/Parameter	Description

LVDS mode	`enable0`
	`enable1`
	`sclkout0`
	`sclkout1`
		LVDS is used to transmit and receive high-speed differential data. It converts data from high-speed serial signals off chip to low-speed parallel signals on chip.
		
		The LVDS receiver is designed to take a high-speed differential serial data stream from a pair of input pins and convert it to a low-speed parallel stream. The LVDS transmitter is designed to take a parallel stream of data from the core and convert it to a serial stream of transmission through a pair of high-speed output pins.
		
		Both circuits require a PLL to provide a high-speed clock for the serial data, as well as a low-speed clock for the parallel data. The receivers and transmitters can share a common PLL, or they may use separate PLLs.
		
		Note that this option is only useful when used in conjunction with the ALTLVDS megafunction. This option does not generate the LVDS TX/RX modules and is merely used as the clocking scheme for these modules.
		
		For more information about LVDS, refer to the *LVDS Megafunction User Guide*. There is an option to set up the PLL in LVDS mode using the ALTPLL megafunction. This option is only available for Stratix II, Stratix II GX, and HardCopy II devices.

Operation mode	`OPERATION_MODE`	Stratix series PLLs can compensate for both on-chip and off-chip delays in the clock path. All Cyclone series PLLs can compensate for on-chip delays. You can specify the following modes: ■ **Normal mode**—aligns the PLL input pin with the register clock. ■ **Source-Synchronous mode**—maintains the same phase relationship for data and clocks that arrive at the same time at the clock and data ports of any I/O element (IOE) input register. ■ **Zero delay buffer mode**—aligns the PLL input pin with the PLL output pin. ■ **External feedback mode**—aligns the PLL input pin with the PLL feedback pin. *(1)* ■ **No compensation mode**—provides jitter performance but does not align the PLL input pin. *(2)* Because Stratix series PLLs can have multiple outputs, you must specify which output clock is used for the feedback. *(3)* For more information about ALTPLL megafunction operation modes, refer to Table 3–3.

Table 1–1. ALTPLL Megafunction Features (3 of 3)

Feature	Port/Parameter	Description
Dynamic configuration options	SCAN_CHAIN	Stratix series PLLs can be dynamically reconfigured by using a scan chain. Depending on the PLL functionality you require, two options are available for the scan chain, long or short. The long chain (10 counters wide) allows the configuration of all six core and four external clocks. The short chain (six counters wide) limits the configuration to the six core clocks. *(4)*
Bandwidth	BANDWIDTH_TYPE	This option allows you to specify the bandwidth of the PLL. By default, this option is set to **auto**. You can either specify the bandwidth using the three provided presets (**LOW**, **MEDIUM**, or **HIGH**), or you can manually specify the bandwidth using the **custom** setting.
Spread spectrum	DOWN_SPREAD, SPREAD_FREQUENCY	This option is used to help reduce electro-magnetic interference (EMI) emissions. The output frequency varies by the down spread percentage below the target frequency. For the exact frequency specification, refer to the PLL chapter in the specific device handbook.
Clock switchover options	clkswitch, clkloss, clkbad	The clock switchover circuit in the enhanced PLL can switch between two input clocks. To activate this functionality, you must enable the inclock1 port and specify the events that cause the PLL to switch its input clock. You can set the PLL to switch automatically when the clock goes bad (clkbad) or when the PLL has lost lock (clkloss). You can also create a clkswitch port. Toggling the clkswitch port causes the PLL to switch the input clock after the specified number of input clock cycles. *(5)*
Clock multiplication factor	CLK[]_MULTIPLY_BY	This option sets the multiplication factor for the output clock. The ALTPLL megafunction displays the actual setting that the PLL uses.
Clock division factor	CLK[]_DIVIDE_BY	This option sets the division factor for the output clock. The ALTPLL megafunction displays the actual setting that the PLL uses.
Clock phase shift	CLK[]_PHASE_SHIFT	This option sets the phase shift for the output clock. The ALTPLL wizard displays the actual setting that the PLL uses.
Clock duty cycle	CLK[]_DUTY_CYCLE	This option sets the duty cycle of the output clock. The output clock is high for the specified percentage of the period. The possible duty cycles are dependent on the input frequency. The ALTPLL megafunction displays the actual setting that the PLL uses.
Clock enable	clkena[]	Each clock output port can have an enable. When the clock is disabled, the voltage controlled oscillator (VCO) continues to operate, but no clock output signal is generated. *(6)*

Notes to Table 1–1:

(1) Cyclone series devices do not support this feature.

(2) Compensated output clocks e[3..0] are not applicable to Stratix II and Cyclone II devices (these devices have only c[] outputs). Refer to the specific device handbook for other devices.

(3) Stratix III, Stratix II, Cyclone III, Cyclone II, and HardCopy II devices also support source-synchronous mode.

(4) This feature is supported by Stratix, Stratix GX, and HardCopy Stratix Enhanced PLLs, which support a distinction between long and short chains. However, scan-chain functionality is not applicable to all device families. In Stratix II and Stratix II GX PLLs, both Fast and Enhanced PLLs support normal dynamic reconfiguration. In Stratix III devices, both Top and Bottom PLLs and Left and Right PLLs support normal dynamic reconfiguration and dynamic phase reconfiguration. Cyclone III devices support only one type of PLLs. These Cyclone III PLLs support both types of dynamic reconfiguration. Refer to the PLL chapter in the relevant device handbook for additional details.

(5) Stratix III, Stratix II, Cyclone III, Cyclone II, and HardCopy II devices also support manual switchover. For more information, refer to the PLL chapter in the specified device handbook.

(6) Stratix III, Stratix II, Cyclone III, Cyclone II, and HardCopy II devices do not support this feature. Clock enable functionality can be achieved when using the ALTCLKCTRL megafunction.

For more information about the ALTPLL megafunction ports and parameters, refer to Chapter 3, Specifications.

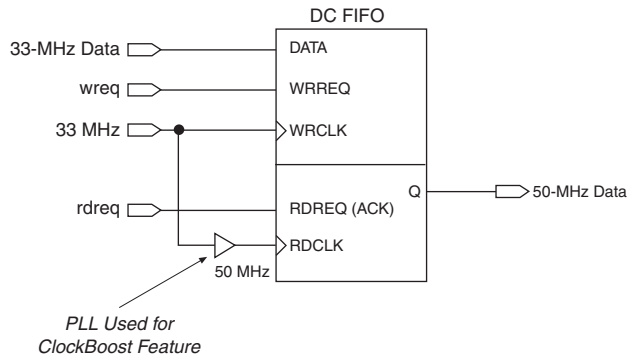
Clock Domain Transfers

For data transfer across clock domains, certain design considerations need to be made when using PLL clocks.

Asynchronous Transfers

For asynchronous register-to-register transfers (for example, 50 MHz to 33 MHz) use the appropriate asynchronous design techniques to transfer data from one clock domain to the other clock domain. For example, you can use the DC FIFO buffer for data transfer (Figure 1–1).

Figure 1–1. Using DC FIFO to Interface between Asynchronous Clock Domains



ClockBoost Feature

If you use shifted and non-shifted clocks in a register-to-register transfer, the f_{MAX} might be reduced or a hold time violation may occur. This depends on the direction and magnitude of the shift (any positive shift past 180° can be considered negative shift) and whether the destination or source register's clock is shifted.

Table 1–2 shows the key features available in enhanced and fast PLLs.

Table 1–2. Key Features for PLLs (1 of 2)

Feature	Stratix II PLLs		Stratix PLLs		Cyclone II PLLs	Cyclone PLLs
	Enhanced PLL	Fast PLL	Enhanced PLL	Fast PLL		
Clock multiplication and division *(1)*	$m \div (n \times$ post-scale counter$)$			$\frac{m}{\text{post-scale counter}}$	$m \div (n \times$ post-scale counter$)$	
Number of clock outputs per PLL	6 *(2)*	4	10	3	3 *(3)*	3
Number of internal clock outputs per PLL	6	4	6	3 *(4)*	3	2

Table 1–2. Key Features for PLLs (2 of 2)

Feature	Stratix II PLLs		Stratix PLLs		Cyclone II PLLs	Cyclone PLLs
	Enhanced PLL	Fast PLL	Enhanced PLL	Fast PLL		
Number of dedicated external clock outputs (PLL#_OUT) per PLL	3 differential/6 single-ended	*(5)*	4 differential/ 8 single-ended *(6)*	*(5)*	1 single-ended or differential	1 *(7)*
Number of feedback clock inputs per PLL	1 single-ended or differential	—	1 single-ended or differential *(8)*	—	—	—
Phase shift *(9)*	Down to 125-ps increments	Down to 125-ps increments	Down to 156.25-ps increments	Down to 125-ps increments	Down to 125-ps increments	Down to 125-ps increments
Advanced control signals (pllena, areset, pfdena)	✓	✓	✓	✓	✓	✓
Programmable duty cycle	✓	✓	✓	✓	✓	✓
Gated lock	✓	✓	—	—	✓	—
Automatic clock switchover	✓	—	✓	—	—	—
Manual clock switchover	✓	✓	✓	—	✓	—
Programmable bandwidth	✓	✓	✓	—	—	—
PLL reconfiguration	✓	✓	✓	—	—	—
Reconfigurable bandwidth	✓	✓	—	—	—	—
Spread spectrum clocking	✓	—	✓	—	—	—

Notes to Table 1–2:

(1) For *m*, *n*, and post-scale counter values, refer to the PLL chapter in the appropriate device family handbook.

(2) PLL output counters can drive the internal clock networks or dedicated external clock output pins.

(3) The Cyclone II PLL has three output counters that drive the global clock network. One of these output counters (c2) can drive a dedicated external clock output pin (single ended or differential). This counter output can also drive the external clock output and internal global clock network at the same time.

(4) PLLs 7, 8, 9, and 10 have two output ports per PLL. PLLs 1, 2, 3, and 4 have three output ports per PLL. On Stratix GX devices, PLLs 3, 4, 9, and 10 are not available for general-purpose use.

(5) The PLL clock outputs of fast PLLs can drive to any I/O pin to be used as an external clock output. For high-speed differential I/O pins, the device uses a data channel to generate the transmitter output clock (txclkout).

(6) Every Stratix and Stratix GX device has two enhanced PLLs (PLLs 5 and 6) with either eight single-ended outputs or four differential outputs each. Two additional enhanced PLLs (PLLs 11 and 12) in EP1S80, EP1S60, EP1S40 (PLLs 11 and 12 are not supported for the F780 package), and EP1SGX40 devices each have one single-ended output.

(7) 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.

(8) Feedback clock input is supported in PLLs 5 and 6 only.

(9) The smallest phase shift increment is determined by the VCO period divided by eight. For VCO ranges, see the relevant chapter in the appropriate device family handbook.

General Description

The ALTPLL megafunction easily configures the PLLs in Altera devices. PLLs are used for clock management. Stratix series devices have two types of PLLs. Cyclone series devices have one type of PLL. Table 1–2 on page 1–4 shows the features available in the enhanced and fast PLLs.

Stratix III and Cyclone III PLL New Features Description

The Stratix III PLL (top and bottom type and left and right type) and the Cyclone III PLL are redesigned versions of the Stratix II and Cyclone II PLL. The new key features of the Stratix III and Cyclone III PLL are described below.

Dynamic Reconfiguration

There are two ways to reconfigure the Stratix III and Cyclone III PLL: reconfiguring just the phase or reconfiguring all the internal PLL settings, excluding the phase settings.

Phase reconfiguration is new in Stratix III and Cyclone III devices and has a much simpler user interface than reconfiguring all of the internal PLL settings.

For details about PLL reconfiguration in Stratix III or Cyclone III devices, refer to the *altpll_reconfig Megafunction User Guide* and the *Clock Networks & PLLs* chapter in the *Stratix III Device Handbook* or the *Cyclone III Device Handbook*.

PLL Types

The two Stratix III PLL types (top and bottom type and left and right type) are almost the same. The analog portions are identical (in other words, they have the same bandwidth configuration, VCO ranges, and so forth), with small differences in the digital portion (for example, more counters on the top and bottom than left and right).

Cyclone III devices, like Cyclone II devices, have only one type of PLL.

For more information about PLL top and bottom type and left and right type, refer to the *Clock Networks & PLLs* chapter in the *Stratix III Device Handbook* or the *Cyclone III Device Handbook*.

LVDS Clock

In Stratix III devices, the `LVDSCLK` and `LOADEN` paths are driven directly by the regular counter outputs, unlike Stratix II devices, where special `sclkout[]` and `enable[]` outputs from the PLL are used.

For more information about LVDS clocks, refer to the *Clock Networks & PLLs in Stratix III Devices* chapter in the *Stratix III Device Handbook*.

Clock Switchover

Similar to Stratix II devices, Stratix III devices support manual switchover and automatic switchover with manual override. However, the Stratix III device switchover is simpler and more symmetric than the Stratix II device switchover.

In Stratix II devices, there is no switchover counter or switchover-on-loss-of-lock. The auto-switchover occurs only once, causing a switch from the primary clock to the secondary clock, and you are required to switch the PLL back to the primary clock manually.

In Stratix III and Cyclone III devices, you can create a counter using core resources and core logic used to switch on loss-of-lock. Switchover is symmetric: if the clock is lost on one input, the PLL switches to the other input, and continues this switching indefinitely. You can still manually override the switchover circuit in auto-switchover mode.

In addition, the status signals, such as `CLKBAD` and `ACTIVECLK`, only operate when at least one good input clock exists. For the status signals to be correct, the frequency of each input clock must be within two times the frequency of the other clock input. This restriction applies to Stratix III and Cyclone III PLLs.

For more information about clock switchovers in Stratix III and Cyclone III device PLLs, refer to the *Clock Networks & PLLs* chapter in the *Stratix III Device Handbook* or the *Cyclone III Device Handbook*.

Post-Scale Counter Cascading and Cascading PLLs

Both Stratix III and Cyclone III device PLLs support post-scale counter cascading. Stratix II and Stratix II GX PLLs have this capability but Cyclone II PLLs do not. Post-scale counter cascading is implemented automatically by the Quartus® II software based on the configuration file.

Both Stratix III and Cyclone III devices support cascading PLLs. Cyclone II and Cyclone devices do not support this feature.

For more information about post-scale counter cascading and cascading PLLs in Stratix III and Cyclone III device PLLs, refer to the *Clock Networks & PLLs* chapter in the *Stratix III Device Handbook* or the *Cyclone III Device Handbook*.

Common Applications

Use ALTPLL megafunction to implement different PLL configurations. PLLs are used to meet design requirements. PLLs are also used for generating and modifying clock signals, distributing clock signals to different devices in a design, reducing clock skew between devices, and generating internal clock signals.

Stratix III and Cyclone III PLLs are very useful in DDR interfaces because of the use of the reconfigurable PLL to implement the dynamic data path (using the ALTMEMPHY megafunction).

For more information, refer to the *ALTMEMPHY Megafunction User Guide*.

For more information about the functionality of the PLLs in the different devices, refer to the relevant chapters in the *Stratix III, Stratix II, Stratix II GX, Stratix, Stratix GX, Cyclone III, Cyclone II, Cyclone,* and *HardCopy Series Device Handbooks*.

Using the MegaWizard Plug-In Manager

Use the MegaWizard® Plug-In Manager to instantiate the ALTPLL megafunction in your design. Certain ALTPLL megafunction features are only available with Stratix® series phase-locked loops (PLLs). All of these additional features apply to enhanced PLLs, while only some apply to fast PLLs. If you target a fast PLL, the MegaWizard Plug-In Manager does not let you select options available only on enhanced PLLs. Similarly, if you target a Stratix III top and bottom PLL or left and right PLL, the MegaWizard Plug-In Manager does not let you select options available only on the other fast PLLs.

Each family in the Cyclone® series of devices supports only a single kind of PLL. However, the features available vary between families. For example, Cyclone III PLLs (like Stratix III PLLs) support dynamic reconfiguration but Cyclone II PLLs do not.

During compilation, the Quartus® II compiler checks the ALTPLL parameters used against the available PLLs and any PLL or clock input location assignments. If you have not assigned the megafunction to a specific PLL or made a clock input location assignment in the **Assignment Organizer**, the compiler automatically assigns it as an enhanced PLL (or a fast PLL if you turn on the **Use Fast PLL** option).

The compiler issues an error if you specify enhanced PLL features but no enhanced PLLs are available for placement. The compiler also returns an error if an ALTPLL megafunction (specified with enhanced PLL features) is assigned to a fast PLL.

This section provides descriptions for the options available in the ALTPLL MegaWizard Plug-In Manager pages. Table 2–1 through Table 2–5 show which features or settings apply to enhanced and/or fast PLLs. Use these tables along with the hardware descriptions of the fast and enhanced PLL features to determine the appropriate settings for your PLL.

Note that for older designs that do not use the ALTPLL MegaWizard Plug-In Manager, the PLL needs to be regenerated using the MegaWizard Plug-In Manager.

Start the MegaWizard Plug-In Manager using one of the following methods:

- On the Tools menu, click **MegaWizard Plug-In Manager**.
- When working in the Block Editor, from the Edit menu, click **Insert Symbol** as **Block**, or right-click in the **Block Editor**, point to **Insert**, and click **Symbol as Block**. In the **Symbol** dialog box, click **MegaWizard Plug-In Manager**.
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt:

```
qmegawiz↵
```

The ALTPLL Megafunction Page Descriptions (Excluding Stratix III and Cyclone III Devices)

This section describes the options available on the individual pages of the ALTPLL MegaWizard Plug-In Manager. Note that this section is valid for all devices except Stratix III and Cyclone III devices.

Page 2a of the ALTPLL megafunction allows the selection of the I/O category, device selection, type of output file to be created (Verilog HDL, VHDL, or AHDL), and entering the output file name. Note that no option is available to enable clear box netlist generation for this megafunction.

On page 3 of the ALTPLL MegaWizard Plug-In Manager, specify the device to be used, speed grade (the available speeds are affected by the device selection), clock input frequency (either MHz or seconds), the mode of the PLL (fast, enhanced, or automatic), and its operation mode.

Note that the **Set up PLL in LVDS mode** option is only available when the Stratix II, Stratix II GX, or HardCopy® II device is selected. On this page you can also specify the data rate to be used.

On page 3 of the ALTPLL megafunction, from the **Documentation** button, select the **Generate Sample Waveforms** or **Quartus II Megafunction Reference** options to generate a sample simulation waveform. This also launches the Quartus II Help.

Figure 2–1 shows the block diagram of the PLL.

Figure 2–1. PLL Block Diagram

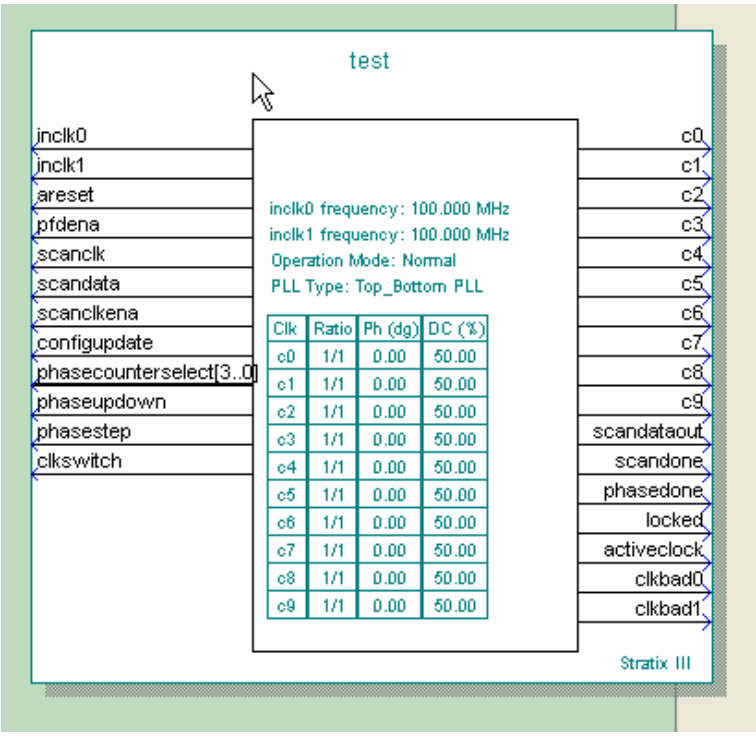


Table 2–1 shows the features and settings on Page 3 of the ALTPLL wizard.

Table 2–1. ALTPLL MegaWizard Plug-In Manager Page 3 Options (1 of 3)

Function	Description	Enhanced PLL	Fast PLL
Which device family will you be using?	Select the Altera® device family you are using.	✓	✓
Which device speed grade will you be using?	Specify the speed grade if you are not already using a device with the fastest speed. The lower the number, the faster the speed grade.	✓	✓
Which PLL type will you be using?	Indicate whether you will have the megafunction use a fast PLL, enhanced PLL, or automatically selected PLL.	✓	✓
What is the frequency of the inclock0 input?	Indicate the input frequency for the `inclock0` input of the PLL.	✓	✓

Table 2–1. ALTPLL MegaWizard Plug-In Manager Page 3 Options (2 of 3)

Function	Description	Enhanced PLL	Fast PLL
Use the feedback path inside the PLL	Indicate which OPERATION_MODE you will use. ■ **Normal mode**—the PLL feedback path comes from either a global or regional clock network, minimizing clock delay to registers for that clock type and specific PLL output. You can specify which PLL output is compensated. ■ **Source-Synchronous mode**—if the data and clock arrive at the same time at the input pins, they are guaranteed to keep the same phase relationship at the clock and data ports of any IOE input register. ■ **Zero Delay Buffer mode**—the PLL feedback path is confined to the dedicated PLL external output pin. The clock port driven off-chip is phase aligned with the clock input for a minimized delay between the clock input and the external clock output. *(1)* ■ **No Compensation mode**—the PLL feedback path is confined to the PLL loop; – it does not come from external source or from clock network. There is no clock network compensation, but this mode minimizes jitter on the clocks. Note that for Source-Synchronous mode and Zero Delay Buffer mode, you need to make assignments (in this case, the PLL_COMPENSATE assignment) in addition to setting the appropriate mode in the megafunction. This allows you to specify an output pin as a compensation target for a PLL in Zero Delay Buffer or External Feedback mode, or an input pin or a group of input pins as compensation targets for a PLL in Source-Synchronous mode. If assigned to an output pin, the pin must be fed by the external clock output port of a PLL in a Stratix, HardCopy Stratix or Cyclone device, or the compensated clock output port of a PLL in other devices. Any other output pins fed by the same PLL generally are not delay compensated, especially if they have different I/O standards. If assigned to an input pin or a group of input pins, the input pins must drive input registers that are clocked by the compensated clock output port of a PLL in Source-Synchronous mode. This option is ignored if it is applied to anything other than an output or input pin as described previously.	✓	✓ *(1)*
Create an 'fbin' input for an external feedback (External Feedback Mode)	**External Feedback mode**—the PLL compensates for the f_{BIN} feedback input into the PLL. The delay between the input clock pin and the feedback clock pin is minimized. Note that Stratix series PLL functional and timing models DO NOT support the simulation of external feedback. The feedback pin is included in the model for connectivity purposes only.	✓	—
Which output clock is to be compensated?	Specify which output port of the PLL is compensated. For Normal mode, select C[5..0]. For Zero Delay Buffer or External Feedback modes, select E[4..0]. *(2)*	✓	—

Table 2–1. ALTPLL MegaWizard Plug-In Manager Page 3 Options (3 of 3)

Function	Description	Enhanced PLL	Fast PLL
Set up PLL in LVDS mode	Indicates whether this mode is used. When checked, this option allows the PLL to supply the necessary clocking signals for the LVDS transmitter and receiver. Note that this option appears only if the selected device is Stratix II, Stratix II GX, or HardCopy II. The PLL type is forced to **Fast**, the operation mode is forced to **Normal Mode**, and two new output ports appear, `sclkout0/1` and `enable0/1`. For more information about this mode, refer to Table 1–1.	—	✓
Data rate	Indicates whether you use this option. This option only appears when Setup PLL in LVDS mode is enabled. The value entered here needs to be used to set the `vco_multiply_by` and `vco_divide_by` parameters, because the VCO frequency corresponds one-to-one with the data rate. For example, if the input frequency is 100M hz, and the data rate is 200 Mbps, then `vco_multiply_by=2`, `vco_divide_by=1`.	—	✓

Notes to Table 2–1:

(1) Fast PLLs do not support the zero delay buffer mode.

(2) Compensated output clocks `e[3..0]` are not applicable to Stratix II and Cyclone II devices (these devices have only `c[]` outputs).

On page 4 of the ALTPLL MegaWizard Plug-In Manager, you can enable dynamic reconfiguration on the enhanced PLLs and set the **LOCK** output options.

Table 2–2 shows the options and settings on page 4 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–2. ALTPLL MegaWizard Plug-In Manager Page 4 Options (1 of 3)

Function	Description	Enhanced PLLs	Fast PLLs
Create optional inputs for dynamic reconfiguration	This option enables all the PLL reconfiguration ports for this instantiation—`scanclk`, `scanaclr`, and `scandata`.	✓	—
Which scan chain type will you be using?	This option lets you specify which PLL to use with PLL reconfiguration. ■ **Long chain**—Specifies that you are using PLLs 5 and 6 with PLL reconfiguration. PLLs 5 and 6 have six logic-array outputs and four external clock outputs and therefore, have a longer reconfiguration chain. ■ **Short chain**—Specifies that you are using PLLs 11 and 12 with PLL reconfiguration. PLLs 11 and 12 have only six logic-array outputs with no dedicated external clock output counters and are considered the shorter reconfiguration chain PLLs.	✓	✓
Create a 'pllena' input to selectively enable the PLL	This option creates a `pllena` port for this PLL instance. Refer to the `pllena` port description in Table 3–1 on page 3–1.	✓	✓
Create an 'areset' input to asynchronously reset the PLL	This option creates an `areset` port for this PLL instance. Refer to the `areset` port description in Table 3–1 on page 3–1.	✓	✓
Create an 'pfdena' input to selectively enable the phase/frequency detector	This option creates a `pfdena` port for this PLL instance. Refer to the `pfdena` port description in Table 3–1 on page 3–1.	✓	✓

Table 2–2. ALTPLL MegaWizard Plug-In Manager Page 4 Options (2 of 3)

Function	Description	Enhanced PLLs	Fast PLLs
Create 'locked' output	This option creates a `locked` output port for indicating PLL lock. Refer to the `locked` port description in Table 3–2 on page 3–3. Note that the number of cycles needed to gate the lock signal is based on the input clock. Gated lock circuitry is clocked by the input clock. The maximum lock time for the PLL is provided in the *DC and Switching Characteristics* chapter of the device handbook. Take the maximum lock time of the PLL and divide that by the period of the input clock. The result is the number of clock cycles needed to gate the lock signal. The lock signal is an asynchronous output of the PLL. The PLL lock signal is derived from the reference clock and feedback (FB) clock feeding the phase frequency detector (PFD). Reference clock = Input Clock/N Feedback clock = VCO/M The PLL generates a locked output when the phases and frequencies of the reference clock and feedback clock are the same or within the lock circuit tolerance. When the difference between the two inputs at the PFD goes beyond the lock circuit tolerance, the PLL loses lock. The lock signal is a function of the PLL input reference clock and the feedback clock, but not exactly synchronous to those clocks because they must be outside of the lock circuit tolerance before the lock signal is deasserted.	✓	✓

Table 2–2. ALTPLL MegaWizard Plug-In Manager Page 4 Options (3 of 3)

Function	Description	Enhanced PLLs	Fast PLLs
Hold 'locked' output low	This option lets you specify the number of cycles that the PLL holds the locked output (up to 10,48,575) after it begins to lock.	✓	✓
Create output file using advanced parameters	 This option is not recommended. This option is intended for users who must know the exact details of their PLL configuration. It is not intended for use in conjunction with the MegaWizard Plug-In Manager, because after the MegaWizard Plug-In Manager specifies the advanced parameters; the compiler cannot change them. Your design cannot benefit from improved algorithms to pick better settings or to make changes to some settings that the MegaWizard Plug-In Manager finds to be incompatible with your design. This option is intended for very advanced PLL users who understand the parameters well and can set them optimally. When this option is turned on, the output file generated from the megafunction contains the entire initial counter values used in the PLL. Use these values during ModelSim® functional simulation, while the PLL parameter calculation is suppressed. Note that this option needs to be used only when the device family, speed grade, and PLL type are specified correctly before performing the simulation. These PLLs do not migrate to other speed grades or families, due to device-family-specific settings. Stratix II devices have different counter sizes, no delay elements, and a different set of loop-filter and charge-pump parameters than Stratix devices. As a result, some parameters available for Stratix devices may not be supported in Stratix II device designs. However, most uses of the PLL do not need to have advanced parameters specified, and most users are not affected by this limitation.	✓	✓

On page 5 of the ALTPLL MegaWizard Plug-In Manager, specify the programmable bandwidth to be used and whether to take advantage of spread spectrum capabilities.

Table 2–3 shows the features and settings on page 5 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–3. ALTPLL MegaWizard Plug-In Manager Page 5 Options

Function	Description	Enhanced PLL	Fast PLL
How would you like to specify the bandwidth?	■ **Auto**—the compiler chooses the bandwidth. ■ **Preset**—select a general **low**, **medium**, or **high** bandwidth for the PLL. Using the low bandwidth option, the PLL has a better jitter rejection but slower lock time. Using the high bandwidth option, bandwidth has a faster lock time but tracks more jitter. Using the medium option offers a balance between the low and high bandwidth options. The compiler tries to minimize, maximize, or set the bandwidth in the middle range according to the other PLL settings. ■ **Custom**—specify a custom bandwidth number. The compiler attempts to achieve the setting that you specify. However, if the compiler cannot achieve these settings, the closest possible value is used. The compiler provides the bandwidth setting in the report file. Use the programmable bandwidth feature only in conjunction with the spread spectrum feature if the bandwidth feature is set to **Auto**.	✓	✓
Use spread spectrum feature and ■ Set down spread to ■ Set modulation frequency to	Enables the spread spectrum feature. You can set the down spread from 0.4 to 0.6%. You can set the modulation frequency from 150 to 500 kHz. Use the spread spectrum feature only in conjunction with the programmable bandwidth feature if the bandwidth feature is set to **Auto**.	✓	—

On page 6 of ALTPLL MegaWizard Plug-In Manager, you specify the options and settings for clock switchover.

Table 2–4 shows the features and settings on page 6 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–4. ALTPLL MegaWizard Plug-In Manager Page 6 Options (1 of 2)

Function	Description	Enhanced PLL	Fast PLL
Create an 'inclock1' input for a second input clock	Adds a second input clock, `inclock1`, to the PLL in addition to the `inclock0` specified on the first page of the MegaWizard Plug-In Manager. The frequency for the second input, `inclock1`, does not have to be the same as the frequency for `inclock0`. You can specify which input (`inclock0` or `inclock1`) is the primary input to the PLL.	✓	✓
Perform input clock switch when the primary clock goes bad	Programs the PLL to switch between input clocks when one clock goes bad.	✓	—

Table 2–4. ALTPLL MegaWizard Plug-In Manager Page 6 Options (2 of 2)

Function	Description	Enhanced PLL	Fast PLL
Create a 'clkswitch' input to dynamically control the switching between input clocks	Creates a control input to manually switch between the input clocks of the PLL. *(1)*	✓	—
Create an 'activeclock' output to indicate the input clock being used	Creates an active clock output port that indicates which input is the current source for the PLL. See the `activeclock` port description in Table 3–2 on page 3–3.	✓	—
Create a 'clkloss' output *(2)*	Creates a `clkloss` output port that indicates when the source input to the PLL has been lost. See the '`clkloss`' port description in Table 3–2 on page 3–3. *(3)*	✓	—
Create a 'clkbad' output for each input clock *(2)*	Creates two `clkbad` outputs, `clkbad1` and `clkbad0`. See the `clkbad` port description in Table 3–2 on page 3–3. *(3)*	✓	—

Notes to Table 2–4:

(1) For more information about performing manual versus automatic clock switchover, refer to the *PLLs in Stratix II Devices* chapter in volume 1 of the *Stratix II Device Handbook*.

(2) This feature is only applicable to Stratix II and Stratix devices.

(3) Stratix II and Cyclone II devices also support manual switchover. For more information, refer to the *PLLs in Stratix II Devices* chapter in volume 1 of the *Stratix II Device Handbook*.

On pages 7 through 16, you can specify the multiplication, division, duty cycle, phase shift, and time shift for each PLL output port (`c0` through `c5` and `e0` through `e3`). Each page represents the settings for one PLL output port.

Table 2–5 shows the features and settings on pages 7–16 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–5. ALTPLL MegaWizard Plug-In Manager Pages 7–16 Options (1 of 2)

Function	Description	Enhanced PLL	Fast PLL
Clock multiplication factor	Specify the clock multiplication for this PLL output.	✓	✓
Clock division factor	Specify the clock division for this PLL output.	✓	✓

Table 2–5. ALTPLL MegaWizard Plug-In Manager Pages 7–16 Options (2 of 2)

Function	Description	Enhanced PLL	Fast PLL
Clock phase shift	This sets the programmable phase shift for the clock output. The equation to determine the precision of the phase shift in degrees is: 45°/(post-scale counter value). The maximum step size is 45. You can set smaller steps using the multiplication and division ratios on the output counter port. For example, if the post-scale counter `g0` is 2, the smallest phase shift step is 22.5°. The MegaWizard Plug-In Manager shows the up and down buttons for the clock phase shift setting on each PLL output. These up and down buttons cycle through the phase shift settings available with the default *m* and post-scale dividers that the MegaWizard Plug-In Manager has chosen for your particular frequency and multiplication. For example, if you enter **125 MHz** with **x1**, it shows 15° increments on the phase shift when hitting the down buttons (15, 30, 45, and so on). To get other granularities of shift, enter a number into the phase shift field manually instead of using the buttons. In this example, if you enter **7.5×**, the MegaWizard Plug-In Manager will verifies this and uses *m* = 6, *g*0 = 6. If you enter 10 and the MegaWizard Plug-In Manager validates that 9° is possible by using *m* = 5, *g*0 = 5.	✓	✓
Clock duty cycle	Specifies the clock duty cycle on the clock output. Use the up and down buttons to cycle through all possible settings.	✓	✓
Enter output clock frequency	Specifies the desired output frequency. The Quartus II software determines the appropriate multiplication and division factor.	✓	✓
Create sclkout0/enable0	This option toggles the PLL to keep or not keep two new output ports, `sclkout0/1` and `enable0/1`. Note that this option is valid when the PLL is in LVDS mode.	—	✓
Enable sclkout phase shift edit	This option only appears when the **Create sclkout0/enable0** option is enabled. When enabled, this option allows you to specify the phase shift of the given `sclkout` output. Note that this option is valid when the PLL is in LVDS mode.	—	✓
sclkout phase shift	This option only appears when the **Enable sclkout** phase shift edit option is enabled. Here you can manually enter the phase shift in degrees, ns, or ps. Note that this option is valid when the PLL is in LVDS mode.	—	✓

Fast PLLs support up to three internal outputs. Enhanced PLLs 5 and 6 support six internal outputs (`c0` through `c5`).

The following information is Stratix series device specific. Enhanced PLLs 11 and 12 are short-chain PLLs without external output clock counters. However, PLLs 11 and 12 support all six internal outputs (`c0` through `c5`) and one external output driven from one of the `c0` counters. To ensure the Quartus II software uses the dedicated `PLL11_OUT` or `PLL12_OUT` pin from the `c0` output on PLLs 11 and 12, follow these steps:

1. On Page 7 of the ALTPLL MegaWizard Plig-In Manager, configure the clock `c0` output to the desired clock settings. This same setting also applies to the one external output available on either PLL 11 or 12.

2. Turn on the **Mirror these settings on external clock e0** option. An `e0` output appears on the PLL instance that mirrors the `c0` settings.
3. Connect `e0` to the output pin in the design.

When complete, the Quartus II software ensures that the `e0` output drives the `PLL11_OUT` or `PLL12_OUT` output pins.

The `PLL11_OUT` or `PLL12_OUT` pins are only applicable to Stratix GX and Stratix devices.

The PLLs are implemented in dedicated circuitry in Stratix series devices.

ALTPLL Megafunction Page Descriptions (Stratix III and Cyclone III Devices Only)

This section describes the options available on the individual pages of the ALTPLL MegaWizard Plug-In Manager. This section is valid for Stratix III and Cyclone III devices only.

Page 2a of the megafunction allows you to select the ALTPLL megafunction from the I/O category and to select the device, type of output file to be created (Verilog HDL, VHDL, or AHDL), and output file name. Note that there is no option available to enable clear box netlist generation for this megafunction.

On page 3 of the ALTPLL MegaWizard Plug-In Manager, you can specify the device family, speed grade (the available speeds are affected by the device selection), clock input frequency (either MHz or fractions of seconds), mode of the PLL (**left_right**, **top_bottom**, or **automatic**), and its operation mode.

Note that the **Set up PLL in LVDS mode** option is not available because in Stratix III devices, the outputs of the PLL are connected directly to the serializer/deserializer (SERDES), and in Cyclone III, the SERDES circuitry is implemented in the logic elements (LEs). Therefore, there is no need to create a special mode with extra LVDS-specific ports and the data rate is not available.

On page 3 of the ALTPLL MegaWizard Plug-In Manager, from the **Documentation** button in the upper right, you can access this user guide as well as related documentation, launch the Quartus II Help system, or generate a sample waveform.

Table 2–6 shows the features and settings on page 3 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–6. ALTPLL MegaWizard Plug-In Manager Page 3 Options (1 of 4)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Which device family will you be using?	Select the Altera device family you are using.	Stratix III	Stratix III	Cyclone III
Which device speed grade will you be using?	Specify the speed grade if you are not already using a device with the fastest speed. The lower the number, the faster the speed grade.	Refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Device Handbook*.	Refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Device Handbook*.	Refer to the *DC & Switching Characteristics of Cyclone III Devices* chapter in the *Cyclone III Device Handbook*.
What is the frequency of the inclock0 input?	Indicate the input frequency for the `inclock0` input of the PLL.	For the input frequency range, refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Device Handbook*.	For the input frequency range, refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Device Handbook*.	For the input frequency range, refer to the *DC & Switching Characteristics of Cyclone III Devices* chapter in the *Cyclone III Devices Handbook*.

Table 2–6. ALTPLL MegaWizard Plug-In Manager Page 3 Options (2 of 4)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Set up PLL in LVDS mode	This option is not available in Stratix III and Cyclone III devices. For clocking LVDS interfaces, the PLL is just configured as a regular PLL in Stratix III and Cyclone III devices. To create ALTPLL: ■ Select left/right PLL type ■ Select **Source-Synchronou**s compensation Clk0: fast-clk (to clock the port on ALTLVDS) ■ Output frequency: datarate ■ Phase shift: -180° ■ Duty cycle: 50% Clk1: load-ena (to enable the port on ALTLVDS) ■ Output frequency: datarate/deser-factor ■ Phase shift: [(deserialization-factor – 2/deserialization-factor] × 360 degrees ■ Duty cycle: (100/deserialization-factor)% Clk2: slow/core-clk (to core logic/syn register) ■ Output frequency: datarate/deserialization-factor ■ Phase shift: (-180/deserialization-factor) degrees ■ Duty Cycle: 50% Set the following in the generated wrapper file if you use DPA: ■ dpa_multiply_by and dpa_divide_by = same mult/div as Clk0 (for example, DPA clk frequency is the same as datarate)	—	—	—
Which PLL type will you be using?	For Stratix III devices, indicate whether you use a top and bottom PLL, left and right PLL, or an automatically selected PLL. For Cyclone III devices, the only available option is automatically selected PLL.	The **Top/Bottom PLL** option must be selected.	The **Left/Right PLL** option must be selected.	—

Table 2–6. ALTPLL MegaWizard Plug-In Manager Page 3 Options (3 of 4)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Use the feedback path inside the PLL	Specify which OPERATION_MODE to use: ■ **Normal mode**—the PLL feedback path comes from either a global or regional clock network, minimizing clock delay to the registers for that clock type and specific PLL output. You can specify which PLL output is compensated. ■ **Source-Synchronous mode**—if the data and clock arrive at the same time at the input pins, they are guaranteed to keep the same phase relationship at the clock and data ports of any IOE input register. ■ **Zero Delay Buffer mode**—the PLL feedback path is confined to the dedicated PLL external output pin. The clock port driven off-chip is phase aligned with the clock input for a minimized delay between the clock input and external clock output. ■ **No Compensation mode**—the PLL feedback path is confined to the PLL loop. It does not come from external source or from the clock network. There is no clock network compensation, but this mode minimizes jitter on clocks. Note that for Source-Synchronous mode and Zero Delay Buffer mode, you need to make assignments (in this case, the PLL_COMPENSATE assignment) in addition to setting the appropriate mode in the megafunction. This allows you to specify an output pin as a compensation target for a PLL in Zero Delay Buffer or External Feedback mode, or an input pin or a group of input pins as compensation targets for a PLL in Source-Synchronous mode. If assigned to an output pin, the pin must be fed by the external clock output port of a PLL in a Stratix, HardCopy Stratix, or Cyclone device, or the compensated clock output port of a PLL in other devices. Any other output pins fed by the same PLL generally are not delay compensated, especially if they have different I/O standards. If assigned to an input pin or a group of input pins, the input pins must drive input registers that are clocked by the compensated clock output port of a PLL in Source-Synchronous mode.	All of the compensation modes specified here are available for this type of PLL.	All of the compensation modes specified here are available for this type of PLL.	All of the compensation modes specified here are available for this type of PLL.

Table 2–6. ALTPLL MegaWizard Plug-In Manager Page 3 Options (4 of 4)

Function	Description	Top and Bottom	Left and Right	Cyclone III
	This option is ignored if it is applied to anything other than an output or input pin as described previously.			
Create an 'fbin' input for an external feedback (External Feedback Mode)	**External Feedback mode**—the PLL compensates for the f_{BIN} feedback input into the PLL. The delay between the input clock pin and the feedback clock pin is minimized. Note that Stratix series PLL functional and timing models DO NOT support the simulation of external feedback. The feedback pin is included in the model for connectivity purposes only.	This option is available for this type of PLL.	This option is available for this type of PLL for single-ended I/O standards only.	—
Which output clock will be compensated for?	Specify which output port of the PLL is compensated.	For Normal, Source Synchronous and Zero Delay Buffer mode. Other modes are not compensated.	For Normal mode only. Other modes are not compensated.	For Normal, Source Synchronous and Zero Delay Buffer mode. Other modes are not compensated.

On page 4 of the ALTPLL MegaWizard Plug-In Manager, adjustable settings allow you to enable dynamic reconfiguration and dynamic phase reconfiguration, to create optional inputs for asynchronous reset and phase/frequency detector, and to activate the **locked** output options and advanced PLL parameters.

Table 2–7 shows the options and settings on page 4 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–7. ALTPLL MegaWizard Plug-In Manager Page 4 Options (1 of 3)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Create optional inputs for dynamic reconfiguration	This option enables all the PLL reconfiguration ports for this instantiation: input ports (`scanclk`, `scandata`, `scanclkena` and `configupdate`) and output ports (`scandataout` and `scandone`).	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Create optional inputs for dynamic phase reconfiguration	This option enables all the PLL phase reconfiguration ports for this instantiation: input ports (`phasecounterselect[3..0]`, `phaseupdown`, `phasestep` and `scanclk`) and output ports (`phasedone`).	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Create an 'pllena' input to selectively enable the PLL	This option creates a `pllena` port for this PLL instance. See the `pllena` port description in Table 3–1 on page 3–1.	Not available for Stratix III devices.	Not available for Stratix III devices.	Available for this type of PLL.

Table 2–7. ALTPLL MegaWizard Plug-In Manager Page 4 Options (2 of 3)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Create an 'areset' input to asynchronously reset the PLL	This option creates an `areset` port for this PLL instance. See the `areset` port description in Table 3–1 on page 3–1.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Create an 'pfdena' input to selectively enable the phase/frequency detector	This option creates a `pfdena` port for this PLL instance. See the `pfdena` port description in Table 3–1 on page 3–1.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Create output file or files using 'Advanced' PLL parameters	CAUTION This option is not recommended. This option is intended for users who must know the exact details of their PLL configuration. It is not intended for use in conjunction with the MegaWizard Plug-In Manager, because after the MegaWizard Plug-In Manager specifies the advanced parameters; the compiler cannot change them. Your design cannot benefit from improved algorithms to pick better settings or to make changes to some settings that the MegaWizard Plug-In Manager finds to be incompatible with your design. This option is intended for very advanced PLL users who understand the parameters well and can set them optimally. When this option is turned on, the output file generated from the megafunction contains the entire initial counter values used in the PLL. Use these values during ModelSim functional simulation, while the PLL parameter calculation is suppressed. Note that this option needs to be used only when the device family, speed grade, and PLL type are specified correctly before performing the simulation. These PLLs do not migrate to other speed grades or families, due to device-family-specific settings. Stratix III devices have different counter sizes, no delay elements, and a different set of loop-filter and charge-pump parameters than Stratix devices. As a result, some parameters available for Stratix devices may not be supported in Stratix III device designs. However, most uses of the PLL do not need to have advanced parameters specified, and most users are not affected by this limitation.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.

Table 2–7. ALTPLL MegaWizard Plug-In Manager Page 4 Options (3 of 3)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Create 'locked' output	This option creates a `locked` port for this PLL instance. See the `locked` port description in Table 3–2 on page 3–3. Note that the number of cycles needed to gate the lock signal is based on the input clock. Gated lock circuitry is clocked by the input clock. The maximum lock time for the PLL is provided in the *DC and Switching Characteristics* chapter of the device handbook. Take the maximum lock time of the PLL and divide that by the period of the input clock. The result is the number of clock cycles needed to gate the lock signal. The lock signal is an asynchronous output of the PLL. The PLL lock signal is derived from the reference clock and feedback clock feeding the phase frequency detector. Reference clock = Input Clock/N Feedback clock = VCO/M The PLL generates a locked output when the phases and frequencies of the reference clock and feedback (FB) clock are the same or within the lock circuit tolerance. When the difference between the two inputs at the PFD goes beyond the lock circuit tolerance, the PLL loses lock. The lock signal is a function of the PLL input reference clock and the feedback clock, but are not exactly synchronous to those clocks because they must be outside of the lock circuit tolerance before the lock signal is deasserted.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Enable self-reset on loss of lock	This option enables the lock counter.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.

On page 5 of the ALTPLL MegaWizard Plug-In Manager, for Stratix III devices, specify the programmable bandwidth to be used. Customizing spread-spectrum capabilities is not available for Stratix III devices. This MegaWizard Plug-In Manager page does not appear for Cyclone III devices, which have no bandwidth options and no spread-spectrum options available.

Table 2–8 shows the features and settings on page 5 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–8. ALTPLL MegaWizard Plug-In Manager Page 5 Options for Stratix III Devices

Function	Description	Top and Bottom	Left and Right
How would you like to specify the bandwidth setting?	■ **Auto**—The compiler chooses the bandwidth. ■ **Preset**— Values are **low**, **medium**, or **high**. ■ **Low**—the PLL will have a better jitter rejection but slower lock time. ■ **Medium**—is a balance between both low and high; the compiler tries to minimize, maximize, or set the bandwidth in the middle range according to the other PLL settings. ■ **High**—has a faster lock time but tracks more jitter.	Available for this type of PLL.	Available for this type of PLL.

On page 6 of the ALTPLL MegaWizard Plug-In Manager, you can specify the options and settings for clock switchover. If the specified device is a Cyclone III device, this page is labelled page 5.

Table 2–9 shows the features and settings on page 6 of the ALTPLL MegaWizard Plug-In Manager.

Table 2–9. ALTPLL MegaWizard Plug-In Manager Page 6 Options (1 of 2)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Create an 'inclock1' input for a second input clock	Adds a second input clock, `inclock1`, to the PLL in addition to the `inclock0` specified on the first page of the MegaWizard Plug-In Manager. The frequency for the second input, `inclock1`, does not have to be the same as the frequency for `inclock0`. Note that the status signals used are valid only if the input clock frequencies are within two times of each other.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Input clock switch	There are two options to customize the input clock switch: ■ **Create a 'clkswitch' input to manually select between the input clocks**—use for manual switchover ■ **Allow PLL to automatically control the switching between input clocks**—enables automatic switchover; you can also create a `clkswitch` input for manual override	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.

Table 2–9. ALTPLL MegaWizard Plug-In Manager Page 6 Options (2 of 2)

Function	Description	Top and Bottom	Left and Right	Cyclone III
Create an 'activeclock' output to indicate the input clock being used	Creates an `activeclock` output port that indicates which input is the current source for the PLL. See the `activeclock` port description in Table 3–2 on page 3–3.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.
Create a 'clkbad' output for each input clock	Creates two `clkbad` outputs, `clkbad0` and `clkbad1`. See the `clkbad` port description in Table 3–2 on page 3–3.	Available for this type of PLL.	Available for this type of PLL.	Available for this type of PLL.

On pages 7 to 16 of the MegaWizard Plug-In Manager for Stratix III devices, and pages 6 to 10 of the MegaWizard Plug-In Manager for Cyclone III devices, specify the multiplication, division, duty cycle, phase shift, and time shift for each PLL output port. Available output ports are `c0` through `c9` for Stratix III devices and `c0` through `c4` for Cyclone III devices. Each page represents the settings for one PLL output port.

Table 2–10 shows the features and settings on pages 7 through 16 (or pages 6 through 10) of the ALTPLL MegaWizard Plug-In Manager.

Table 2–10. ALTPLL MegaWizard Plug-In Manager Page 7 through 18 Options

Function	Description	Top and Bottom	Left and Right	Cyclone III
Enter output clock frequency	Use Table 2–5 on page 2–10 as reference.	Has 10 available output clocks where this option can be set.	Has seven available output clocks where this option can be set.	Has five available output clocks where this option can be set.
Clock multiplication/ division factor	Use Table 2–5 on page 2–10 as reference.	Has 10 available output clocks where this option can be set.	Has seven available output clocks where this option can be set.	Has five available output clocks where this option can be set.
Clock phase shift	Use Table 2–5 on page 2–10 as reference.	Has 10 available output clocks where this option can be set.	Has seven available output clocks where this option can be set.	Has five available output clocks where this option can be set.
Clock duty cycle	Use Table 2–5 on page 2–10 as reference.	Has 10 available output clocks where this option can be set.	Has seven available output clocks where this option can be set.	Has five available output clocks where this option can be set.

Timing Analysis

The register-to-register timing for each PLL clock output that drives the logic array is reported with slack. In the Timing Analysis section of the report, you can see the actual point-to-point delay, the required setup relationship, and a list of the most critical paths for each clock. For each path, both the slack and f_{MAX} are provided. Perform a List Path to view the various timing parameters (for example, microparameters, t_{CO} and t_{SU}).

During timing analysis for designs using PLLs, the project clock settings override the PLL input clock frequency and duty cycle settings.

Note the following requirements and conditions:

- A warning issued 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 the project clock setting to 170 MHz so the Fitter attempts 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 are not overridden.
- Overriding the PLL clock settings changes only the timing requirements; it does not change the overall multiplication and 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 megafunction parameters.
- A Default Required f_{MAX} setting does not override the PLL clock settings. Only individual clock settings override the PLL clock settings.

Overriding PLL clock settings is useful when you have configured a 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. This feature therefore allows you to overwrite the PLL input clock frequency settings for timing analysis, which means that you do not have to resynthesize or refit your design. The following procedure overrides the PLL input frequency setting and regenerates timing analysis.

1. On the Assignments menu, click **Timing Analysis Settings**.
2. Under **Timing Analysis Settings**, expand **Classic Timing Analyzer Settings** and click **Individual Clocks**.
3. In the **Individual Clocks** dialog box, click **New...**
4. In the **New Clock Settings** dialog box, type a name for the new clock settings.

5. If you want to specify timing requirements for an absolute clock, follow these steps:
   a. Under **Relationship to other clock** settings, select the **Independent of other clock** settings.
   b. In the **Required f_{MAX}** box, type the required frequency of the clock signal and select a time unit from the list.
   c. In the **Duty Cycle** box, enter the required duty cycle for the clock.


Cyclone PLLs accept input clocks with duty cycles between 40 and 60%.

   d. Click **OK**.
6. Click **OK** to close the **Settings** dialog box.
7. On the Assignments menu, click **Assignment Editor**.
8. In the spreadsheet, double-click an empty cell in the **Assignment Name** column, and scroll to and select **Clock Settings**.
9. Double-click an empty row in the **To** column, click on the arrow, and click **Node Finder** to search for the external feedback input pin.
10. In the **Node Finder** dialog box, click the **List** button and locate the name of the input PLL.
11. Click **OK**.
12. In the **Assignment Editor** spreadsheet, double-click the **Value** cell in the same row as the clock setting that you created in an earlier step.
13. On the Processing menu, point to Start and click **Start Classic Timing Analyzer**.

Simulation

The Quartus II Simulation tool provides an easy-to-use, integrated solution for performing simulations. The following sections describe the simulation options.

The ALTPLL megafunction supports behavioral and timing simulation. Simulation supports all control signals and clock outputs. Table 2–11 shows the simulation support for the ALTPLL megafunction.

Table 2–11. ALTPLL Simulation Support (1 of 2)

Feature	Simulation Support
Lock	Modeled for a high bandwidth condition only. The PLL will lock or relock in 2 to 10 cycles in simulation. This does not necessarily reflect the real lock time, which can take thousands of cycles for low bandwidth settings.
Programmable bandwidth	Not modeled.
PLL reconfiguration	Can simulate on-the-fly changes of PLL parameters. Any relock upon changing *m* or *n* is modeled for high bandwidth only just as in the lock feature.
External feedback	Modeled *(1)*.
PFD enable	Modeled. The finite frequency drift of VCO is not modeled if phase frequency detector (PFD) is disabled.

Table 2–11. ALTPLL Simulation Support (2 of 2)

Feature	Simulation Support
Clock switchover	Manual and automatic switch and control signals modeled. Frequency drift on lost clock and frequency overshoot (relock on secondary or switched clock) is not modeled.
Frequency input change	If the input frequency of the PLL is changed in simulation, the model checks that (f_{IN} x m)/n is within the VCO range and locks as if configured for high bandwidth.
Spread Spectrum	Frequency modulation is not modeled in simulation.
Jitter	Jitter is not modeled in simulation. *(2)*
`pllena`	Modeled. When this signal is driven low, the PLL loses lock and the PLL clock outputs are driven to logic low.
`areset`	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.

Note to Table 2–11:

(1) For more information about external feedback simulation, refer to "Simulating External Feedback Board Delay in Stratix II and Stratix II GX Devices".

(2) Jitter is not modeled in the Quartus II Simulator, but both TimeQuest and the Classic Timing Analyzer allow you to specify the expected clock setup or hold uncertainty (associated with jitter) when you perform setup and hold checks for clocks or clock-to-clock transfers. For more information, refer to the *Timing Analysis* section in the *Quartus II Handbook* .

Simulating External Feedback Board Delay in Stratix II and Stratix II GX Devices

This option is available for Stratix II and Stratix II GX devices only. The functional and timing models of these devices do not support the simulation of external feedback. Set the **PLL External Feedback Board Delay** option on the external feedback input pin (`fbin`) to simulate External Feedback mode by performing the following steps:

1. In the Quartus II software, open an existing project or create a new project.
2. On the Assignments menu, click **Assignment Editor**.
3. In the **Category** bar, under Timing, click **Other Timing**.
4. In the spreadsheet, double-click an empty row in the **To** cell and either type in the pin name or click on the arrow to use the **Node Finder** to search for the external feedback input pin.
5. Double-click the **Assignment Name** cell, and select **PLL External Feedback Board Delay**.
6. In the **Value** cell, double-click and type the amount of time for the signal to propagate between the external clock output pin through the trace on the board and into the external feedback input pin.

Use the ALTPLL behavioral model to simulate the Stratix II and Stratix II GX enhanced and fast PLLs. The Stratix II and Stratix II GX devices' behavioral model instantiation needs to follow the same guidelines and restrictions as the design entry. ALTPLL behavioral and timing models do not simulate jitter.

The behavioral models for the ALTPLL megafunction reside in the **\quartus\eda\sim_lib** directory. The **altera_mf.vhd** file contains the VHDL behavioral models and can be used for the Stratix II/Stratix II GX ALTPLL megafunction. The **altera_mf.v** file contains the Verilog HDL behavioral models and can be used for Stratix II ALTPLL behavioral simulation. The behavioral model does not perform parameter error checking. You must specify only valid values.

You must set the resolution of the VHDL simulator to picoseconds (ps) to simulate the model successfully. A larger resolution rounds off the calculations, providing incorrect multiplication or division.

Calculate the Clock Cycles to Gate the Lock Signal

Occasionally, you must calculate the number of cycles needed to gate the lock signal.

Gated lock circuitry is clocked by the input clock. The maximum lock time for the PLL is provided in the appropriate chapter of the device handbook that is used in the design. You must take the maximum lock time of the PLL and divide it by the period of the input clock. The result is the number of clock cycles needed to gate the lock signal.

Design Examples

This section presents two design examples that use the ALTPLL megafunction to generate an external differential clock from an enhanced PLL (as shown in Figure 2–2 on page 2–26) and generate and modify internal clock signals (as shown in Figure 2–4 on page 2–31).

These examples use the MegaWizard Plug-In Manager in the Quartus II software. Each page of the MegaWizard is described in detail. When you are finished with the examples, you can incorporate them into your overall projects.

Design Files

The design files are available on the Literature page of the Altera website (www.altera.com). The files are located under the following sections:

- On the Quartus II Literature page, expand the **Using Megafunctions** section and then expand the **I/O** section
- User Guides section

Example 1: Differential Clock

This section presents a design example that uses the ALTPLL megafunction to generate an external differential clock from an enhanced PLL. It is often necessary to generate or modify clock signals to meet design specifications. When you interface to double data rate (DDR) memory, you must generate a differential SSTL clock signal for the external device. A DDR DIMM requires three pairs of differential SSTL clocks. You can use enhanced PLLs in Stratix devices to generate these clock signals.

In this example, perform the following activities:

- Generate a 166-MHz differential SSTL external clock (`ddr_clk`) output from a 33.33-MHz input clock using the ALTPLL megafunction and the MegaWizard Plug-In Manager.
- Implement the `DDR_CLK` design by assigning the EP1S10F780 device to the project and compiling the project.
- Simulate the `DDR_CLK` design.

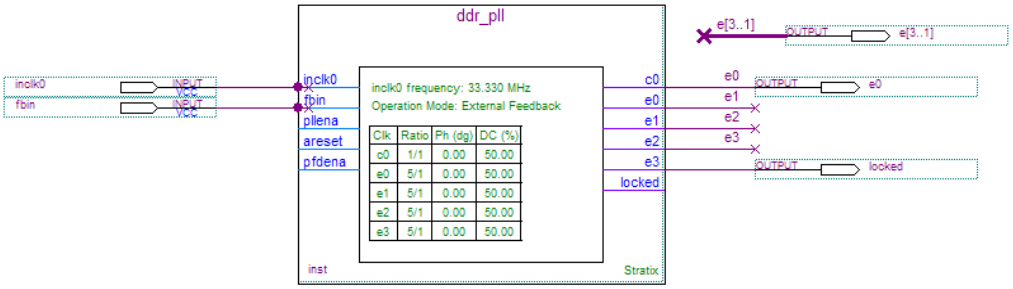
Generate a 166-MHz Differential SSTL External Clock

1. In the Quartus II software, open the project file **\ddr_clk\ddr_clk.qpf**.
2. Open the top-level **\ddr_clk\ddr_clk.bdf** file. You will complete this project in this example.
3. Double-click on a blank area in the block design (**.bdf)** file, and click **MegaWizard Plug-In Manager** in the Symbol window, or, on the Tools menu, click **MegaWizard Plug-In Manager**.
4. Under **What action do you want to perform?**, click **Create a new custom megafunction**. Click **Next**. Page 2a appears.
5. On Page 2a of the MegaWizard Plug-In Manager, expand the **I/O** folder and select **ALTPLL**.
6. For **Which device family will you be using?**, select **Stratix**.
7. Under **Which type of output file do you want to create?**, select **AHDL**.
8. For **What name do you want for the output file?**, name the output file **ddr_pll**.
9. Click **Next**. Page 3 appears.
10. On Page 3, in the **General** section, for **What is the frequency of the inclock0 input?**, type `33.33`, and select **MHz**.
11. Under **PLL type**, click **Select the PLL type automatically**.
12. Under **Operation mode**, select **Create an 'fbin' input for an external feedback (External Feedback Mode)**.
13. Under **Operation mode**, for **Which output clock will have a board-level connection?**, select **e0** from the drop-down menu.
14. Click **Next**. Page 4 appears.
15. In the **Dynamic configuration** section, leave the default settings.
16. In the **Optional inputs** section:
    a. Turn on **Create an 'pllena' input to selectively enable the PLL**.
    b. Turn on **Create an 'areset' input to asynchronously reset the PLL.**
    c. Turn off **Create an 'pfdena' input to selectively enable the phase/frequency detector**.
17. In the **Lock output** section, turn on **Create 'locked' output**.
18. Leave the remaining options with the default settings.
19. Click the **Output Clocks** tab. Page 7 appears.
20. On page 7, click **extclk e0**. Page 13 appears.
21. Turn on **Use this clock**.
22. Under **Enter output clock parameters**, in the **Clock multiplication factor** box, type `5`.
23. In the **Clock division factor** box, type `1`.
24. In the **Clock duty cycle (%)** box, type `50.00`.

25. Click **Next**. Page 14 appears.
26. On page 14, repeat steps 21 through 24 for **extclk e1**.
27. Click **Next**.
28. On Page 15, repeat steps 21 through 24 for **extclk e2**.
29. Click **Next**.
30. On Page 16, repeat steps 21 through 24 for **extclk e3**.
31. Click **Next**. Page 17 appears. No input is required for this page.
32. Click **Next**. Page 18 appears.
33. On page 18, ensure that the Variation file (**.tdf**), PinPlanner ports PPF file (**.ppf**), AHDL Include file (**.inc**), Quartus II symbol file (**.bsf**), and Sample waveforms in summary file (**.html** and **.jpg**) are turned on.
34. Click **Finish**. The **ddr_pll** module is built.
35. In the **Symbol** dialog box of the **.bdf** file, click **OK**.
36. Move the pointer to place the **ddr_pll** symbol between the input and output ports in the **ddr_clk.bdf** file, connecting the inputs and outputs to the symbol. Click to place the symbol.

    You have now completed the design file shown in Figure 2–2.

Figure 2–2. ALTPLL ddr_pll Design Schematic



37. On the File menu, click **Save Project** to save the design.

Implement the ddr_clk Design

In this step you will assign the EP1S10F780 device to the project and compile the project.

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click **Device**. Ensure that **Stratix** is selected in the Family field.
3. In the **Target device** section, under **Available devices**, select **EP1S10F780C5**.
4. Click **OK**.
5. On the Processing menu, click **Start Compilation**.
6. When the **Full Compilation was successful** message box appears, click **OK**.

7. To view how the module is implemented in the Stratix device, on the Assignments menu, click **Timing Closure Floorplan**.

The `ddr_clk` design is now implemented.

Functional Results—Simulate the ddr_clk Design in the ModelSim-Altera Software

In this section you will simulate the design in the ModelSim-Altera software to compare the results of both simulators. Note that this ModelSim design example is for the ModelSim-Altera (Verilog) version.

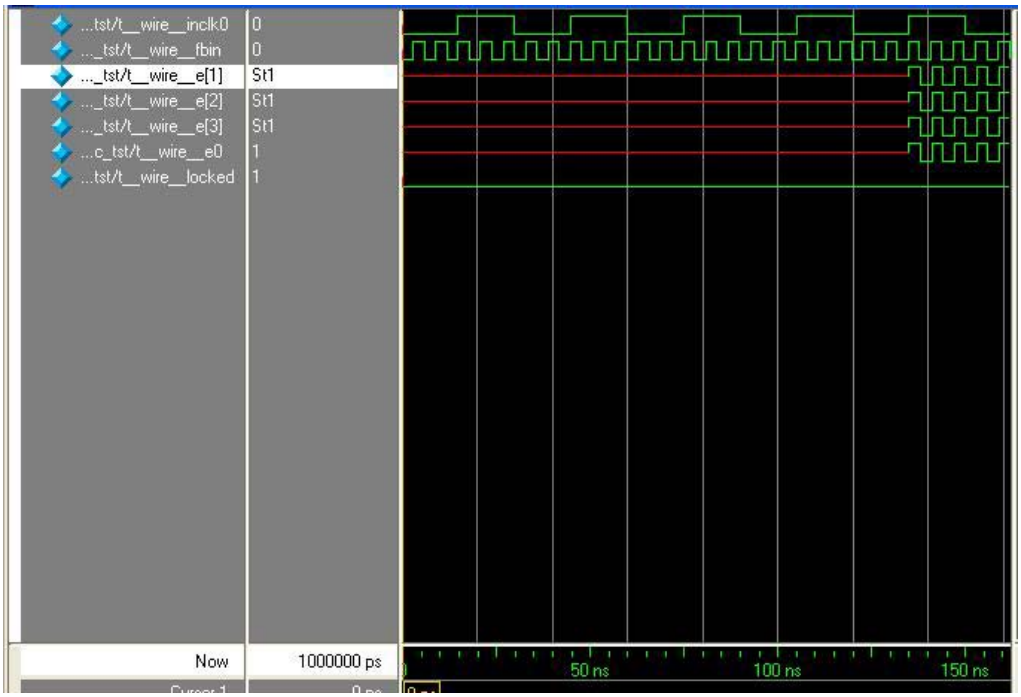
You need to be familiar with the ModelSim-Altera software before trying out the design example. If you are unfamiliar with the ModelSim-Altera software, refer to the ModelSim support page on the Altera website (www.altera.com). On this support page, there are links to such topics as installation, usage, and troubleshooting.

Set up the ModelSim-Altera Simulator by performing the following steps:

1. Unzip **DDR_CLK_msim.zip** to any working directory on your PC.
2. Locate the folder in which you unzipped files and open the **DDR_CLK.do** file in a text editor.
3. In line 1, replace <*insert_directory_path_here*> with the directory path of the appropriate library files. For example, `C:/Modeltech_ae/altera/verilog/stratix`
4. On the File menu, click **Save**.
5. Start the ModelSim-Altera software.
6. On the File menu, click **Change Directory**.
7. Select the folder in which you unzipped the files. Click **OK**.
8. On the Tools menu, click **Execute Macro**.
9. Select the **DDR_CLK.do** file and click **Open**. The **DDR_CLK.do** file is a script file for the ModelSim-Altera software to automate all the necessary settings for the simulation.

10. Verify the results shown in the Waveform Viewer window. You may need to rearrange signals, remove redundant signals, and change the radix to suit the results in the Quartus II Simulator. Figure 2–3 shows the expected simulation results in the ModelSim-Altera software.

Figure 2–3. ModelSim Simulation Results



Example 2: Generating Clock Signals

This section presents a design example that uses the ALTPLL megafunction to generate and modify internal clock signals. This example generates three internal clock signals from an external 100 MHz clock signal.

In this example, perform the following activities:

- Generate 133 MHz, 200 MHz, and 200 MHz clocks that are time shifted by 1.00 ns from a 100 MHz external input clock using the ALTPLL megafunction and the MegaWizard Plug-In Manager.
- Implement the `shift_clk` design by assigning the EP1S10F780 device to the project and compiling the project.
- Simulate the `shift_clk` design.

Generate 133 MHz, 200 MHz, and 200 MHz Time-Shifted Clocks

1. In the Quartus II software, open the project file **shift_clk.qpf**.
2. Open the top-level **shift_clk.qpf** file. You will complete this project in this example.

3. Double-click on a blank area in the block design (**.bdf)** file, and click **MegaWizard Plug-In Manager** in the **Symbol** dialog box, or, on the Tools menu, click **MegaWizard Plug-In Manager**. Page 1 of the MegaWizard Plug-In Manager appears.
4. On Page 1 of the MegaWizard Plug-In Manager, in the **What action do you want to perform?** section, click **Create a new custom megafunction variation** and click **Next**. Page 2a appears.
5. On Page 2a of the MegaWizard Plug-In Manager, expand the **I/O** folder and click **ALTPLL**.
6. In **Which type of output file do you want to create?**, make sure the **AHDL** option is selected.
7. Name the output file **shift_pll**.
8. Click **Next**. Page 3 appears.

In the following steps, specify the 100 MHz external input clock.

1. In the **General** section, for **What is the frequency of the inclock0 input?** type `100` and select **MHz**. Leave the other options as default.
2. In the **PLL type** section, for **Which PLL type will you be using?**, click **Select the PLL type automatically**.
3. In the **Operation mode** section, make sure the **Use the feedback inside the PLL** and **In Normal Mode** options are turned on.
4. For **Which output clock will be compensated for?**, select **c0**.
5. Click **Next**. Page 4 appears.
6. In the **Dynamic configuration** section, make sure **Create optional inputs for dynamic reconfiguration** is turned off.
7. In the **Optional inputs** section:
   a. Turn on **Create an 'pllena' input to selectively enable the PLL**.
   b. Turn on **Create an 'areset' input to asynchronously reset the PLL.**
   c. Turn off **Create an 'pfdena' input to selectively enable the phase/frequency detector**.
8. In the **Lock output** section, turn on **Create 'locked' output**.
9. Leave the **Advanced PLL** parameters as the default.

In the following steps, specify the 133 MHz internal clock (`c0`).

1. Click the **Output Clocks** tab to access configuration for all of the output clocks in the PLL. Page 7 appears.
2. Under **Core Output Clock**, turn on **Use this clock**.

3. In the **Clock Tap Settings** section:
   a. Turn off **Enter output clock frequency:**.
   b. Turn on **Enter output clock parameters:**.
   c. For **Clock multiplication factor**, type `4`.
   d. For **Clock division factor**, type `3`.
   e. For **Clock phase shift**, type `0` and select **deg**.
   f. For **Clock duty cycle (%)**, type `50.00`.
4. Leave the other options as the default.
5. Click **Next**. Page 8 appears.

In the following steps, specify the 200 MHz internal clock (`c1`).

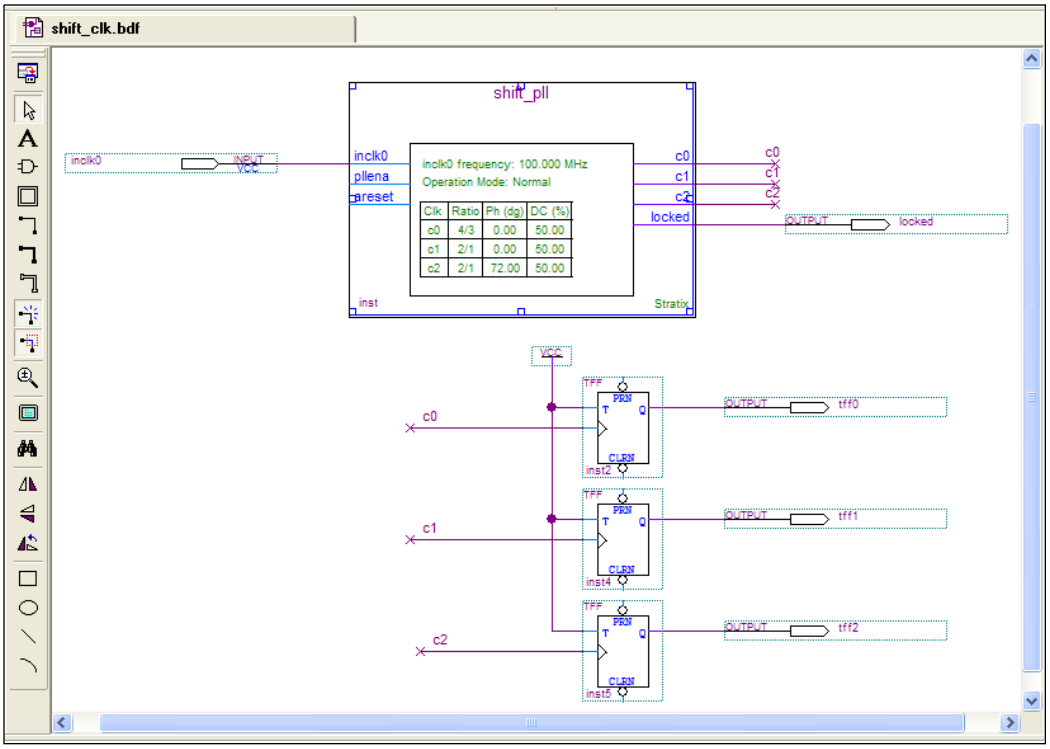
1. Under **Core Output Clock**, turn on **Use this clock**.
2. In the **Clock Tap Settings** section:
   a. Turn off **Enter output clock frequency:**.
   b. Turn on **Enter output clock parameters:**.
   c. For **Clock multiplication factor**, type `2`.
   d. For **Clock division factor**, type `1`.
   e. For **Clock phase shift**, type `0.00` and select **ns**.
   f. For **Clock duty cycle (%)**, type `50.00`.
3. Leave the other options as the default.
4. Click **Next**. Page 9 appears.

In the following steps, specify the 200 MHz internal clock (`c2`) with a 1.00 nanosecond delay.

1. Under **Core Output Clock**, turn on **Use this clock**.
2. In the **Clock Tap Settings** section:
   a. Turn off **Enter output clock frequency:**.
   b. Turn on **Enter output clock parameters:**.
   c. For **Clock multiplication factor**, type `2`.
   d. For **Clock division factor**, type `1`.
   e. For **Clock phase shift**, type `1.00` and select **deg**.
   f. For **Clock duty cycle (%)**, type `50.00`.
3. Leave the other options as the default.
4. Click **Finish**. The `shift_pll` module is built.
5. In the **Symbol** dialog box, click **OK**.

6. Move the pointer to put the **shift_pll** symbol between the input and output ports in the **shift_clk.bdf**. Click to place the symbol. You have now completed the design file shown in Figure 2–4.
7. On the File menu, click **Save Project** to save the design.

Figure 2–4. ALTPLL shift_pll Design Schematic



Implement the shift_clk Design

In this section you will assign the EP1S10F780C5 device to the project and compile the project.

1. On the Assignments menu, click **Settings**.
2. In the Category list, click **Device**.
3. In the Target device section, under the Available devices list, select **EP1S10F780C5**.
4. Leave all other selections as the default.
5. Click **OK**.
6. Click **Start**, or on the Processing menu, click **Start Simulation**.
7. If prompted to **Save changes to shift_clk?**, click **Yes** to save changes.
8. When the **Full Compilation was successful** message box appears, click **OK**.
9. To view how the module is implemented in the Stratix device, from the Assignments menu, click **Timing Closure Floorplan**.

Simulate the shift_clk Design in the ModelSim-Altera Software

In this section you will simulate the design in the ModelSim-Altera software to compare the results of both simulators. Note that this ModelSim design example is for the ModelSim-Altera (Verilog) version.

You need to be familiar with the ModelSim-Altera software before trying out the design example. If you are unfamiliar with the ModelSim-Altera software, refer to the ModelSim-Altera support page on the Altera website (www.altera.com). On this support page, there are links to such topics as installation, usage, and troubleshooting.

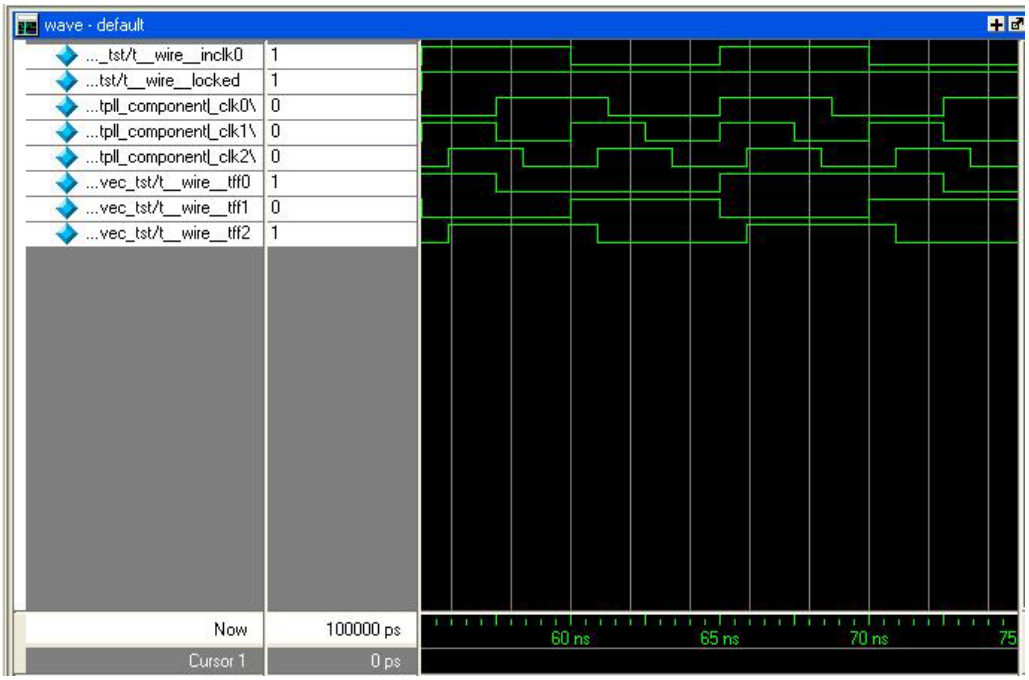
Set up the ModelSim-Altera simulator by performing the following steps.

1. Unzip the **shift_clk_msim.zip** file to any working directory on your PC.
2. Browse to select the folder in which you unzipped the files, and open the **shift_clk.do** file in a text editor.
3. In line 1 of the **shift_clk.do** file, replace <*insert_directory_path_here*> with the directory path of the appropriate library files. For example, `C:/Modeltech_ae/altera/verilog/stratix`.
4. On the File menu, click **Save**.
5. Start the ModelSim-Altera software.
6. On the File menu, click **Change Directory**.
7. Select the folder in which you unzipped the files. Click **OK**.
8. On the Tools menu, click **Execute Macro**.
9. Select the **shift_clk.do** file and click **Open**. The **shift_clk.do** file is a script file for the ModelSim-Altera software to automate all the necessary settings for the simulation.

10. Verify the results in the Waveform Viewer window.

    You may need to rearrange signals, remove redundant signals, and change the radix to suit the results in the Quartus II Simulator. Figure 2–5 shows the expected simulation results in the ModelSim-Altera software.

Figure 2–5. ModelSim Simulation Results



Ports and Parameters

The Quartus® II software provides the ALTPLL megafunction that supports PLL functionality. This chapter describes the ports and parameters of the ALTPLL megafunction.

The parameter details are only relevant for users who bypass the MegaWizard® Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in their design. The details of these parameters are hidden from MegaWizard Plug-In Manager interface users. The options listed in this section describe all of the ports and parameters that are available for each device to customize the ALTPLL megafunction according to your application.

Refer to the latest version of the Quartus II software Help for the most current information on the ports and parameters for this megafunction.

Table 3–1 describes the input ports, Table 3–2 describes the output ports, and Table 3–3 describes the ALTPLL megafunction parameters.

Table 3–1. ALTPLL Megafunction Input Ports (1 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
`inclk[]`	✓	The clock port that drives the PLL.	Input port `[3..0]`. If more than one `inclk[]` signal is specified, the `clkselect` signal specifies which clock is used. The `inclk[0]` port must be connected; connect other clock inputs if switching is necessary. Clock pins and clock outputs from the PLL can drive this port.	✓
`fbin`	—	The external feedback input port for the PLL.	The `fbin` port must be specified if the **Operation Mode** parameter is set to **External Feedback** mode. To complete the feedback loop, there must be a board-level connection between the `fbin` pin and the external clock output pin of the PLL. In Stratix III devices, the `fbout` pin feeds the dedicated `extclk` pin, and through a board connection, feeds the `fbin` pin. The `fbin` pin is also required for zero delay buffer (ZDB) mode to connect to mimic the `bidir` I/O. In Cyclone III devices, this pin is not required because ZDB mode is not available.	✓

Table 3–1. ALTPLL Megafunction Input Ports (2 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
pllena	—	The PLL enable signal.	When the pllena port is high, the PLL drives out a signal. When the pllena port is low, the PLL does not drive out a signal and it loses lock. The pllena port acts as a combined enable and reset pin. When the pin is reasserted, the PLL has to relock. The device contains only one enable pin. If a PLL on the device uses the enable pin, all PLLs on the device must use the same enable pin.	—
clkswitch	—	Switches between input clock ports.	The clkswitch port can only be connected if both the inclk0 and inclk1 ports are connected.	✓
areset	—	Resets all counters to initial values.	This port resets all counters, including the **GATE_LOCK_COUNTER** parameter, to their initial values. You can program the PLL after the device has been configured.	✓
pfdena	—	Enables the phase frequency detector (PFD).	Allows the VCO pin to continue to operate. When PFD is disabled, the PLL continues to operate regardless of the input clock. 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.	✓
clkena[]	—	Enables the clk[] ports to the PLL.	The clkena[] port is not applicable to Stratix III, Stratix II, Cyclone III, and Cyclone II devices.	—
extclkena[]	—	Enables the extclk[] ports to the PLL.	The extclkena[] port is not applicable to Stratix III, Stratix II, Cyclone III, and Cyclone II devices.	—
configupdate	—	Dynamic full PLL reconfiguration.	—	✓
scanclk	—	Clock signal for the serial scan chain.	—	✓
scanclkena	—	Clock enable for the serial scan chain.	The scanclkena port is available for Stratix III and Cyclone III devices only.	✓
scanaclr	—	Asynchronous clear for the real-time programming scan chain or the serial scan chain.	—	—
scandata	—	Contains the data for the serial scan chain.	—	✓

Table 3–1. ALTPLL Megafunction Input Ports (3 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
scanread	—	Read control signal.	The scanread port determines whether the serial scan chain reads input from the scandata port.	—
scanwrite	—	Write control signal.	The scanwrite port determines if the real-time programming scan chain writes to the PLL.	—
phaseupdown	—	Specifies dynamic phase adjustment up or down.	Available for Stratix III and Cyclone III devices only.	✓
phasestep	—	Specifies dynamic phase shifting.	Available for Stratix III and Cyclone III devices only.	✓
phasecounterselect[]	—	Specifies counter select.	Available for Stratix III and Cyclone III devices only.	✓

Table 3–2. ALTPLL Megafunction Output Ports (1 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
clk[]	✓	The clock output of the PLL.	Output port [WIDTH_CLOCK-1..0].	✓
extclk[]	—	The clock output that feeds the dedicated pins.	Not available for Stratix III, Stratix II, Cyclone III, and Cyclone II devices.	—
clkbad[]	—	Specifies which signal goes high.	If the inclk0 stops toggling, the clkbad0 signal is high. If the inclk1 stops toggling, the clkbad1 signal is high.	✓
activeclock	—	Specifies which clock is driving the PLL.	If this signal is low, inclk0 drives the PLL. If this signal is high, inclk1 drives the PLL.	✓
clkloss	—	Specifies when the clock switchover circuit initiates.	The clock switchover circuit initiates when the primary reference clock is not toggling correctly or if you specify with the clkswitch input port. This signal is not used in Stratix III and Cyclone III devices.	—

Table 3–2. ALTPLL Megafunction Output Ports (2 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
locked	—	Indicates the PLL status.	When the PLL is locked, the signal is VCC. When the PLL is out-of-lock, the signal is GND. The locked port may pulse high and low when the PLL is in the process of achieving lock. Note that the number of cycles needed to gate the lock signal is based on the input clock. The gated lock circuitry is clocked by the input clock. The maximum lock time for the PLL is provided in the *DC and Switching Characteristics* chapter of the device handbook. Take the maximum lock time of the PLL and divide that by the period of the input clock. The result is the number of clock cycles needed to gate the lock signal. The lock signal is an asynchronous output of the PLL. The PLL lock signal is derived from the reference clock and feedback clock feeding the Phase Frequency Detector (PFD). Reference clock = Input Clock/N Feedback clock = VCO/M The PLL generates a locked output when the phases and frequencies of the reference clock and feedback (FB) clock are the same or within the lock circuit tolerance. When the difference between the two inputs at the PFD goes beyond the lock circuit tolerance, the PLL loses lock. The lock signal is a function of the PLL input reference clock and the feedback clock, but not exactly synchronous to those clocks because they must be outside of lock circuit tolerance before the lock signal is deasserted.	✓
scandataout	—	The data output for the serial scan chain.	You can use the scandataout output to determine when reconfiguration is complete. The last output is cleared when reconfiguration is finished.	✓
fbout	—	Specifies the output to the mimic circuitry and feeds into the fbin port.	This port is fed by the M counter and drives the output MUX. If a feedback path is not connected, the compiler automatically connects fbout to fbin. Additionally, clkbuf is added to specify the resource type used, similar to other clock networks. This port is not required for Cyclone III devices. This port is available only if the specified operation mode of the PLL is External Feedback mode.	✓
enable0	—	Enable pulse output port.	This port is available only when the ALTPLL megafunction is in LVDS mode.	—

Table 3–2. ALTPLL Megafunction Output Ports (3 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III or Cyclone III Devices?
enable1	—	Enable pulse output port.	This port is available only when the ALTPLL megafunction is in LVDS mode.	—
sclkout0	—	Serial clock output port.	This port is available only when the ALTPLL megafunction is in LVDS mode.	—
sclkout1	—	Serial clock output port.	This port is available only when the ALTPLL megafunction is in LVDS mode.	—
vcoover range	—	Specifies whether the VCO frequency has exceeded the legal VCO range.	—	—
vcounder range	—	Specifies whether the VCO frequency has not met the legal VCO range.	—	—
phasedone	—	Specifies whether dynamic phase reconfiguration is complete.	—	✓
scandone	—	Output signal that determines when reconfiguration is complete.	The scandone signal goes high when scanchain write initiates and goes low when the PLL completes reconfiguration.	✓

Table 3–3. ALTPLL Megafunction Parameters (1 of 7)

Parameter	Type	Required?	Comments
OPERATION_MODE	String	✓	Specifies the operation of the PLL. Values are EXTERNAL_FEEDBACK, NO_COMPENSATION, NORMAL, ZERO_DELAY_BUFFER, and SOURCE_SYNCHRONOUS. If omitted, the default is NORMAL. ■ In No Compensation mode, the PLL does not align a clock to the input, which leads to better jitter performance. ■ In Source-Synchronous mode, the clock delay from pin to I/O input register matches the data delay from pin to I/O input register. ■ Source-Synchronous mode can be used with Cyclone II and Stratix II devices. This allows the clock delay from pin to I/O input register to match the data delay from pin to I/O input register. ■ In Normal mode, the PLL compensates for the delay of the internal clock network used by the clock output specified in the COMPENSATE_CLOCK parameter. If the PLL is also used to drive an external clock output pin, a corresponding phase shift of the output pin results. ■ In Zero Delay Buffer mode, the PLL must feed an external clock output pin and compensate for the delay introduced by that pin. The signal observed on the pin is synchronized to the input clock. If the PLL is also used to drive the internal clock network, a corresponding phase shift of that network results. ■ In External Feedback mode, the fbin input port must be connected to an input pin, and there must be a board-level connection between this input pin and an external clock output pin, which is specified with FEEDBACK_SOURCE parameter. The fbin port is aligned with the input clock. Use the maximum input delay assignment on the fbin port to specify external board delay. Note that for Source-Synchronous mode and Zero Delay Buffer mode, you need to make assignments (in this case, the PLL_COMPENSATE assignment) in addition to setting the appropriate mode in the megafunction. This allows you to specify an output pin as a compensation target for a PLL in Zero Delay Buffer or External Feedback mode, or an input pin or a group of input pins as compensation targets for a PLL in Source-Synchronous mode. If assigned to an output pin, the pin must be fed by the external clock output port of a PLL in a Stratix, HardCopy Stratix or Cyclone device, or the compensated clock output port of a PLL in other devices. Any other output pins fed by the same PLL generally are not delay compensated, especially if they have different I/O standards. If assigned to an input pin or a group of input pins, the input pins must drive input registers that are clocked by the compensated clock output port of a PLL in Source-Synchronous mode. This option is ignored if it is applied to anything other than an output or input pin as described previously.

Table 3–3. ALTPLL Megafunction Parameters (2 of 7)

Parameter	Type	Required?	Comments
PLL_TYPE	String	—	Specifies the type of PLL to instantiate. Values are **AUTO**, **ENHANCED**, **FAST**, **TOP/BOTTOM** and **LEFT/RIGHT**. If omitted, the default is **AUTO**.
COMPENSATE_CLOCK	String	—	Specifies the output clock port which should be compensated. If the OPERATION_MODE parameter is set to **NORMAL**, the values are CLK[], GCLK[], LCLK[], or LVDSCLK[]. ■ If the OPERATION_MODE parameter is set to Zero Delay Buffer, value is EXTCLK[]. ■ If the OPERATION_MODE parameter is set to Source-Synchronous, the values are CLK[], LCLK[], GCLK[], or LVDSCLK[]. This clock cannot offset with respect to the reference clock. This relationship is preserved closely even upon temperature and frequency changes. ■ If the OPERATION_MODE parameter is set to **NORMAL**, the values are CLK[], LCLK[], GCLK[], or LVDSCLK[]. ■ In Normal mode, default is CLK0. ■ In Zero Delay Buffer mode, default is EXTCLK0. For example, if CLK0 is set when the OPERATION_MODE parameter is specified to **NORMAL**, the compiler's compensation selection, in terms of GCLK[], LCLK[], or LVDSCLK[], is based on CLK0 routing.
SCAN_CHAIN	String	—	Specifies the length of the scan chain. Values are **LONG** or **SHORT**. If omitted, the default is **LONG**. If **LONG** is specified, the scan chain length is 10 counters. If **SHORT** is specified, the scan chain length is 6 counters.
PRIMARY_CLOCK	String	—	Specifies the primary reference clock of the PLL. Values are **INCLK0** or **INCLK1**. If omitted, the default is **INCLK0**. Use the clock switch scheme to switch between clocks. Automatic clock switchover is defined differently in different device families; refer to the relevant PLL chapter in the relevant device handbook.
INCLK0_INPUT_FREQUENCY	Integer	✓	Specifies the input frequency for the inclk0 clock. The Compiler uses the frequency of the clk0 port to calculate the PLL parameters, but also analyzes and reports the phase shifts for the clk1 port.
INCLK1_INPUT_FREQUENCY	Integer	—	Specifies the input frequency for the inclk1 clock. The Compiler uses the frequency of the clk0 port to calculate the PLL parameters, but also analyzes and reports the phase shifts for the clk1 port.
GATE_LOCK_SIGNAL	String	—	Specifies whether the locked port should be gated internally with a 20-bit programmable counter so it does not oscillate during initial power-up. Values are **NO** and **YES**. If omitted, default is **NO**.
GATE_LOCK_COUNTER	Integer	—	Specifies the value for the 20-bit counter that gates the locked output port before sending it to the locked port. This parameter is required for simulation with other EDA simulators.

Table 3–3. ALTPLL Megafunction Parameters (3 of 7)

Parameter	Type	Required?	Comments
LOCK_HIGH	Integer	—	Specifies the number of half-clock cycles that the output clocks must be locked before the locked port goes high. This parameter is required for simulation with other EDA simulators. Available for Stratix III and Cyclone III devices only.
LOCK_LOW	Integer	—	Specifies the number of half-clock cycles that the output clocks must be out-of-lock before the locked port goes low. This parameter is required for simulation with other EDA simulators. Available for Stratix III and Cyclone III devices only.
SWITCH_OVER_ON_LOSSCLK	String	—	Specifies whether the loss-of-lock condition should initiate a clock switch over. Values are **ON** or **OFF**. If omitted, the value is **OFF**.
SWITCH_OVER_COUNTER	String	—	Specifies, in clock cycles after a switchover condition, when the input clock is switched. Values are **0** through **31**. If omitted, the value is **0**.
SWITCH_OVER_TYPE	String	—	Specifies the switchover type. If omitted, the value is **AUTO**.
ENABLE_SWITCH_OVER_ COUNTER	String	—	Specifies whether to use the SWITCH_OVER_COUNTER parameter. Values are **ON** or **OFF**. If omitted, the value is **OFF**.
FEEDBACK_SOURCE	String	—	Specifies which clock output has a board-level connection to the fbin port. If the OPERATION_MODE parameter is specified to EXTERNAL_FEEDBACK, the FEEDBACK_SOURCE parameter is used. Values are **EXTCLK[]**. If omitted, the value is **EXTCLK0**.
BANDWIDTH	Integer	—	Specifies, in megahertz (MHz), bandwidth of the PLL. If this parameter is not specified, the Compiler automatically determines the value of the **BANDWIDTH** parameter to satisfy other PLL settings.
BANDWIDTH_TYPE	String	—	Specifies the type of bandwidth for BANDWIDTH. Values are **AUTO**, **CUSTOM**, **HIGH**, **LOW**, or **MEDIUM**. If omitted, default value is **AUTO**. For the low bandwidth option, the PLL has a better jitter rejection but slower lock time. For the high bandwidth option, the PLL has a faster lock time but tracks more jitter. The medium option is a balance between both previous options.
SPREAD_FREQUENCY	String	—	Specifies, in picoseconds (ps), the modulation frequency for spread spectrum.
DOWN_SPREAD	String	—	Specifies the down spectrum percentage. Values range from **0** through **0.5**.
INVALID_LOCK_MULTIPLIER	Integer	—	Specifies the scaling factor, in half-clock cycles, for which the clock output ports must be out-of-lock before the locked pin goes low.
VALID_LOCK_MULTIPLIER	Integer	—	Specifies the scaling factor, in half-clock cycles, for which the clock output ports must be locked before the locked pin goes high.
C[]_HIGH	Integer	—	Parameter [9..0]. Specifies the high period count for the corresponding C[9..0] counter. If omitted, default is **1**. Counters C[9..5] are not available in Cyclone III devices.

Table 3–3. ALTPLL Megafunction Parameters (4 of 7)

Parameter	Type	Required?	Comments
C[]_LOW	Integer	—	Parameter [9..0]. Specifies the low period count for the corresponding C[9..0] counter. If omitted, default is **1**. Counters C[9..5] are not available in Cyclone III devices.
C[]_INITIAL	Integer	—	Parameter [9..0]. Specifies the initial value for the corresponding C[9..0] counter. If omitted, default is **1**. Counters C[9..5] are not available in Cyclone III devices.
C[]_PH	Integer	—	Parameter [9..0]. Specifies the phase tap for the C[9..0] counter. If omitted, default is **0**. Counters C[9..5] are not available in Cyclone III devices.
C[]_MODE	String	—	Parameter [9..0]. Specifies the mode for the corresponding C[9..0] counter. Values are **BYPASS**, **ODD**, and **EVEN**. If omitted, the default is **BYPASS**. Counters C[9..5] are not available in Cyclone III devices.
C[]_TEST_SOURCE	Integer	—	Parameter [9..0]. Specifies the test source for the corresponding C[9..0] counter. If omitted, default is **0**. Counters C[9..5] are not available in Cyclone III devices.
C[]_USE_CASC_IN	String	—	Parameter [9..0]. Specifies whether to use cascade input for the corresponding C[9..0] counter. Values are **ON** and **OFF**. If omitted, default is **OFF**. Counters C[9..5] are not available in Cyclone III devices.
CLK[]_OUTPUT_FREQUENCY	Integer	—	Parameter [2..0]. Specifies the output frequency of the corresponding CLK[2..0] port. This parameter is ignored if the corresponding clk[2..0] port is not used. This parameter is unavailable if multiplication or division factors are specified. If omitted, default is **0**.
CLK[]_MULTIPLY_BY	Integer	—	Parameter [9..0]. Specifies the integer multiplication factor for the VCO frequency for the corresponding CLK[9..0] port. The value must be greater than **0**. Specify this parameter only if the corresponding clk[9..0] port is used; however, it is not required if a **Clock Settings** assignment is specified for the corresponding clk[9..0] port. If omitted, the default is **0**. Parameters CLK[9..5]_MULTIPLY_BY are not available in Cyclone III devices.
CLK[]_DIVIDE_BY	Integer	—	Specifies the integer division factor for the VCO frequency for the corresponding CLK[5..0] port. The value must be greater than **0**. Specify this parameter only if the corresponding clk[5..0] port is used; however, it is not required if a **Clock Settings** assignment is specified for the corresponding clk[5..0] port. If omitted, the default is **0**. Parameters CLK[9..5]_DIVIDE_BY are not available in Cyclone III devices.

Table 3–3. ALTPLL Megafunction Parameters (5 of 7)

Parameter	Type	Required?	Comments
CLK[]_PHASE_SHIFT	Integer	—	Specifies, in picoseconds (ps), the phase shift for the corresponding clk[9..0] port. If omitted, the default is **0**. Parameters CLK[9..5]_PHASE_SHIFT are not available in Cyclone III devices.
CLK[]_TIME_DELAY	String	—	Specifies, in picoseconds (ps), a delay value to be applied to the corresponding clk[5..0] port. This parameter affects only the corresponding clk[5..0] port and is independent of the corresponding CLK[5..0]_PHASE_SHIFT parameter; therefore, you can use the two ports simultaneously. If no units are specified, the default is picoseconds (ps). Legal time delay values range from **–3 ns** through **6 ns** in increments of **0.25 ns**. Do not use these values as parameters except when reprogramming the PLL using the real-time programming interface.
CLK[]_DUTY_CYCLE	Integer	—	Specifies the duty cycle for the corresponding clk[9..0] port by providing the percentage of high time. Parameters CLK[9..5]_DUTY_CYCLE are not available in Cyclone III devices. If omitted, the default is **50**.
CLK[]_USE_EVEN_ COUNTER_MODE	String	—	Specifies whether the clock output needs to be forced to use even counter mode for the corresponding CLK[9..0] port. If omitted, the default is **OFF**. **Note:** Clock output widths vary by device. Refer to the PLL chapter in the specific device handbook.
CLK[]_USE_EVEN_ COUNTER_VALUE	String	—	Specifies whether the clock output needs to be forced to use even counter values for the corresponding CLK[9..0] port. If omitted, the default is **OFF**. **Note:** Clock output ports vary by device. Refer to the PLL chapter in the relevant device handbook.
EXTCLK[]_MULTIPLY_BY	Integer	—	Specifies the integer multiplication factor for the corresponding extclk[3..0] port with respect to the input clock frequency. The value must be greater than **0**. You can specify this parameter only if you use the corresponding extclk[3..0] port is used. However, it is not required if a **Clock Settings** assignment is specified for the corresponding extclk[3..0] port. If omitted, the default is **1**. This parameter is not available for Stratix II devices.
EXTCLK[]_DIVIDE_BY	Integer	—	Specifies the integer division factor for the corresponding extclk[3..0] port with respect to the input clock frequency. The value must be greater than **0**. You can specify this parameter only if the corresponding extclk[3..0] port is used; however, it is not required if a **Clock Settings** assignment is specified for the corresponding extclk[3..0] port. If omitted, the default is **1**. This parameter is not available for Stratix II devices.

Table 3–3. ALTPLL Megafunction Parameters (6 of 7)

Parameter	Type	Required?	Comments
EXTCLK[]_PHASE_SHIFT	Integer	—	Specifies the phase shift for the corresponding extclk[3..0] port. This parameter is not available for Stratix II devices.
EXTCLK[]_TIME_DELAY	String	—	Specifies, in picoseconds (ps), a delay value to be applied to the corresponding extclk[3..0] port. The EXTCLK[3..0]_TIME_DELAY parameter affects only the corresponding extclk[3..0] port and is independent of the EXTCLK[3..0]_PHASE_SHIFT parameter; therefore you can use the two ports simultaneously. If no units are specified, picoseconds (ps) are assumed. Legal values range from **–3 ns** through **6 ns** in increments of **0.25 ns**. Do not use these values as parameters except when reprogramming the PLL using the real-time programming interface. This parameter is not available for Stratix II devices.
EXTCLK[]_DUTY_CYCLE	Integer	—	Specifies the duty cycle for the corresponding extclk[3..0] port. If omitted, the default is **50**. This parameter is not available for Stratix II devices.
VCO_FREQUENCY_CONTROL	String	—	Specifies the frequency control value for the VCO pin. Values are **AUTO**, **MANUAL_FREQUENCY**, and **MANUAL_PHASE**. If omitted, the default is **AUTO**. ■ **AUTO**—VCO_MULTIPLY_BY and VCO_DIVIDE_BY values are ignored and VCO frequency is set automatically. ■ **MANUAL_FREQUENCY**—Specifies the VCO frequency as a multiple of the input frequency. ■ **MANUAL_PHASE**—Specifies the VCO frequency as a phase shift step value.
VCO_MULTIPLY_BY	Integer	—	Specifies the integer multiplication factor for the VCO pin. If omitted, the default is **0**.
VCO_DIVIDE_BY	Integer	—	Specifies the integer division factor for the VCO pin. If omitted, the default is **0**. If VCO_FREQUENCY_CONTROL is set to MANUAL_PHASE, specify the VCO frequency as a phase shift step value; that is, one-eighth of the VCO period.
VCO_POST_SCALE	Integer	—	Specifies the VCO operating range. The VCO post-scale divider value is **1** or **2**. If omitted, the default is **1**.
VCO_PHASE_SHIFT_STEP	Integer	—	Specifies the phase shift for the VCO pin. if omitted, the default is **0**.
WIDTH_CLOCK	Integer	—	Specifies the clock width. Values are **10** for Stratix III devices, **5** for Cyclone III devices, and **6** for all other supported device families. If omitted, the default is **6**. For Stratix III and Cyclone III device designs, the WIDTH_CLOCK parameter is required for both clear box and non-clear box implementation to reflect the correct width.

Table 3–3. ALTPLL Megafunction Parameters (7 of 7)

Parameter	Type	Required?	Comments
SELF_RESET_ON_LOSS_LOCK	String	—	Specifies the gate-lock counter. If omitted, the default is **OFF**.
SELF_RESET_ON_ GATED_LOSS_LOCK	String	—	Specifies the gate-lock counter. If omitted, the default is **OFF**.
SKIP_VCO	String	—	If omitted, the default is **OFF**.
PFD_MIN	Integer	—	Specifies the minimum value for the PFD pin.
PFD_MAX	Integer	—	Specifies the maximum value for the PFD pin.
M_INITIAL	Integer	—	Specifies the initial value for the M counter. Provides direct access to the internal PLL parameters. If the M_INITIAL parameter is specified, all advanced parameters must be used. If omitted, the default is **1**.
M	Integer	—	Specifies the modulus for the M counter. Provides direct access to the internal PLL parameters. If the M parameter is specified, all advanced parameters must be used. Values range from **1** through **512**. If omitted, the default is **0**.
M_PH	Integer	—	Specifies the phase tap for the M counter. Values range from **0** through **7**. If omitted, the default is **0**.
M_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the M_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**. This parameter is not available for Cyclone II and Stratix II devices.
N_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the N_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**. This parameter is not available for Cyclone II and Stratix II devices.
QUALITY_CONF_DONE	String	—	If omitted, the default is **OFF**.
SCLKOUT[]_PHASE_SHIFT	Integer	—	Parameter [1..0]. Specifies, in picoseconds (ps), the phase shift for the corresponding sclkout[1..0] output port. The maximum phase value is 7/8 of one VCO period. The VCO phase tap is shared with the corresponding clk[1..0] output port, and must have the same phase amount that is less than one VCO period. In LVDS mode, this parameter default value is **0**.
CLK[]_COUNTER	String	—	Parameter [9..0]. Specifies the counter for the corresponding clk[9..0] port. Values are **G0**, **G1**, **G2**, **G3**, **L0**, or **L1**. If omitted, the default is **L0**. This parameter is not available for Cyclone II and Stratix II devices. Counters CLK[9..5]_COUNTER are not available for Cyclone III devices.

Note to Table 3–3:

(1) For device-specific clock and PLL information, refer to the device-specific handbook available in the Literature section of the Altera website (www.altera.com).

Table 3–4 shows the advanced parameters for the ALTPLL megafunction. Advanced parameters offer full control over a device. These parameters are not available from the MegaWizard Plug-In Manager.

Do not use advanced ALTPLL megafunction parameters with other ALTPLL megafunction parameters that are set in the MegaWizard Plug-In Manager. OPERATION_MODE is always a required parameter.

For more information about using advanced parameters for the ALTPLL megafunction, contact Altera Applications.

Table 3–4. Advanced ALTPLL Megafunction Parameters (1 of 6)

Parameter	Type	Required?	Comments
VCO_MIN	String	—	Specifies the minimum value for the VCO pin. These are only simulation parameters.
VCO_MAX	String	—	Specifies the maximum value for the VCO pin. These are only simulation parameters.
VCO_CENTER	String	—	Specifies the center value for the VCO pin. These are only simulation parameters.
PFD_MIN	String	—	Specifies the minimum value for the PFD pin.
PFD_MAX	String	—	Specifies the maximum value for the PFD pin.
M_INITIAL	Integer	—	Specifies the initial value for the M counter. Provides direct access to the internal PLL parameters. If the M_INITIAL parameter is specified, all advanced parameters must be used. Values range from **1** through **512**. If omitted, the default is **1**. **Note**: For device-specific clock and PLL information, refer to the device-specific handbook available in the Literature section of the Altera website (www.altera.com).
M	Integer	—	Specifies the modulus for the M counter. Provides direct access to the internal PLL parameters. If the M parameter is specified, all advanced parameters must be used. Values range from **1** through **512**. If omitted, the default is **0**.
N	Integer	—	Specifies the modulus for the **N** counter. Provides direct access to the internal PLL parameters. If the **N** parameter is specified, all advanced parameters must be used. Values range from **1** through **512**.
M2	Integer	—	Specifies the spread spectrum modulus for the M counter. Provides direct access to the internal PLL parameters. If the M2 parameter is specified, all advanced parameters must be used. Values range from **1** through **512**.
N2	Integer	—	Specifies the spread spectrum modulus for the N counter. Provides direct access to the internal PLL parameters. If the N2 parameter is specified, all advanced parameters must be used. Values range from **1** through **512**.

Table 3–4. Advanced ALTPLL Megafunction Parameters (2 of 6)

Parameter	Type	Required?	Comments
SS	Integer	—	Specifies the modulus for the spread spectrum counter. Provides direct access to the internal PLL parameters. If the SS parameter is specified, all advanced parameters must be used. Values range from **1** through **32768**.
E0_HIGH	Integer	—	Specifies the high period count for the E0_HIGH counter. Values range from **1** through **512**. If omitted, the default is 1.
E0_LOW	Integer	—	Specifies the low period count for the E0_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
E0_INITIAL	Integer	—	Specifies the initial value for the E0_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
E0_MODE	String	—	Specifies the mode for the E0_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
E0_PH	Integer	—	Specifies the phase tap for the E0_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
E0_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the E0_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
E1_HIGH	Integer	—	Specifies the high period count for the E1_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
E1_LOW	Integer	—	Specifies the low period count for the E1_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
E1_INITIAL	Integer	—	Specifies the initial value for the E1_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
E1_MODE	String	—	Specifies the mode for the E1_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
E1_PH	Integer	—	Specifies the phase tap for the E1_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
E1_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the E1_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
E2_HIGH	Integer	—	Specifies the high period count for the E2_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
E2_LOW	Integer	—	Specifies the low period count for the E2_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
E2_INITIAL	Integer	—	Specifies the initial value for the E2_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
E2_MODE	String	—	Specifies the mode for the E2_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
E2_PH	Integer	—	Specifies the phase tap for the E2_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
E2_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the E2_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.

Table 3–4. Advanced ALTPLL Megafunction Parameters (3 of 6)

Parameter	Type	Required?	Comments
E3_HIGH	Integer	—	Specifies the high period count for the E3_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
E3_LOW	Integer	—	Specifies the low period count for the E3_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
E3_INITIAL	Integer	—	Specifies the initial value for the E3_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
E3_MODE	String	—	Specifies the mode for the E3_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
E3_PH	Integer	—	Specifies the phase tap for the E3_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
E3_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the E3_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
G0_HIGH	Integer	—	Specifies the high period count for the G0_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
G0_LOW	Integer	—	Specifies the low period count for the G0_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
G0_INITIAL	Integer	—	Specifies the initial value for the G0_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
G0_MODE	String	—	Specifies the mode for the G0_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
G0_PH	Integer	—	Specifies the phase tap for the G0_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
G0_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the G0_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
G1_HIGH	Integer	—	Specifies the high period count for the G1_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
G1_LOW	Integer	—	Specifies the low period count for the G1_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
G1_INITIAL	Integer	—	Specifies the initial value for the G1_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
G1_MODE	String	—	Specifies the mode for the G1_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
G1_PH	Integer	—	Specifies the phase tap for the G1_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
G1_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the G1_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
G2_HIGH	Integer	—	Specifies high period count for G2_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
G2_LOW	Integer	—	Specifies the low period count for the G2_LOW counter. Values range from **1** through **512**. If omitted, the default is 1.

Table 3–4. Advanced ALTPLL Megafunction Parameters (4 of 6)

Parameter	Type	Required?	Comments
G2_INITIAL	Integer	—	Specifies the initial value for the G2_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
G2_MODE	String	—	Specifies the mode for the G2_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
G2_PH	Integer	—	Specifies the phase tap for the G2_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
G2_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the G2_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
G3_HIGH	Integer	—	Specifies the high period count for the G3_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
G3_LOW	Integer	—	Specifies the low period count for the E0_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
G3_INITIAL	Integer	—	Specifies the initial value for the G3_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
G3_MODE	String	—	Specifies the mode for the G3_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.
G3_PH	Integer	—	Specifies the phase tap for the G3_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
G3_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the G3_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
L0_HIGH	Integer	—	Specifies the high period count for the L0_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
L0_LOW	Integer	—	Specifies the low period count for the L0_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
L0_INITIAL	Integer	—	Specifies the initial value for the L0_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
L0_MODE	String	—	Specifies the mode for the L0_MODE counter. Values are **BYPASS**, **ODD** or **EVEN**. If omitted, the default is **BYPASS**.
L0_PH	Integer	—	Specifies the phase tap for the L0_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
L0_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the L0_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
L1_HIGH	Integer	—	Specifies the high period count for the L1_HIGH counter. Values range from **1** through **512**. If omitted, the default is **1**.
L1_LOW	Integer	—	Specifies the low period count for the L1_LOW counter. Values range from **1** through **512**. If omitted, the default is **1**.
L1_INITIAL	Integer	—	Specifies the initial value for the L1_INITIAL counter. Values range from **1** through **512**. If omitted, the default is **1**.
L1_MODE	String	—	Specifies the mode for the L1_MODE counter. Values are **BYPASS**, **ODD**, or **EVEN**. If omitted, the default is **BYPASS**.

Table 3–4. Advanced ALTPLL Megafunction Parameters (5 of 6)

Parameter	Type	Required?	Comments
L1_PH	Integer	—	Specifies the phase tap for the L1_PH counter. Values range from **0** through **7**. If omitted, the default is **0**.
L1_TIME_DELAY	Integer	—	Specifies, in nanoseconds (ns), the time delay for the L1_TIME_DELAY counter. Values range from **0** ns through **3** ns. If omitted, the default is **0**.
EXTCLK[]_COUNTER	String	—	Specifies the external counter for the corresponding extclk[3..0] port. Values are **E0**, **E1**, **E2**, or **E3**. If omitted, the default is **E[].** This parameter is available for Stratix (extclk[3..0]), Stratix GX (extclk[3..0]), and Cyclone (extclk[0]) devices.
CLK[]_COUNTER	String	—	Parameter [9..0]. Specifies the counter for the corresponding clk[9..0] port. Values are **G0**, **G1**, **G2**, **G3**, **L0**, or **L1**. If omitted, the default is **L0**. This parameter is not available for Cyclone II and Stratix II devices. Counters CLK[9..5]_COUNTER are not available for Cyclone III devices.
ENABLE[]_COUNTER	String	—	Specifies the counter for the corresponding enable[1..0] port. Values are **L0** or **L1**.
CHARGE_PUMP_CURRENT	Integer	—	Specifies, in microamperes (mA), the value of the charge pump current. Refer to the *DC & Switching Characteristics* chapter of the appropriate device handbook for the supported charge pump current value ranges.
LOOP_FILTER_C	Integer	—	Specifies, in picofarads (pF), the value of the loop capacitor. Values range from **5** to **20** pF. The Compiler cannot achieve all values. If omitted, the default value is **10**.
LOOP_FILTER_R	Integer	—	Specifies, in kilohms (K), the value of the loop resistor. Values range from **1** K through **20** K. The Compiler cannot achieve all values.
INTENDED_DEVICE_FAMILY	String	—	This parameter is used for modeling and behavioral simulation purposes. Create the ALTPLL megafunction with the MegaWizard Plug-In Manager to calculate the value for this parameter.

Table 3–4. Advanced ALTPLL Megafunction Parameters (6 of 6)

Parameter	Type	Required?	Comments
SCLKOUT0_PHASE_SHIFT	Integer	—	Specifies in picoseconds the phase shift of the given sclkout output. The sclkout[0] output can only use the VCO phase taps to implement phase, so the maximum legal phase value is 7/8th of one VCO period. The VCO phase tap is shared with the corresponding clk[0] output, so both must have the same "fine grain" phase (for example, phase amount that is less than one VCO period). In LVDS mode, this parameter default value is **0**.
SCLKOUT1_PHASE_SHIFT	Integer	—	Specifies in picoseconds the phase shift of the given sclkout output. The sclkout[1] output can only use the VCO phase taps to implement phase, so the maximum legal phase value is 7/8th of one VCO period. The VCO phase tap is shared with the corresponding clk[1] output, so both must have the same "fine grain" phase (for example, phase amount that is less than one VCO period). In LVDS mode, this parameter defaults to a phase of **0**.

Document Revision History

The following table shows the revision history for the chapters in this user guide.

Date	Document Version	Changes Made
December 2008	7.0	■ Updated the following sections: →"Device Family Support" section →"Introduction" section →"Features" section →"General Description" section →"Design Examples" section →"Simulation" section →"Ports and Parameters" section →"How to Contact Altera" section ■ Removed the following sections: →"Resource Utilization & Performance" section →"Software and System Requirements" section →"Instantiating Multifunction in HDL Code" section →"Identifying a Multifunction after Compilation" section →"Signature II Embedded Logic Analyzer" section ■ Removed all screenshots on "Using the MegaWizard Plug-In Manager" section ■ Reorganized the "Using the MegaWizard Plug-In Manager" section into table format. ■ Renamed "About this User Guide" section to "Additional Information" and moved the section to the end of the user guide.
March 2007	6.0	Updates for software version 7.0, including: ■ Added support for Cyclone III devices ■ Added Referenced Documents section
December 2006	5.0	Updated to reflect new document organization, additions, and GUI changes for Quartus 6.1, including adding information relating to Stratix® III devices
May 2006	4.0	Updated to reflect new document organization, additions, and GUI changes for Quartus 6.0
December 2004	3.0	Updated to reflect new document organization and GUI changes

Referenced Documents

This user guide references the following documents:

- *AN 409: Design Example Using the ALTLVDS Megafunction and the External PLL Option in Stratix II Devices Design Example*
- *ALTLVDS Megafunction User Guide*
- *ALTPLL_RECONFIG Megafunction User Guide*
- *Clock Networks and PLLs in Stratix III Devices* chapter of the *Stratix III Device Handbook*
- *Clock Networks and PLLs in Cyclone III Devices* chapter of the *Cyclone III Device Handbook*
- *ALTMEMPHY Megafunction User Guide*
- *Stratix IV Device Handbook*
- *Stratix III Device Handbook*
- *Stratix II Device Handbook*
- *Stratix II GX Device Handbook*
- *Stratix Device Handbook*
- *Stratix GX Device Handbook*
- *Cyclone III Device Handbook*
- *Cyclone II Device Handbook*
- *HardCopy Series Handbook*
- *Timing Analysis* section in volume 3 of the *Quartus II Handbook*
- *Design and Synthesis* section in volume 1 of the *Quartus II Handbook*
- *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown in the following table.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.