altpll Megafunction

User Guide

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com

Quartus II Version: 6.1
Document Version: 5.0
Document Date: December 2006

UG-ALTPLL-5.0

Contents

About this User Guide

Document Revision History

The following table displays the revision history for this User Guide.

Date	Document Version	Changes Made
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

How to Contact Altera

For the most up-to-date information about Altera® products, go to the Altera world-wide web site at www.altera.com. For technical support on this product, go to www.altera.com/mysupport. For additional information about Altera products, consult the sources shown below.

Information Type	USA & Canada	All Other Locations
Technical support	www.altera.com/mysupport/	www.altera.com/mysupport/
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	(408) 544-7000 *(1)* (7:00 a.m. to 5:00 p.m. Pacific Time)
Product literature	www.altera.com	www.altera.com
Altera literature services	literature@altera.com *(1)*	literature@altera.com *(1)*
Non-technical customer service	(800) 767-3753	+ 1 408-544-7000 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
FTP site	ftp.altera.com	ftp.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 below.

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, filenames, filename 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.

1. About this Megafunction

Device Family Support

Megafunctions provide either full or preliminary support for target Altera® device families, as described below:

- *Full support* means the megafunction meets all functional and timing requirements for the device family and may be used in production designs.
- *Preliminary support* means the megafunction meets all functional requirements, but may still be undergoing timing analysis for the device family; it may be used in production designs with caution.

Table 1–1 shows the level of support offered by the altpll megafunction for each Altera device family.

Table 1–1. Device Family Support

Device Family	Support
Cyclone®	Full
Cyclone II	Full
HardCopy® II	Full
HardCopy Stratix®	Full
Stratix	Full
Stratix II	Full
Stratix GX	Full
Stratix II GX	Full
Stratix III	Preliminary
All other device families	No support

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 megafunction's size by setting parameters.

Features

The altpll megafunction configures the phase-locked loops (PLLs) in the Stratix and Cyclone series of devices. Table 1–2 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 1–3 compares features between Stratix II and Stratix III devices.

Table 1–2. altpll Megafunction Features (Part 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 any clock signals.
Asynchronous reset	`areset`	This option adds an asynchronous reset to the PLL. The active high input resets the PLL when enabled.

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 into 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 may 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 will not generate the LVDS TX/RX modules, and is merely used as the clocking scheme for these modules.
		
		For more information regarding LVDS, refer to the *LVDS Megafunction User Guide*. There is an option to set up the PLL in LVDS mode via the altpll megafunction. This option is only available for the Stratix II, Stratix II GX, and HardCopy II devices.

Table 1–2. altpll Megafunction Features (Part 2 of 3)

Feature	Port/Parameter	Description
Operation mode	OPERATION_MODE	The 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 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 the Stratix series PLLs can have multiple outputs, you must specify which output clock is used for the feedback. *(3)* For more detailed information about the altpll megafunction operation modes, refer toTable 3–3.
Dynamic configuration options	SCAN_CHAIN	The Stratix series PLLs can be dynamically reconfigured by using a scan chain. Depending on the PLL functionality you require, there are two options 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 (6 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.

Table 1–2. altpll Megafunction Features (Part 3 of 3)

Feature	Port/Parameter	Description
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 wizard 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 wizard 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 wizard 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 1–2:

(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 specific device handbook for other devices.
(3) Stratix III, Stratix II, Cyclone II, and HardCopy II devices also support source-synchronous mode.
(4) Only applicable to Stratix and Stratix GX Enhanced PLLs. Fast PLLs in Stratix II devices support dynamic reconfiguration. However, the scan_chain functionality is not applicable to all device families.
(5) Stratix III, Stratix II, 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 II, and HardCopy II devices do not support this feature. Clock enable functionality can be achieved when using the altclkctrl megafunction.

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

Table 1–3 compares the features of Stratix II and Stratix III devices.

Table 1–3. Family Feature Comparison

Feature	Stratix II	Stratix III
PLL types	Fast Enhanced	Left/right Top/bottom
Number of counters	Fast: 4 Enhanced: 6	Left/Right: 7 Top/Bottom: 10
Counter size	Fast: 4 bit Enhanced 8-9 bit	All 8 bits
EXTCLK output pins	Fast: 0 Enhanced: 6 single-ended, 3 differential	Left/Right: 2 single-ended/1 differential Top/Bottom: 6 single-ended, 1 differential
EXTCLK fbin pin	Fast: none Enhanced: single-ended/differential	Left/Right: single-ended only Top/Bottom: single-ended/differential
Spread Spectrum	Yes	No
SERDES clocks	Special SLKOUT/ENABLE outputs; DPACLK is same frequency as SLKOUT and connection is implied	Regular counter outputs to LVDSCLK/LDEN clock network.
PLL cascading	Via clock net only	Via clock net and dedicated paths between adjacent PLLs
Dynamic reconfiguration	Yes	Yes, but very different from Stratix II: • Can perform phase-reconfiguration separately and more simple (no serial scan-chain, parallel interface). Does not require altpll_reconfig • Reconfig counters use serial scan-chain, but follow a different protocol. Requires altpll_reconfig
PLL areset & enable control	Yes, both	No, only areset control. The behavior of areset is essentially the same with pllena

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

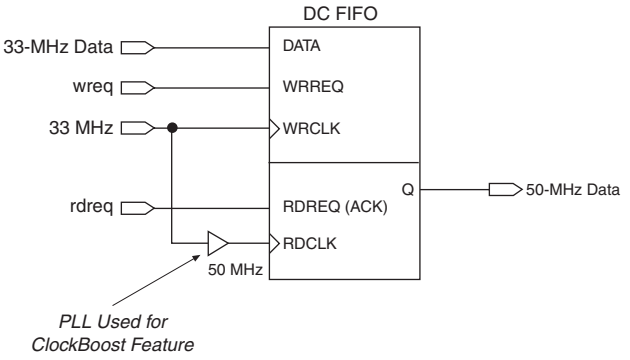
Clock Domain Transfers

For data transfer across clock domains, certain design considerations should 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. For example, you can use the DC FIFO buffer for data transfer (Figure 1–1).

Figure 1–1. Using DCFIFO 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} may be reduced or a hold time violation may occur. This depends on the direction and magnitude of the shift (any positive shift past 180 degrees can be considered negative shift) and whether the destination or source register's clock is shifted.

Table 1–4 shows the features available in the enhanced and fast PLLs.

Table 1–4. Features for Enhanced & Fast PLLs (Part 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 n post-scale counter			$\frac{m}{\text{post-scale counter}}$	m n 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
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	✓	✓	—	—	—	—

Table 1–4. Features for Enhanced & Fast PLLs (Part 2 of 2)

Feature	Stratix II PLLs		Stratix PLLs		Cyclone II PLLs	Cyclone PLLs
	Enhanced PLL	Fast PLL	Enhanced PLL	Fast PLL		
Spread spectrum clocking	✓	—	✓	—	—	—

Notes for Table 1–4:

(1) For *m*, *n*, and post-scale counter values, see the PLL chapter in the appropriate device family handbook.
(2) The PLL output counters can drive the internal clock networks or the 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 the 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 (PLL 11 and 12 not supported for 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 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.

LVDS Mode

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 takes a high-speed differential serial data stream from a pair of input pins and converts it into a low-speed parallel stream. The LVDS transmitter takes a parallel stream of data from the core and converts 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 may share a common PLL, or they may use separate PLLs.

This option is only useful when used in conjunction with the altlvds megafunction. This option will not generate the LVDS transmitter/receiver modules, and is used simply as the clocking scheme for these modules.

For more information about the altlvds megafunction, refer to *AN 409: Design Example Using the altlvds Megafunction & the External PLL Option in Stratix II Devices Design Example*.

For more information about LVDS, refer to the *LVDS Megafunction User Guide*. There is an option to set up the PLL in LVDS mode with the altpll megafunction. This option is only available for the HardCopy II, Stratix II, and Stratix II GX devices.

General Description

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

Stratix III PLL New Features Description

The Stratix III PLL is a redesigned version of the Stratix II PLL, and contains all of the Stratix II device features, as well as some new ones. All the new features of the Stratix III PLL are described below.

Dynamic Reconfiguration

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

Phase reconfiguration is new in Stratix 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 devices, refer to the *altpll_reconfig Megafunction User Guide* and the *Clock Networks & PLLs in Stratix III Devices* chapter in the *Stratix III Handbook*.

PLL Types

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

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

LVDS Clock

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

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

Clock Switchover

Similar to Stratix II devices, Stratix III devices support Manual Switchover and Automatic Switchover with Manual override. However, Stratix III switchover is simpler and symmetric than the Stratix II switchover.

In Stratix II devices, there is no switchover counter or switchover-on-loss-of-lock. The auto-switchover happens only once, from the primary clock to secondary clock, and switches back only with user intervention.

In Stratix III devices, a counter can be created using core resources and core logic used to switch on loss-of-lock. It 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 (CLKBAD, ACTIVECLK, and so forth) will only operate when at least one good input clock exists. The input clock frequencies must be within twice of each other in order for the status signals to be correct.

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

Common Applications

The altpll megafunction is used for implementing different PLL configurations. The 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 PLLs are very useful in DDR 2/3 interfaces because of the usage of the reconfigurable PLL to implement the dynamic data path (via the ALTMEMPHY megafunction). The PLL is necessary for driving the delay-locked loop (DLL) used in the dynamic external memory interface operation.

Refer to the *ALTMEMPHY Megafunction User Guide* for further details.

Figure 1–2 shows a block diagram of a Stratix III PLL, Figure 1–3 shows a block diagram of a fast PLL, and Figure 1–4 shows the Cyclone PLLs.

For more information about the functionality of these PLLs, refer to the relevant chapters in the Stratix III, Stratix II, Stratix, Cyclone II, and Cyclone Handbooks.

Figure 1–2. Stratix III Series PLL

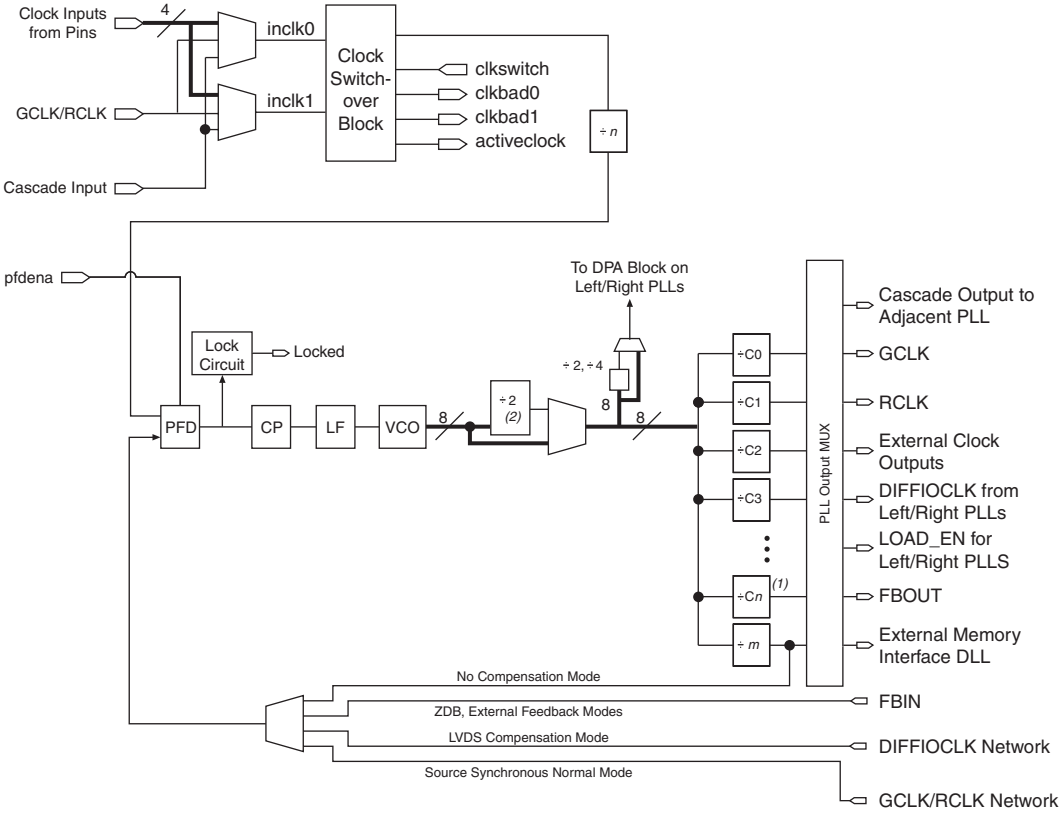


Figure 1–3. Stratix & Stratix GX Fast PLL

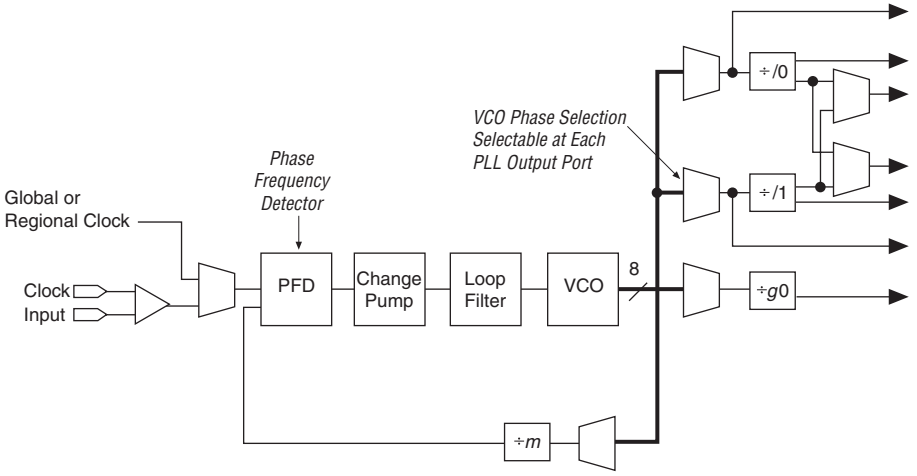
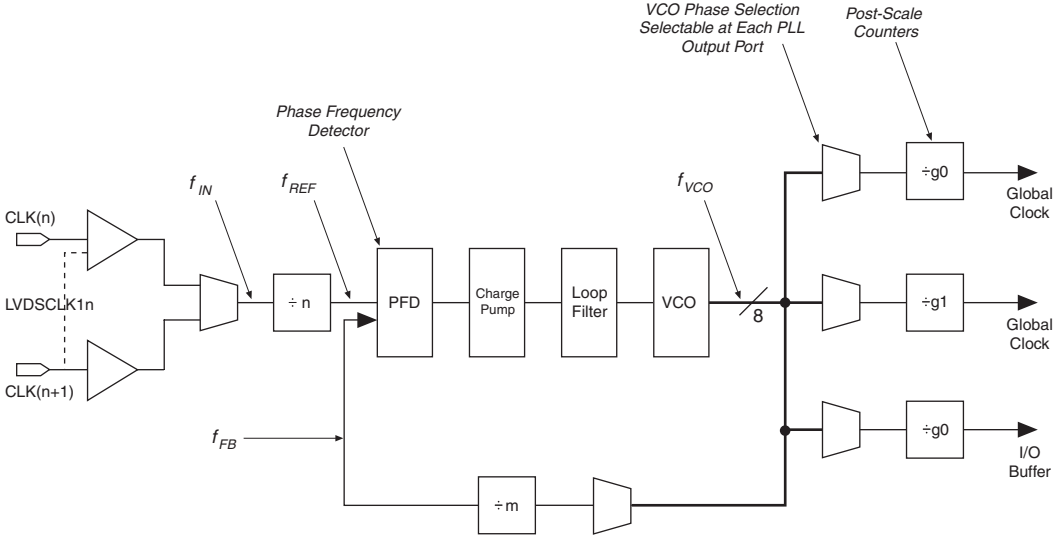


Figure 1–4. Cyclone PLL



2. Getting Started

System & Software Requirements

The instructions in this section require the following hardware and software:

- The Quartus® II software version 6.1 or higher
- Refer to the support section of the Altera® website (**www.altera.com**) for operating system support information.

MegaWizard Plug-In Manager Customization

The MegaWizard® Plug-In Manager creates or modifies design files that contain custom megafunction variations, which can then be instantiated in a design file. The MegaWizard Plug-In Manager provides a wizard that allows you to specify options for the altpll megafunction. You can use the wizard to set the altpll megafunction features in your design.

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, click **MegaWizard Plug-In Manager** in the Symbol window.
- Start the stand-alone version of the **MegaWizard Plug-In Manager** by typing the following command at the command prompt:

```
qmegawiz↵
```

Using the MegaWizard Plug-In Manager

You should use the MegaWizard Plug-in Manager to instantiate the altpll megafunction in your design. Certain altpll megafunction features are only available with Stratix® series 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 will not let you select options available on enhanced PLLs.

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.

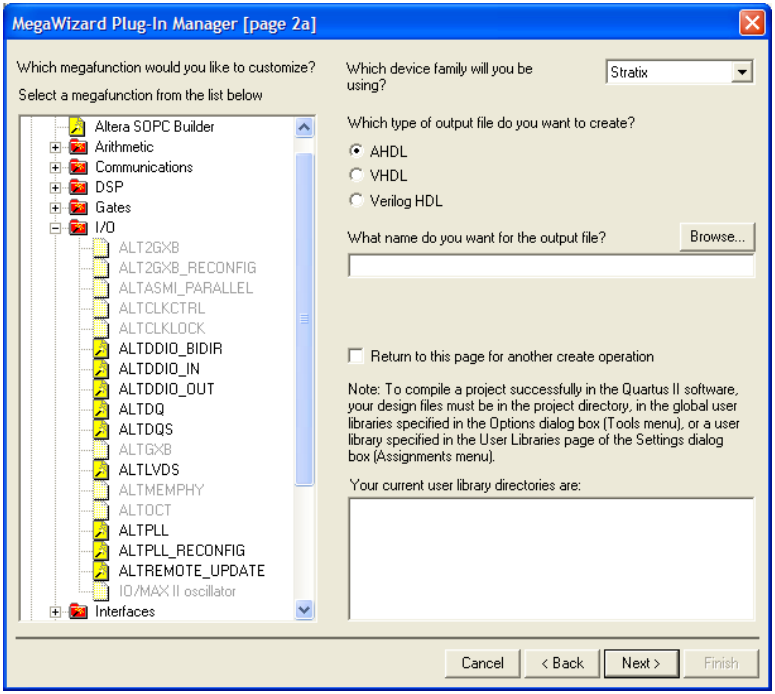
This section provides descriptions for the options available in the altpll MegaWizard Plug-In Manager pages.Tables 2–1 through 2–5 show which features or settings apply to enhanced and/or fast PLLs. Use these tables along with hardware descriptions of the fast and enhanced PLL features to determine appropriate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

The altpll Megafunction Page Descriptions (Except Stratix III Devices)

This section provides descriptions of the options available on the individual pages of the altpll MegaWizard Plug-In Manager. Note that this sub-section is valid for all devices except Stratix III devices.

Page 2a of the megafunction wizard allows the selection of the altpll megafunction from the I/O category, device selection, the type of output file to be created (Verilog HDL, VHDL, or AHDL), and the entering of the output file name (Figure 2–1). Note that there is no option available to enable clearbox netlist generation for this megafunction.

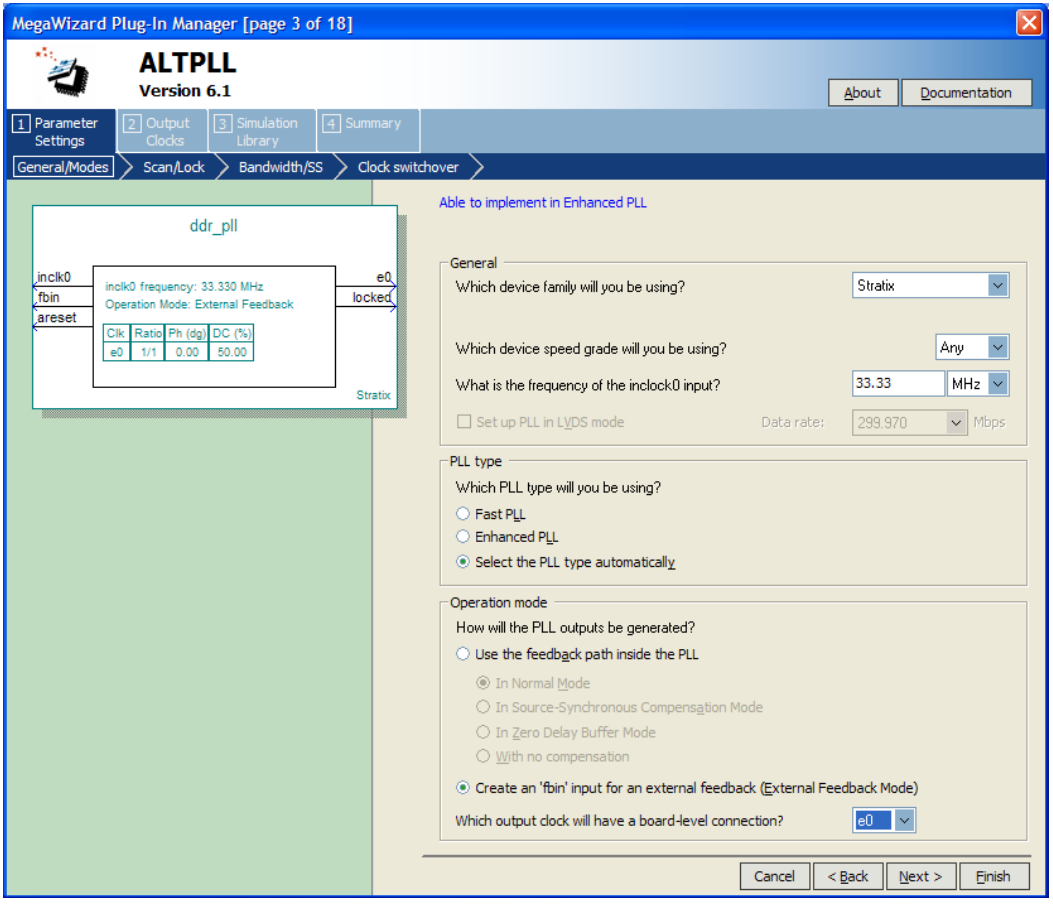
Figure 2–1. MegaWizard Plug-In Manager—altpll [Page 2a]



On page 3 of the altpll MegaWizard Plug-In Manager (Figure 2–2), you specify the device to be used, the 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.

Figure 2–2. MegaWizard Plug-In Manager —altpll [Page 3 of 18]



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

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 (Part 1 of 2)

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 use the megafunction to 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.	✓	✓
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 clock input and 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 clocks.	✓	✓ *(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.	✓	—
Which output clock is to be compensated?	Specify which output port of the PLL is compensated. For NORMAL mode, you can select `C[5..0]`. For ZERO DELAY BUFFER or EXTERNAL FEEDBACK modes, you can select `E[4..0]`. *(2)*	✓	—

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

Function	Description	Enhanced PLL	Fast PLL
Set up PLL in LVDS mode	Indicates whether this mode is used. This option, when checked, allows the PLL to supply the necessary clocking signals for the LVDS transmitter/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`.	—	✓
Data rate	Indicates whether you will use this option. This option only appears when the Setup PLL in LVDS mode is enabled. The value entered here should be used to set the `vco_multiply_by` and `vco_divide_by` parameters, since the VCO frequency corresponds 1 to 1 with the data rate. For example, if the input frequency is 100Mhz, and the data rate is 200Mbps, 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).

(3) Fast PLLs in Stratix devices support dynamic reconfiguration. However, the `scan_chain` functionality is not applicable to all device families.

On page 4 of the altpll wizard, you can enable dynamic reconfiguration on the enhanced PLLs and set the LOCK output options (Figure 2–3).

Figure 2–3. MegaWizard Plug-In Manager—altpll [page 4 of 18]

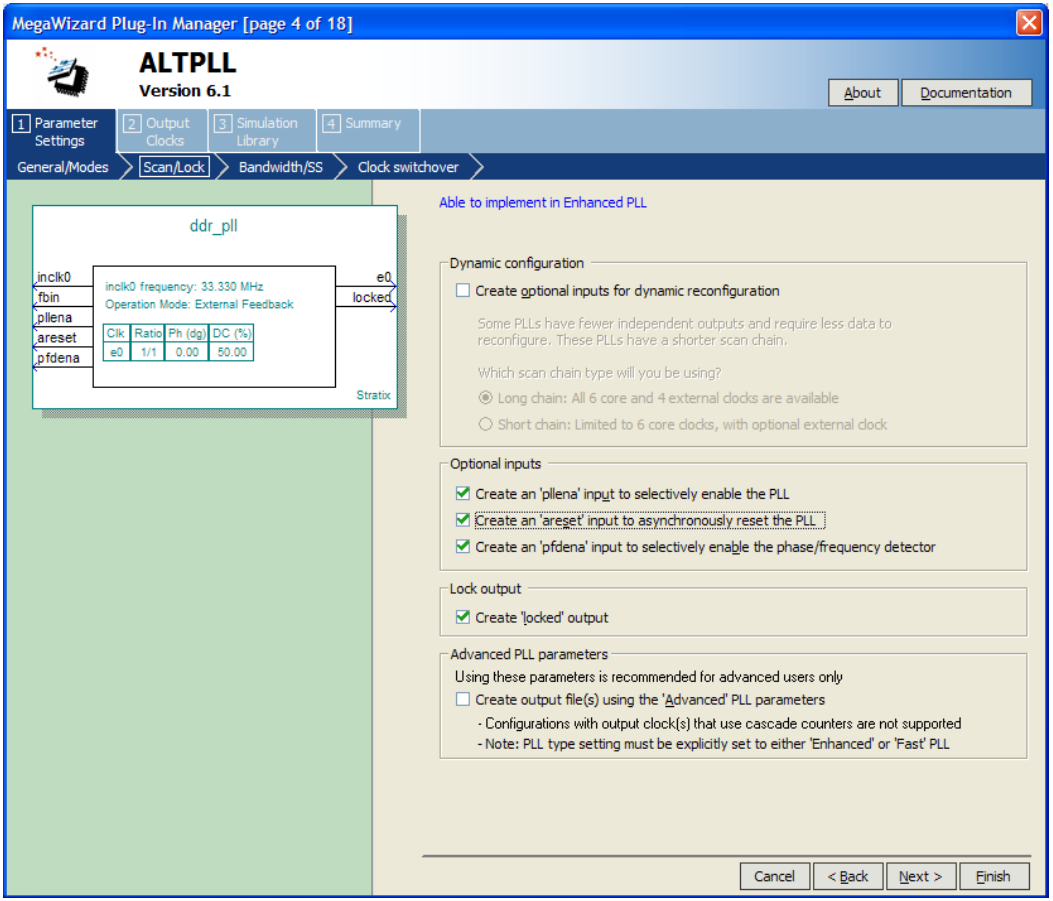


Table 2–2 shows the options and settings on page 4 of the altpll wizard.

Table 2–2. altpll MegaWizard Plug-in Manager Page 4 Options (Part 1 of 2)

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 will use 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 will use 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.	✓	✓
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.	✓	✓
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.	✓	✓
Create 'locked' output	This option creates a `locked` output port for indicating PLL lock. Refer to the `locked` port description in Table 3–2.	✓	✓

Table 2–2. altpll MegaWizard Plug-in Manager Page 4 Options (Part 2 of 2)

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 1048575) after it begins to lock.	✓	✓
Create output file using advanced 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 wizard, because after the wizard 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 wizard 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 should 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 will not be affected by this limitation.	✓	✓

On page 5 of the altpll wizard, specify the programmable bandwidth to be used and whether to take advantage of spread spectrum capabilities (Figure 2–4).

Figure 2–4. MegaWizard Plug-In Manager –altpll [Page 5 of 18]

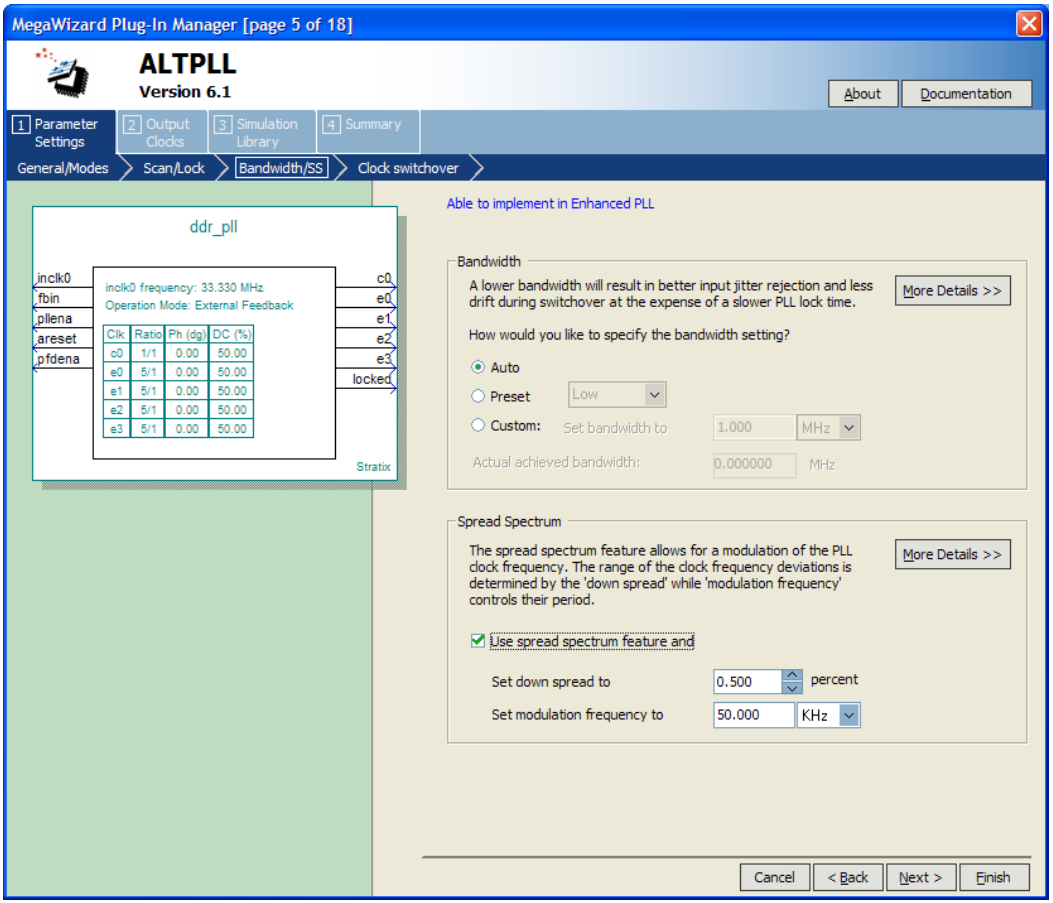


Table 2–3 shows the features and settings on page 5 of the altpll wizard.

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. For low bandwidth option, the PLL will have a better jitter rejection but slower lock time. For the high bandwidth option, it has a faster lock time but tracks more jitter. The medium option is a balance between both previous options. The compiler tries to minimize, maximize, or set the bandwidth in the middle range according to 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. The programmable bandwidth feature can be used 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. You can set the down spread from 0.4 to 0.6%. You can set the modulation frequency from 150 to 500 kHz. The spread spectrum feature can only be used in conjunction with the programmable bandwidth feature if the bandwidth feature is set to Auto.	✓	—

On page 6 of altpll wizard, you specify the options and settings for clock switchover (Figure 2–5).

Figure 2–5. MegaWizard Plug-In Manager –altpll [Page 6 of 18]

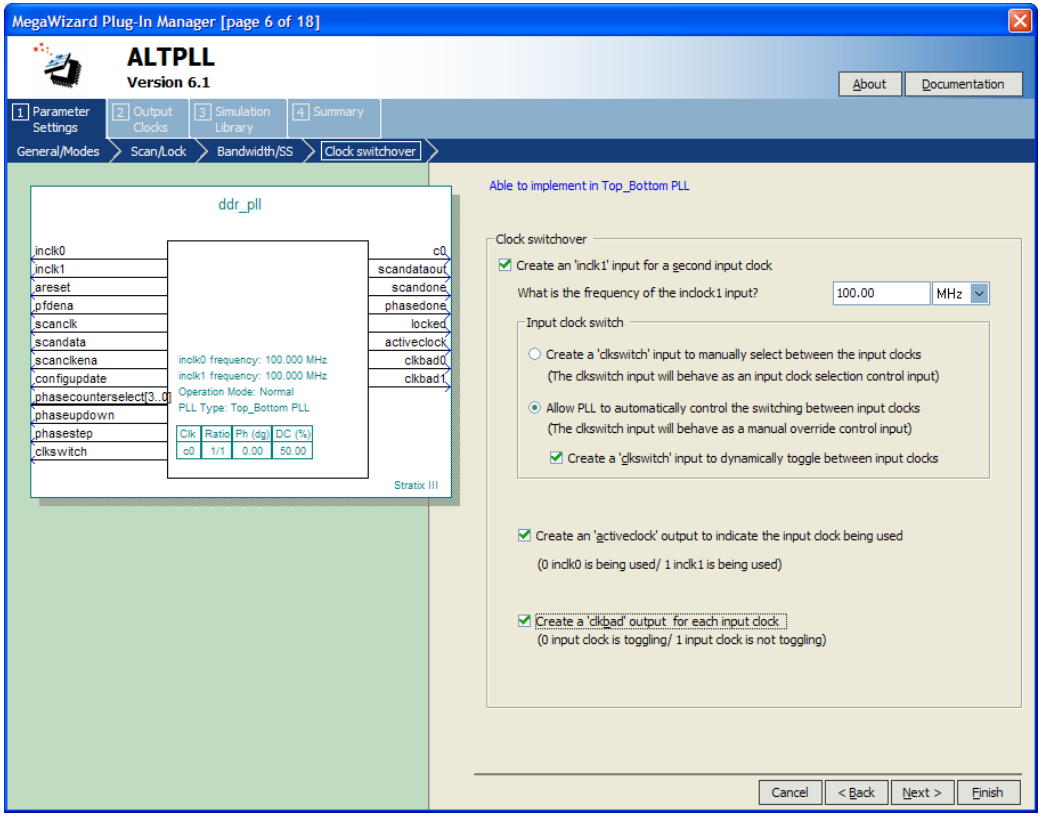


Table 2–4 shows the features and settings on page 6 of the altpll wizard.

Table 2–4. altpll MegaWizard Plug-in Manager Page 6 Options

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 wizard. 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.	✓	—
Create a 'clkswitch' input to dynamically control the switching between input clocks	Creates a control input to manually switch between input clocks of the PLL. *(1)*	✓	—
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.	✓	—
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. *(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. *(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 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 Handbook*.

On pages 7 through 16, 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 (Figure 2–6).

Figure 2–6. MegaWizard Plug-In Manager —altpll [Page 7 of 18]

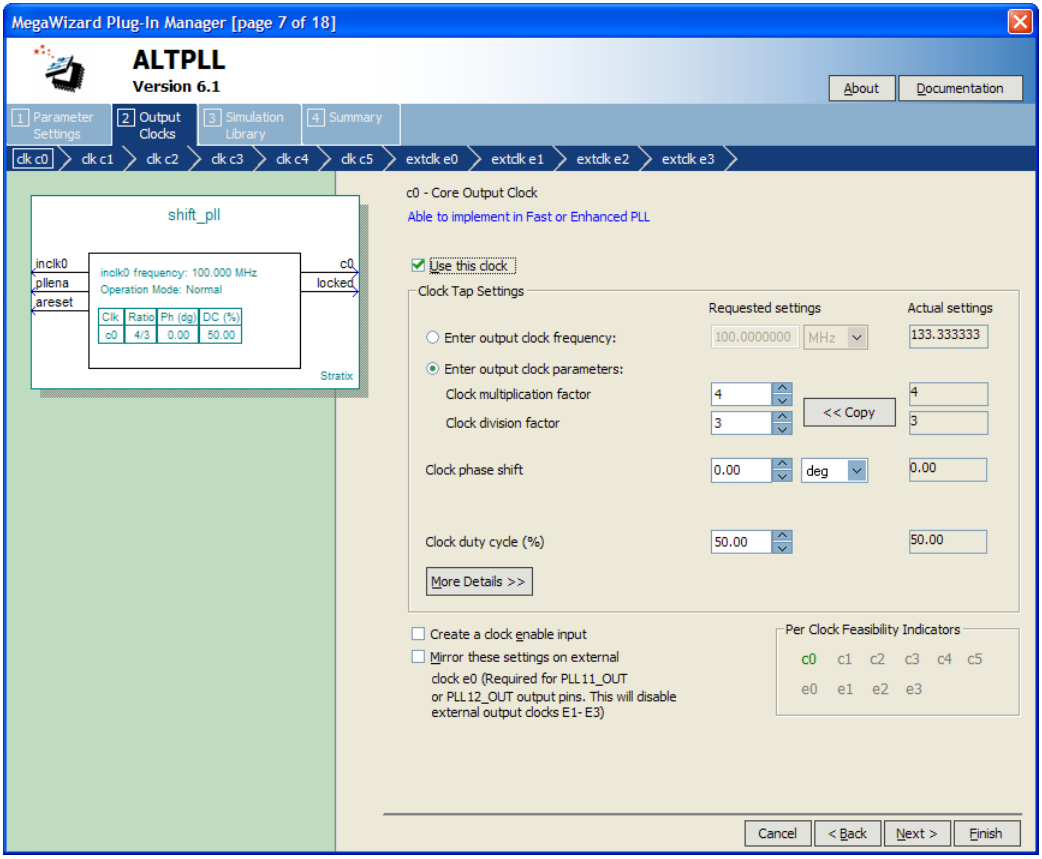


Table 2–5 shows the features and settings on pages 7–16 of the altpll wizard.

Table 2–5. altpll MegaWizard Plug-in Manager Pages 7–16 Options

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.	✓	✓
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 post-scale counter `g0` is 2, the smallest phase shift step is 22.5 . The wizard 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 wizard 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, etc.). 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 wizard will verify this and use *m* = 6, *g*0 = 6. You could enter 10 and the wizard will validate that 9 degrees 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	Specify desired output frequency. The Quartus II software determines the appropriate multiplication/division factor.	✓	✓
Create sclkout0/enable0	This option toggles the PLL to keep or not keep two new output ports, which are `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. This option, when enabled, allows the user 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 the user 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).

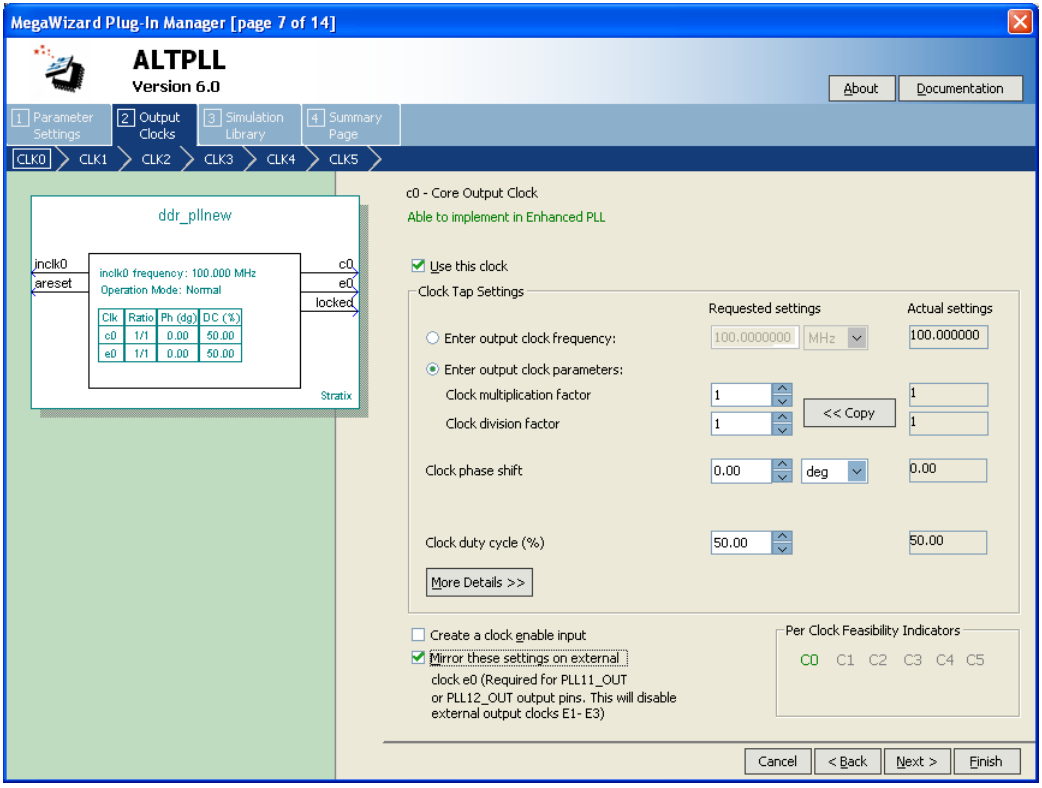
The information here is Stratix series device-specific. Enhanced PLLs 11 or 12 are short chain PLLs without external output clock counters. However, PLLs 11 and 12 support all six internal outputs (c0 through c5) and one external output driven from one of the c0 counters. To ensure the Quartus II software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

1. On Page 7 of the altpll wizard, 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 (short chain mode only)** option as shown in Figure 2–7. 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.

Figure 2–7. MegaWizard Plug-In Manager —altpll [Page 7 of 18]



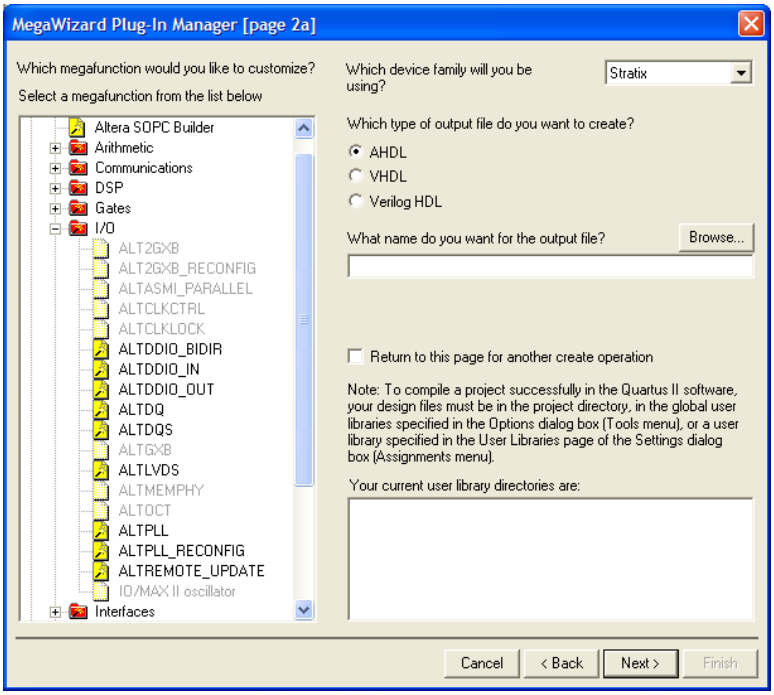
The PLLs are implemented in dedicated circuitry in Stratix series devices.

The altpll Megafunction Page Descriptions (Stratix III Devices Only)

This section provides descriptions of the options available on the individual pages of the altpll MegaWizard Plug-In Manager. This sub-section is valid for Stratix III devices only.

Page 2a of the megafunction wizard allows the selection of the altpll megafunction from the I/O category, device selection, the type of output file to be created (Verilog HDL, VHDL, or AHDL), and the entering of the output file name (Figure 2–8). Note that there is no option available to enable clearbox netlist generation for this megafunction.

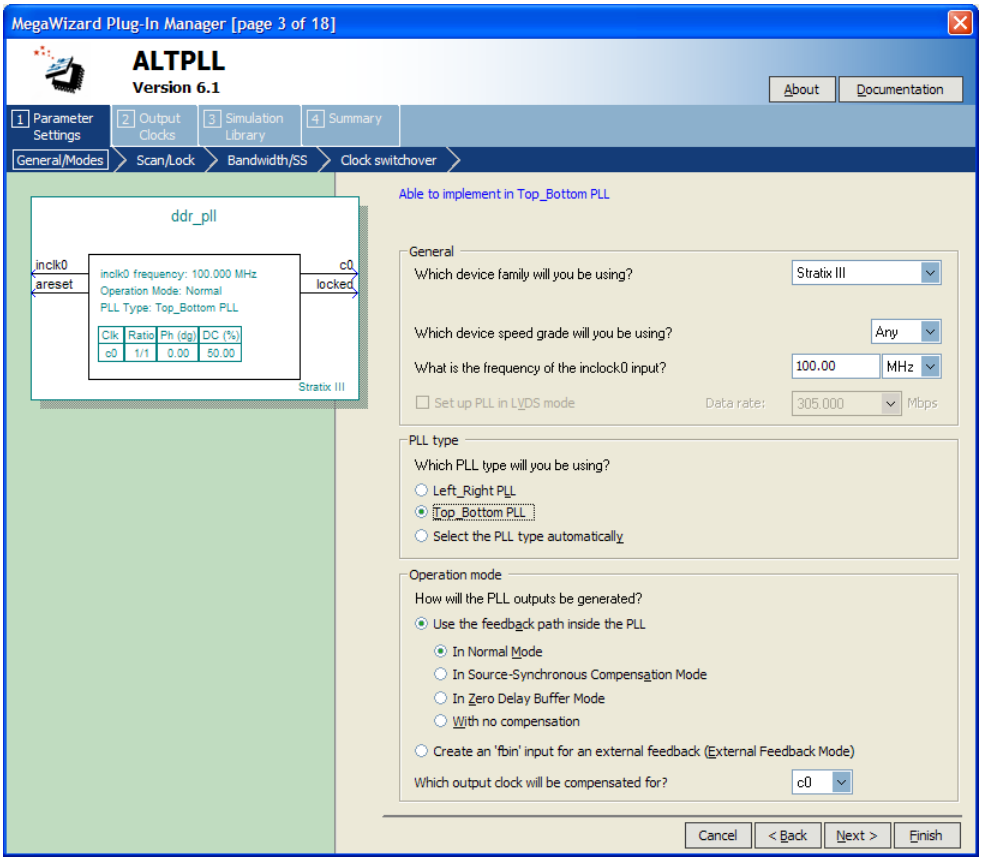
Figure 2–8. MegaWizard Plug-in Manager—altpll [Page 2a]



On page 3 of the altpll MegaWizard Plug-In Manager (Figure 2–9), you specify the device family, the speed grade (the available speeds are affected by the device selection), clock input frequency (either MHz or fractions of seconds), the 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, the outputs of the PLL are connected directly to the SERDES. For this reason, there is no need to create a special mode with extra LVDS-specific ports. This also affects the data rate, which is also not available.

Figure 2–9. MegaWizard Plug-In Manager —altpll [Page 3 of 18]



On page 3 of the altpll wizard, 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 wizard.

Table 2–6. altpll MegaWizard Plug-in Manager Page 3 Options (Part 1 of 2)

Function	Description	Top/Bottom	Left/Right
Which device family will you be using?	Select the Altera device family you are using.	Stratix III	Stratix 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 Handbook*.	Refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Handbook*.
What is the frequency of the inclock0 input?	Indicate the input frequency for the `inclock0` input of the PLL.	For input frequency range, refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Handbook*.	For input frequency range, refer to the *DC & Switching Characteristics of Stratix III Devices* chapter in the *Stratix III Handbook*.
Set up PLL in LVDS mode	This option is not available in Stratix III devices.	—	—
Which PLL type will you be using?	Indicate whether you will use the megafunction to use a top/bottom PLL, left/right PLL, or 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 (Part 2 of 2)

Function	Description	Top/Bottom	Left/Right
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 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 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 clock network. There is no clock network compensation, but this mode minimizes jitter on clocks.	All of the compensation modes specified here are available to this type of PLL.	All of the compensation modes specified here are available to this type of PLL.
Create an 'fbin' input for an external feedback (External Feedback Mode)	**External Feedback mode**—the PLL compensates for the fBIN feedback input into the PLL. The delay between the input clock pin and the feedback clock pin is minimized.	This option is available for this type of PLL.	This option is available for this type of PLL except that it is only for single-ended I/O standards.
Which output clock will have a board-level connection?	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.
Data rate	This option is not available in Stratix III devices.	—	—

On page 4 of the altpll wizard, settings for enable dynamic reconfiguration, dynamic phase reconfiguration, optional inputs for asynchronous reset and phase/frequency detector, the LOCKED output options and advanced PLL parameter are all adjustable (Figure 2–10).

Figure 2–10. MegaWizard Plug-In Manager —altpll [Page 4 of 18]

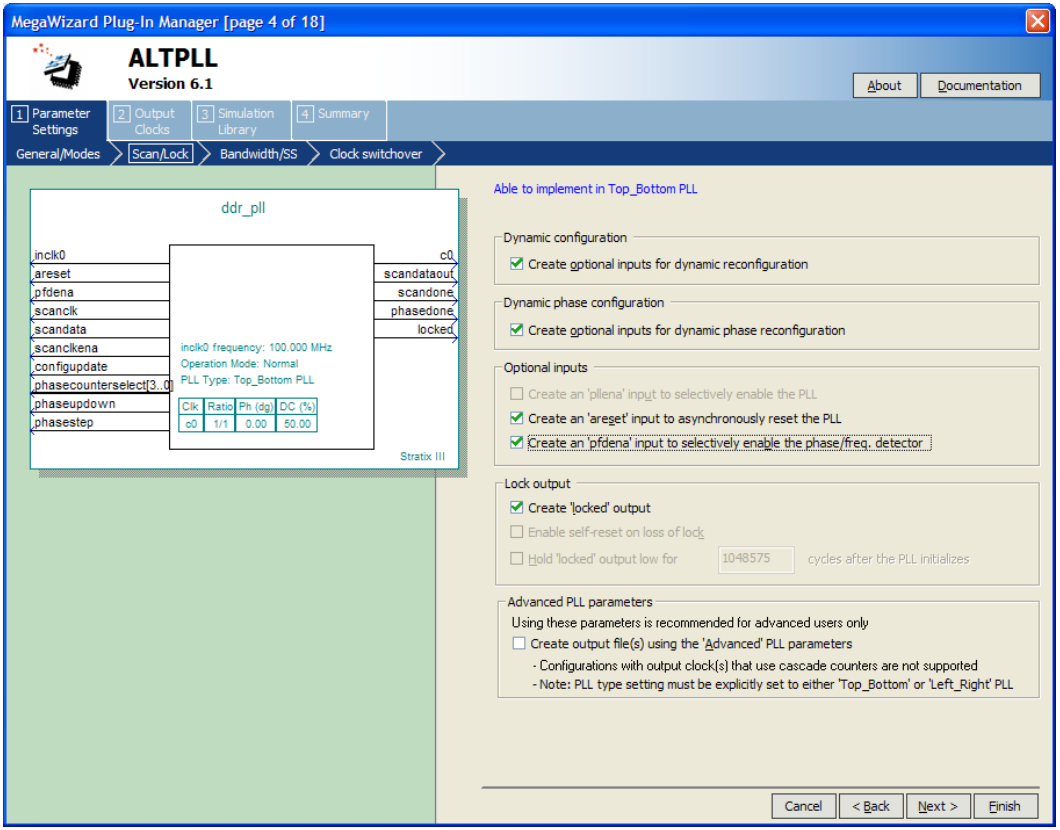


Table 2–7 shows the options and settings on page 4 of the altpll wizard.

Table 2–7. altpll MegaWizard Plug-in Manager Page 4 Options (Part 1 of 2)

Function	Description	Top/Bottom	Left/Right
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
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
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 3–3.	Not available for Stratix III devices	Not available for Stratix III devices
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 3–3.	Available for this type of PLL	Available for this type of PLL

Table 2–7. altpll MegaWizard Plug-in Manager Page 4 Options (Part 2 of 2)

Function	Description	Top/Bottom	Left/Right
Create a '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 3–3.	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 wizard, because after the wizard 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 wizard 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 should 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 will not be affected by this limitation.	Available for this type of PLL	Available for this type of PLL

On page 5 of the altpll wizard, specify the programmable bandwidth to be used. Customizing spread spectrum capabilities are not available for Stratix III devices (Figure 2–11).

Figure 2–11. MegaWizard Plug-In Manager—altpll [Page 5 of 18]

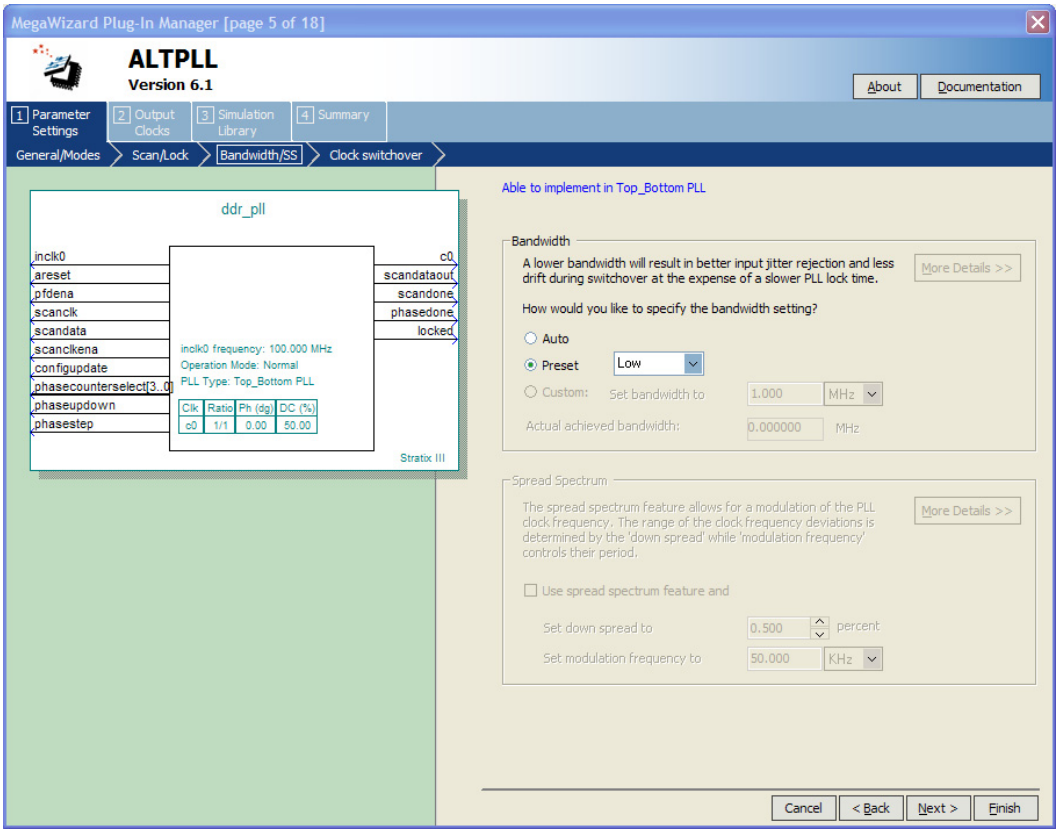


Table 2–8 shows the features and settings on page 5 of the altpll wizard.

Table 2–8. altpll MegaWizard Plug-In Manager Page 5 Options

Function	Description	Top/Bottom	Left/Right

How would you like to specify the bandwidth?	• **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 previous options; the compiler tries to minimize, maximize, or set the bandwidth in the middle range according to 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 altpll wizard, specify the options and settings for clock switchover (Figure 2–12).

Figure 2–12. MegaWizard Plug-In Manager —altpll [Page 6 of 18]

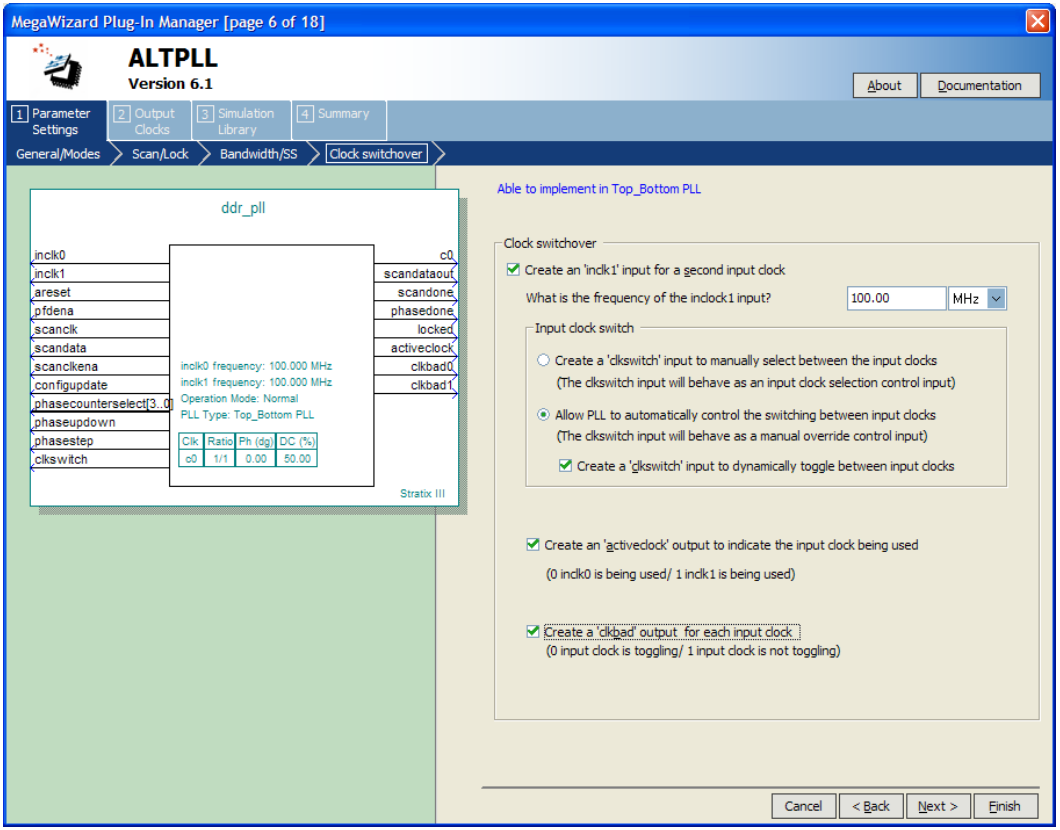


Table 2–9 shows the features and settings on page 6 of the altpll wizard.

Table 2–9. altpll MegaWizard Plug-in Manager Page 6 Options

Function	Description	Top/Bottom (such as Enhanced) PLL	Left/Right (such as 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 wizard. 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
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
Create an 'activeclock' output to indicate which input clock is 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 3–5.	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 3–5.	Available for this type of PLL	Available for this type of PLL

On pages 7 to 18, specify the multiplication, division, duty cycle, phase shift, and time shift for each PLL output port (c0 through c9). Each page represents the settings for one PLL output port (Figure 2–13).

Figure 2–13. MegaWizard Plug-In Manager—altpll [Page 7 of 18]

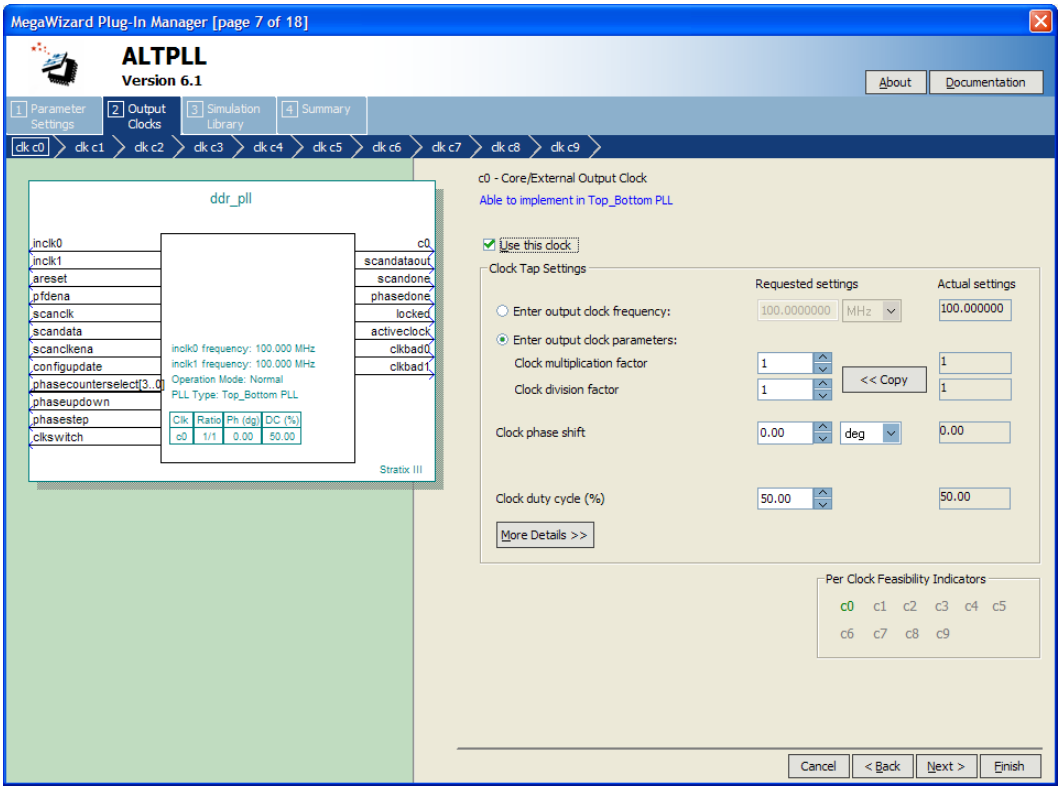


Table 2–10 shows the features and settings on pages 7 through 18 of the altpll wizard.

Table 2–10. altpll MegaWizard Plug-in Manager Page 7 through 18Options

Function	Description	Top/Bottom	Left/Right
Enter output clock frequency	Use Table 2–5 on page 2–15 as reference.	Has ten available output clocks where this option can be set.	Has seven available output clocks where this option can be set.
Clock multiplication/ division factor	Use Table 2–5 on page 2–15 as reference.	Has ten available output clocks where this option can be set.	Has seven available output clocks where this option can be set.
Clock phase shift	Use Table 2–5 on page 2–15 as reference.	Has ten available output clocks where this option can be set.	Has seven available output clocks where this option can be set.
Clock duty cycle	Use Table 2–5 on page 2–15 as reference.	Has ten available output clocks where this option can be set.	Has seven available output clocks where this option can be set.

Inferring Megafunctions from HDL Code

Synthesis tools, including Quartus II integrated synthesis, recognize certain types of HDL code and automatically infer the appropriate megafunction when a megafunction will provide optimal results. However, altpll cannot be inferred and must be instantiated in your design. Refer to "Instantiating Megafunctions in HDL Code" for details about instantiating megafunctions.

Instantiating Megafunctions in HDL Code

When you use the MegaWizard Plug-In Manager to set up and parameterize a megafunction, the wizard creates either a VHDL or Verilog HDL wrapper file that instantiates the megafunction (a black-box methodology). For some megafunctions, you can generate a fully synthesizable netlist for improved results with EDA synthesis tools, such as Synplify and Precision RTL Synthesis (a clear-box methodology).

Both clear- and black-box methodologies are described in the Synthesis section of volume 1 of the *Quartus II Handbook*.

Identifying a Megafunction after Compilation

During compilation with the Quartus II software, analysis and elaboration is performed to build the structure of your design. Locate your megafunction in the Project Navigator window by expanding the compilation hierarchy and locate the megafunction by its name.

You can search for node names within the megafunction (using the Node Finder), or in the **Hierarchy** box, browse to and locate the megafunction.

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} is provided. Perform a List Path to view the various timing parameters (for example, the 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 during compilation reports that the project clock settings override the PLL clock settings.
- The project clock setting overrides the PLL clock settings for timing-driven compilation. When you compile a design with timing-driven compilation turned on, you are overconstraining the design so that the fitter can give you a better f_{MAX} performance. For example, if the PLL is set to output a 150-MHz clock, you can set a project clock setting for 170 MHz so 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 will not be overridden.
- Overriding the PLL clock settings changes only the timing requirements; it does not change the overall multiplication/division and phase delay on each clock output of the PLL. The MegaWizard Plug-In Manager does not use the project clock settings to determine the altpll parameters.
- A Default Required f_{MAX} setting does not override the PLL clock settings. Only individual clock settings will override the PLL clock settings.

Overriding the 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 **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** window.
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

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 lock feature.
External feedback	Modeled *(1)*.
PFD Enable	Modeled. The finite frequency drift of V_{CO} is not modeled if phase frequency detector (PFD) is disabled.
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 will check that (f_{IN} x *m*)/*n* is within the V_{CO} range and will lock as if configured for high bandwidth.
Spread Spectrum	Frequency modulation is not modeled in simulation.
Jitter	Jitter is not modeled in simulation.
`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) Refer to "Simulating External Feedback Board Delay in Stratix II & Stratix II GX Devices" for information about external feedback simulation.

Simulating External Feedback Board Delay in Stratix II & 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 the 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.

You can 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 should follow the same guidelines and restrictions as the design entry. The altpll behavioral and timing models do not simulate jitter.

The behavioral models for altpll reside in the **\quartus\eda\sim_lib** directory. The **altera_mf.vhd** file contains the VHDL behavioral models and can be used for Stratix II/Stratix II GX altpll. 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, and 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 will round off the calculations, providing incorrect multiplication or division.

Quartus II Simulation

With the Quartus II Simulator, you can perform two types of simulations: functional and timing.

A functional simulation in the Quartus II program enables you to verify the logical operation of your design without taking into consideration the timing delays in the FPGA. This simulation is performed using only RTL code. When performing a functional simulation, add only signals that exist before synthesis. You can find these signals with the Registers: pre-synthesis, Design Entry, or Pin filters in the Node Finder. The top-level ports of megafunctions are found using these three filters.

In contrast, timing simulation in the Quartus II software verifies the operation of your design with annotated timing information. This simulation is performed using the post place-and-route netlist. When performing a timing simulation, add only signals that exist after place-and-route. These signals are found with the Post-Compilation filter of the Node Finder. During synthesis and place-and-route, the names of RTL signals change. Therefore, it might be difficult to find signals from megafunction instantiation in the Post-Compilation filter. However, if you want to preserve the names of your signals during the synthesis and place-and-route stages, you must use the synthesis attributes `keep` or `preserve`. These are Verilog and VHDL synthesis attributes that direct analysis and synthesis to keep a particular wire, register, or node intact. Use these synthesis attributes to keep a combinational logic node so you can observe the node during simulation.

More information about these attributes is available in volume 1 of the *Quartus II Handbook*.

EDA Simulation

Depending on which simulation tool you use, refer to the appropriate chapter in the Simulation section in volume 3 of the *Quartus II Handbook*. The *Quartus II Handbook* chapters show you how to perform functional- and gate-level timing simulations that include the megafunctions, with details about the files that are needed and the directories where those files are located.

The behavioral simulation models for Verilog HDL simulations are located in: **<*Quartus_install_directory*>/eda/lib/altera_mf.v**.

The behavioral simulation models for VHDL simulations are located in: **<*Quartus_install_directory*>/eda/lib/altera_mf.vhd**.

Reporting

A compilation information message displays whether the requested multiplication and division factors and/or the requested phase shift were successful. If you enter an invalid multiplication and/or division factor, compilation fails, and the Quartus II software displays an error message with alternative factors. If you enter invalid phase shift values, the compilation proceeds, and you will see an information message displaying the best alternative values, which the software proceeds to use.

The resource section of the compilation report provides two PLL reports and the Quartus II software displays the PLL summary and the PLL usage reports. The PLL Summary provides information about each PLL instance parameter. Table 2–12 describes the parameters shown in the PLL Summary. The PLL Summary is column-based; each column represents a different PLL.

Table 2–12. PLL Summary in Report File (Part 1 of 2)

Parameter	Definition
PLL type	Indicates whether it is an enhanced or fast PLL
Scan chain	Indicates whether it is a long reconfiguration chain (PLL 5 or 6) or a short reconfiguration chain (PLL 11 or 12)
PLL mode	Indicates feedback mode
Feedback source	Indicates which external output has a board level connection to fbin
Compensate clock	Indicates which clock output port (internal or external) should be compensated for
Switchover on loss of clock	On or off
Switchover counter	Shows the value of the switchover delay counter
Primary clock	Shows which input, `inclk0` or `inclk1`, is the primary clock for switchover
Input frequency 0	Clock input frequency for `inclk0`
Input frequency 1	Clock input frequency for `inclk1`
Nominal VCO frequency	Shows the V_{CO} frequency, or (input frequency *m*)/*n*
Freq min lock	Shows the minimum input frequency for which the current combination of *m*/*n* still provides a valid V_{CO} lock
Freq max lock	Shows the minimum input frequency for which the current combination of *m* and *n* still provides a valid V_{CO} lock
Clock Offset	Shows the clock offset value
MVCO Tap	Shows the V_{CO} tap value for the *m* counter
m Initial	Shows the number of initial V_{CO} cycles before the *m* counter starts
m value	*m* counter value
n value	*n* counter value
m counter delay	Time delay setting on *m* counter to provide negative shift for all PLL outputs

Table 2–12. PLL Summary in Report File (Part 2 of 2)

Parameter	Definition
n counter delay	Time delay setting on *n* counter to provide positive shift for all PLL outputs
*m*2 value	*m*2 counter value (for spread-spectrum modulation)
*n*2 value	*n*2 counter value (for spread-spectrum modulation)
SS counter	Spread-spectrum counter setting (controls frequency of modulation for SS)
Downspread	Downspread setting
Spread frequency	Spread frequency (should be input frequency divided by SS counter)
Charge pump current	Charge pump current setting
Loop filter resistance	Loop filter resistor value
Loop filter capacitance	Loop filter capacitance value
Freq zero	Loop filter zero location in the frequency domain
Bandwidth	Bandwidth of this PLL
Freq pole	Loop filter pole location in the frequency domain
Enable 0 counter	For `RXLOADEN` or `TXLOADEN` in fast PLL (does not apply in general purpose mode)
Enable 1 counter	For `RXLOADEN` or `TXLOADEN` in fast PLL (does not apply in general purpose mode)
Real time reconfigurable	On or off
Scan chain **.mif** file	Points to the **.mif** file with the initial configuration of the PLL counters.
Preserve counter order	On or off
PLL Location	The location of the PLL
Inclock0 signal	Name of the pin driving the `inclk0` port of the PLL
Inclock1 signal	Name of the pin driving the `inclk1` port of the PLL

The PLL usage report shows detailed information for each PLL output. This report is categorized by PLL output ports. Each row represents a different PLL output port used in the design. Table 2–13 lists the parameters shown in the PLL usage. In the report file, this information is shown in a row format as opposed to the column format shown in Table 2–13.

Table 2–13. PLL Usage Summary in Report File (Part 1 of 2)

Parameter	Definition
PLL type	Indicates whether it is an enhanced or fast PLL
Output clock	Indicates the PLL output `c0` through `c5` or `e0` through `e3` for which the parameter information in this row applies to

Table 2–13. PLL Usage Summary in Report File (Part 2 of 2)

Parameter	Definition
Mult	Overall multiplication ratio
Div	Overall division ratio
Output frequency	Output frequency for this row's output clock
Phase shift	Achieved phase shift (may differ from user entered value)
Delay	Overall delay setting on this clock output
Duty cycle	Duty cycle for this clock output
Counter	Post-scale counter used for this clock output
Counter delay	Delay for the output counter (combined with *m* and *n* delays gives overall delay)
Counter value	Value for post-scale counter
High/low	High and low time counts that make up the counter value. The ratio of high and low counts is directly proportional to duty cycle.
Initial	Initial value for this post-scale counter (achieves the coarse granularity for phase shift)
VCO tap	V_{CO} tap ranges from 0 to 7 (achieves fine granularity for phase shift in units of 1/8 of a V_{CO} period)

Calculate the Clock Cycles to Gate the Lock Signal

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

The 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.

SignalTap II Embedded Logic Analyzer

The SignalTap II embedded logic analyzer provides you with a non-intrusive method to debug all of the Altera megafunctions within your design. With the SignalTap II embedded logic analyzer, you can capture and analyze data samples for the top-level ports of the Altera megafunctions in your design while your system is running at full speed.

To monitor signals from your Altera megafunctions, you must first configure the SignalTap II embedded logic analyzer in the Quartus II software, and then include the analyzer as part of your Quartus II project. The Quartus II software then embeds the analyzer seamlessly with your design in the selected device.

For more information about using the SignalTap II embedded logic analyzer, refer to the *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

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 2–20 on page 2–46) and generate and modify internal clock signals (as shown in Figure 2–29 on page 2–58).

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 example design files are available in the Quartus II projects section under the Design Examples page of the Altera web site.

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 a 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 the enhanced PLLs in Stratix devices to generate these clock signals.

In this example, you 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**. 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.

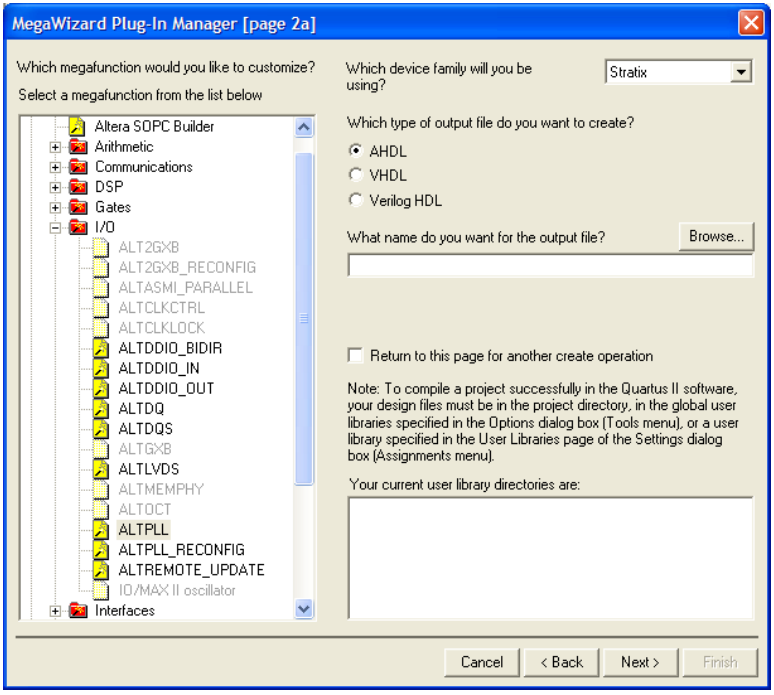
5. On Page 2a of the wizard, expand the **I/O** folder and select **ALTPLL**.

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**.

   Figure 2–14 shows page 2a after these parameters are set.

Figure 2–14. MegaWizard Plug-In Manager —altpll [Page 2a]

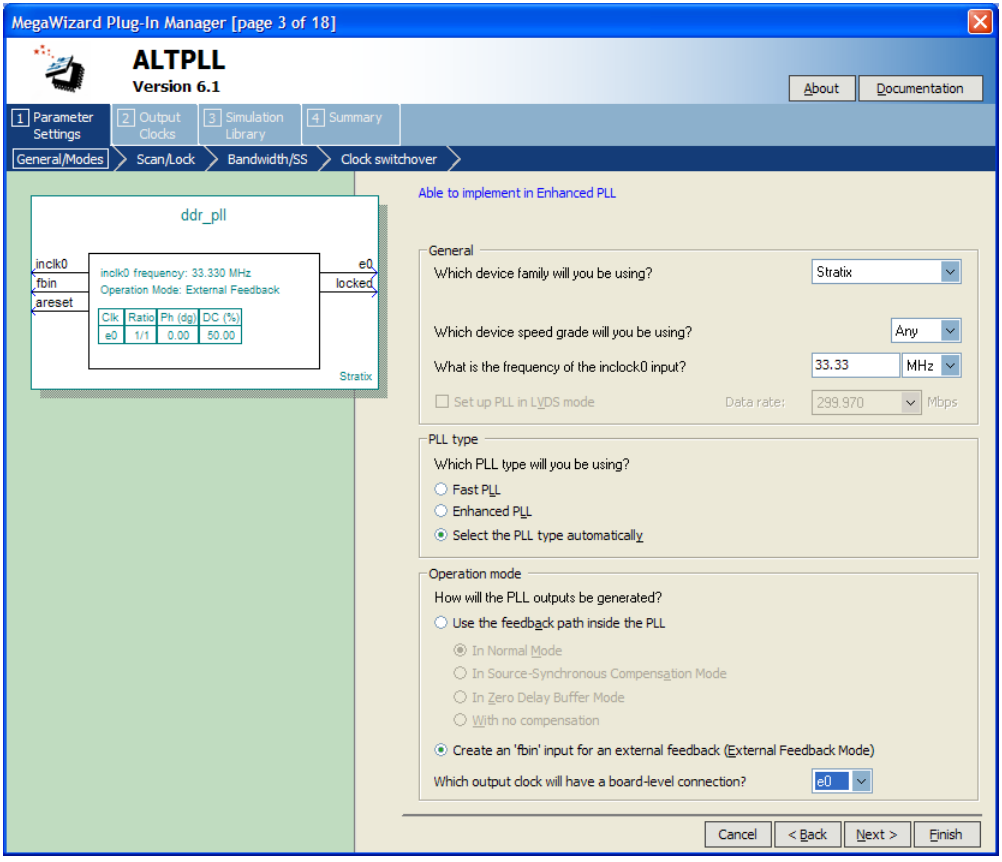


8. For **What name do you want for the output file?**, name the output file **ddr_pll**.

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.

    Figure 2–15 shows page 3 after you have set these parameters.

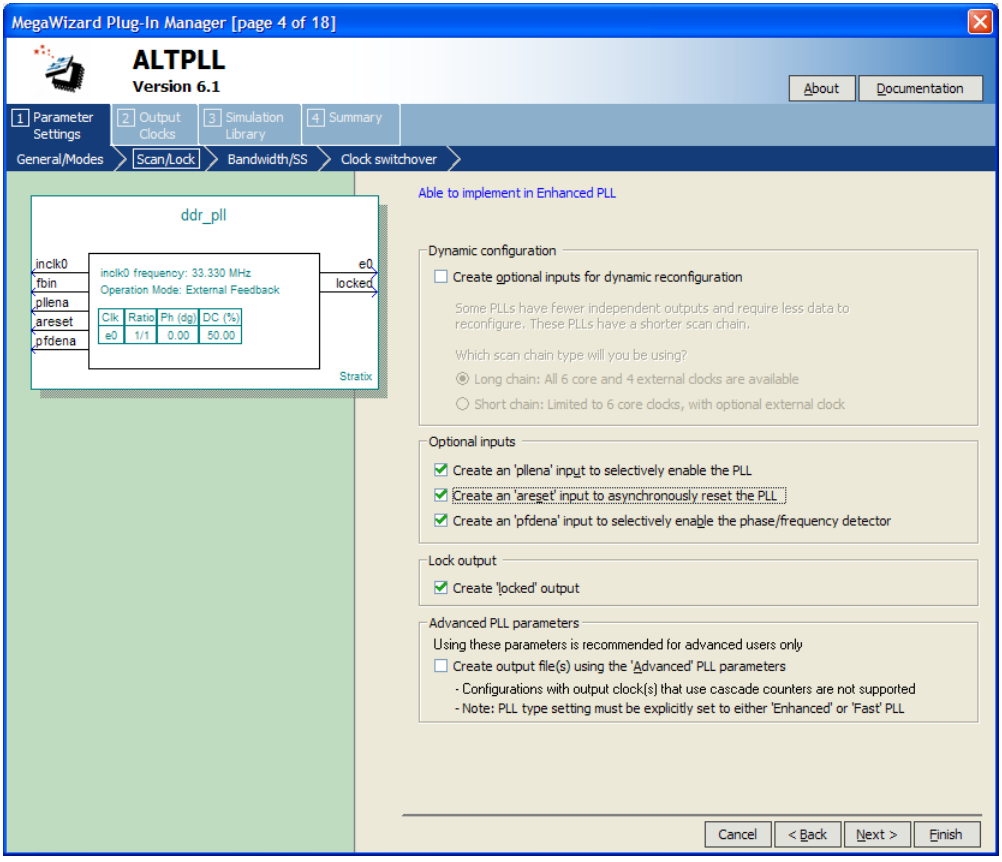
Figure 2–15. MegaWizard Plug-In Manager—altpll [Page 3 of 18]



14. Click **Next**. Page 4 appears.
15. In the **Dynamic configuration** section, leave the default settings.
16. In the **Optional inputs** section, turn on **Create an 'pllena' input to selectively enable the PLL**, **Create an 'areset' input to asynchronously reset the PLL**, and **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.

    Figure 2–16 shows page 4 after you have made these selections.

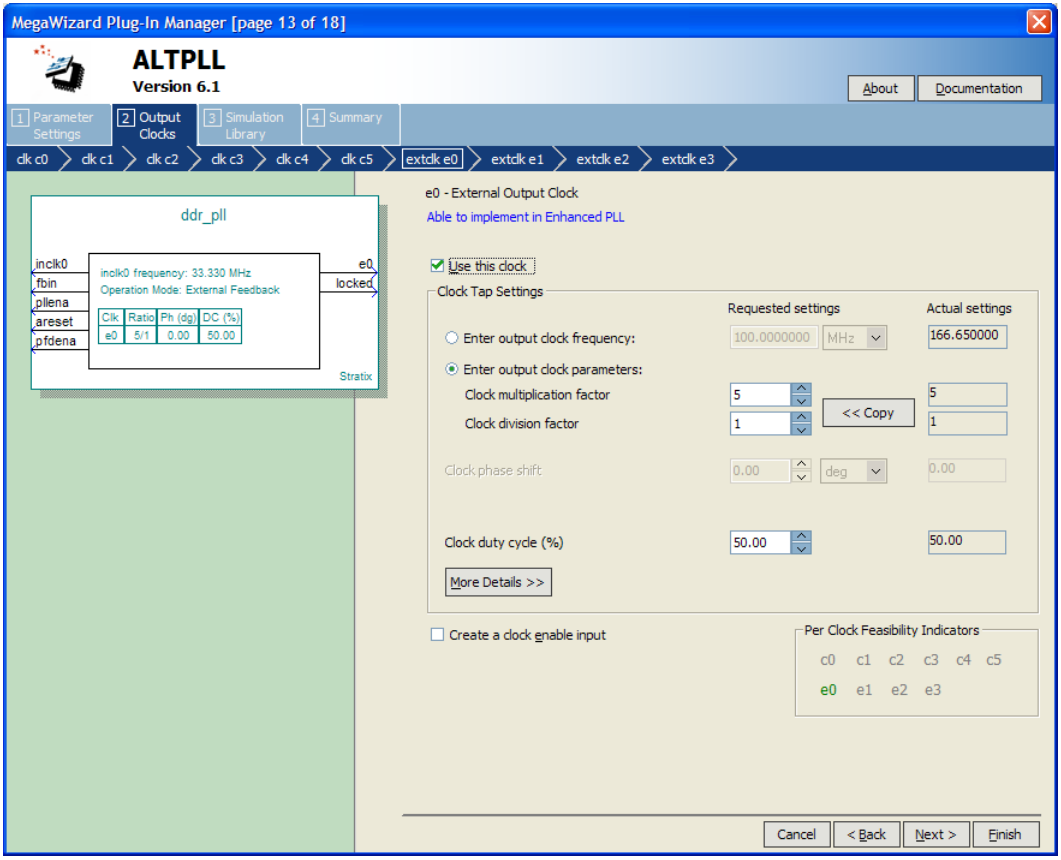
Figure 2–16. MegaWizard Plug-In Manager —altpll [Page 4 of 18]



19. Click on 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`.

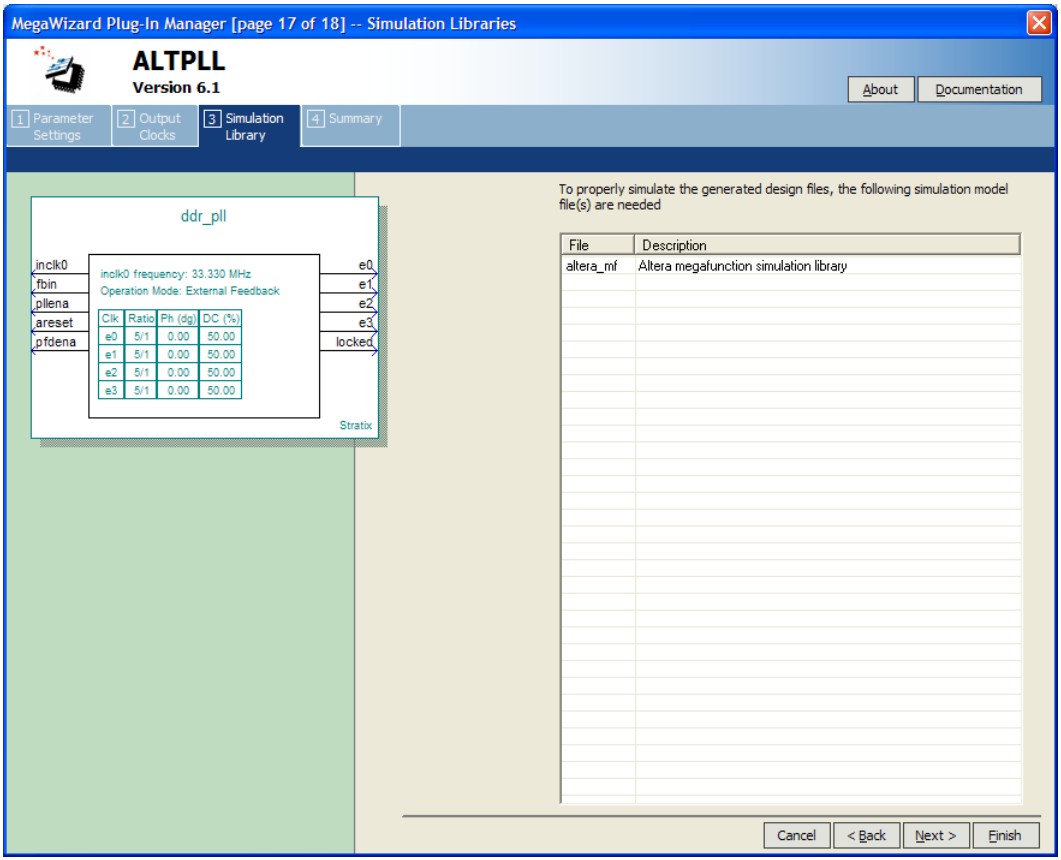
    Figure 2–17 shows page 13 after you have made these settings.

Figure 2–17. MegaWizard Plug-In Manager—altpll [Page 13 of 18]



25. Click **Next**. Page 14 appears.
26. On page 14, repeat steps 20 through 24 for **extclk e1**.
27. Click **Next**.
28. On Page 15, repeat steps 20 through 24 for **extclk e2**.
29. Click **Next**.
30. On Page 16, repeat steps 20 through 24 for **extclk e3**.
31. Click **Next**. Page 17 appears (Figure 2–18). No input is required for this page.

Figure 2–18. MegaWizard Plug-In Manager—altpll [Page 17 of 18]

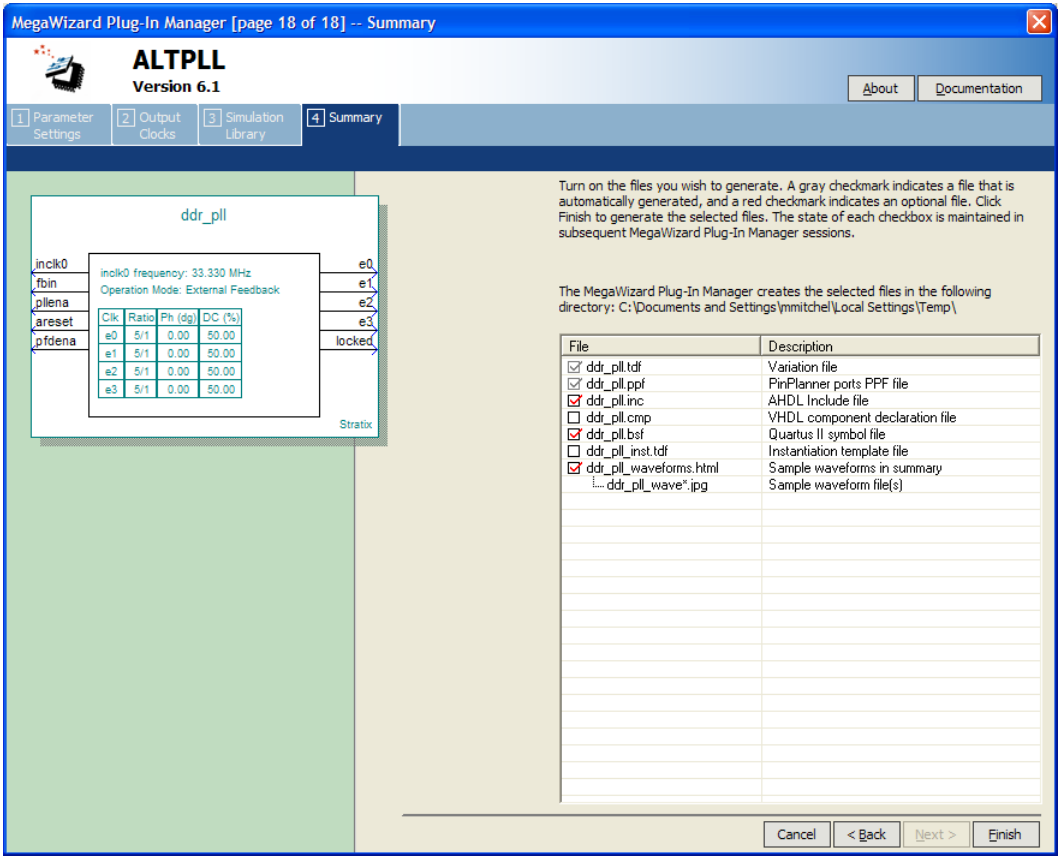


32. Click **Next**. Page 18 appears.

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.

    Figure 2–19 shows page 18 after you have made these selections.

Figure 2–19. MegaWizard Plug-In Manager—altpll [Page 18 of 18]



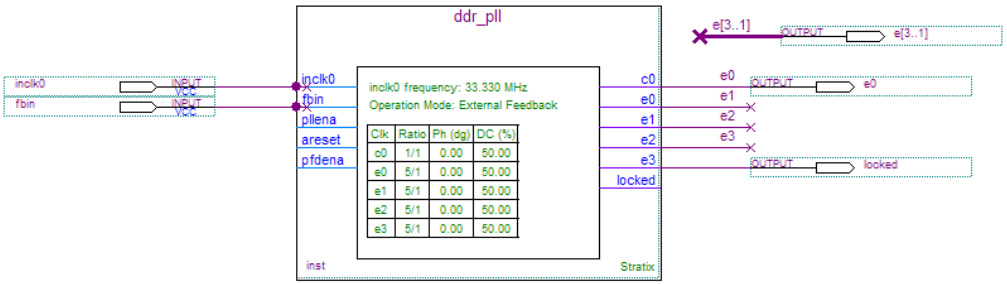
34. Click **Finish**. The **ddr_pll** module is built.

35. In the Symbol window 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**, connecting the inputs and outputs to the symbol. Click to place the symbol.

    You have now completed the design file as shown in Figure 2–20.

Figure 2–20. 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**.

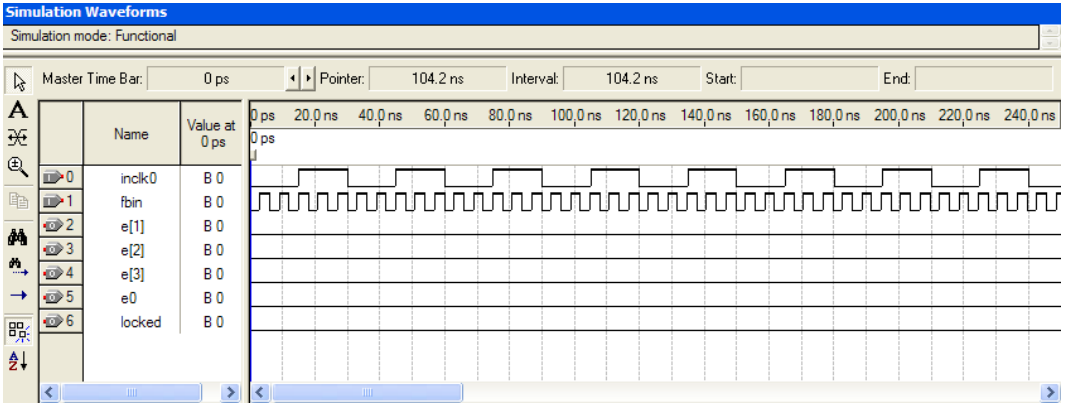
The `ddr_clk` design is now implemented.

Functional Results—Simulate the ddr_clk Design in Quartus

In this section you will simulate the design to verify the results. Set up the Quartus II Simulator by performing the following steps:

1. On the Processing menu, click **Generate Functional Simulation Netlist**.
2. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.
3. On the Assignments menu, click **Settings**.
4. In the **Category** list, click **Simulator Settings**.
5. From the **Simulation mode** list, select **Functional**.
6. In the **Simulation input** field, browse to select the simulation input file **ddr_pll.vwf**.
7. Under Simulation period, select **Run simulation until all vector stimuli are used**.
8. Click **OK**.
9. Click **Start**, or on the Processing menu, click **Start Simulation**.
10. When the **Simulation was successful** message box appears, click **OK**.
11. In the **Simulation Report** window, verify the results of the simulation output waveform. Figure 2–21 shows the expected simulation results.

Figure 2–21. Functional Waveform for ddr_clk Design



Functional Results—Simulate the ddr_clk Design in ModelSim-Altera

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

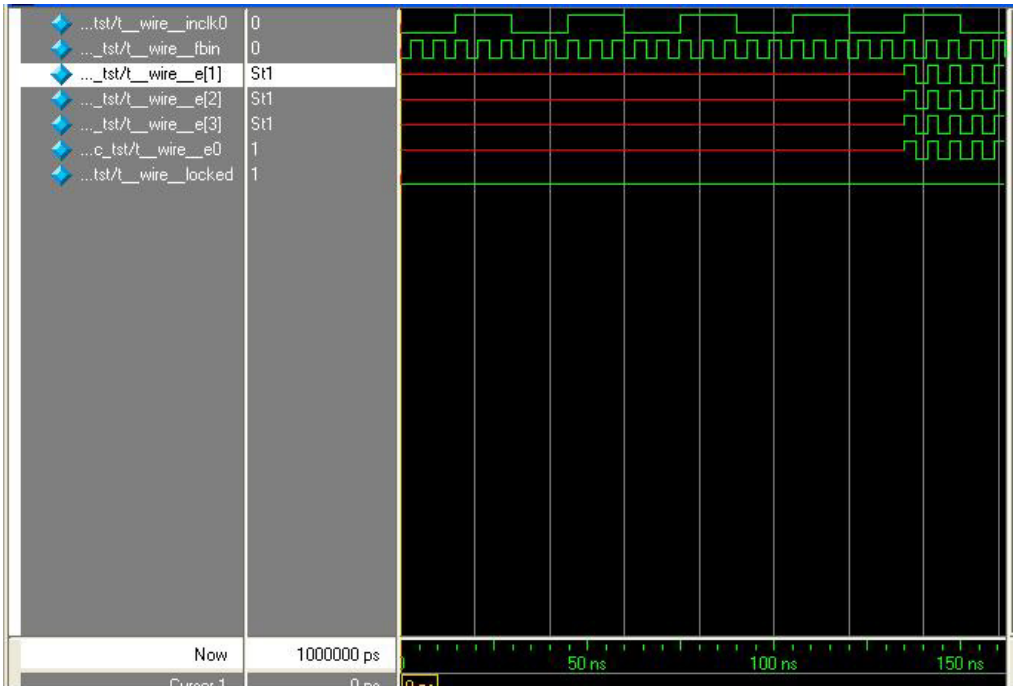
This User Guide assumes that you are familiar with using ModelSim-Altera before trying out the design example. If you are unfamiliar with ModelSim-Altera, refer to the support page for ModelSim-Altera on the Altera web site. There are links to various topics, including installation, usage, and troubleshooting.

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 **ModelSim-Altera**.
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 **DDR_CLK.do**, and click **Open**. This is a script file for ModelSim that automates 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 2–22 shows the expected simulation results in ModelSim.

Figure 2–22. 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, you 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

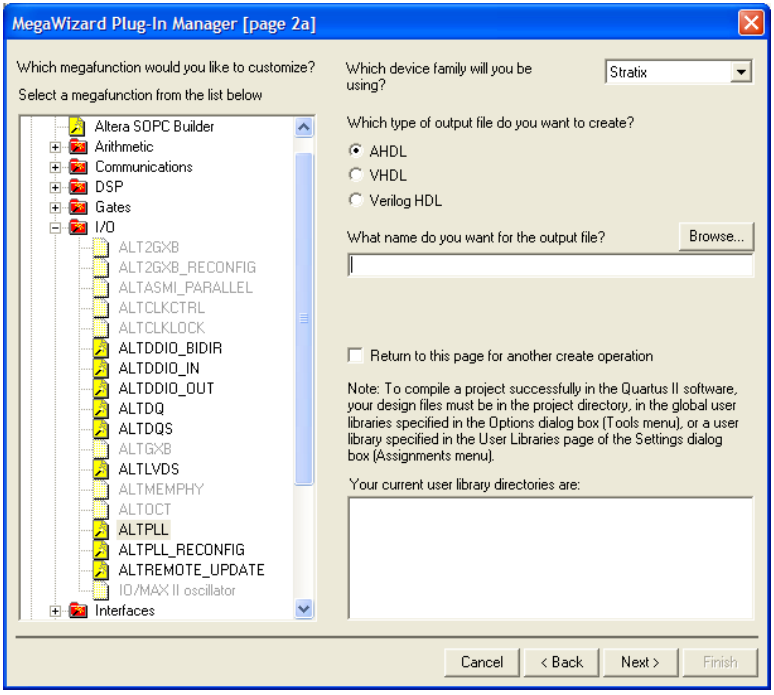
Generate 133 MHz, 200 MHz & 200 MHz Time-Shifted Clocks

1. In the Quartus II software, open the project file **shift_clk.qpf**.
2. Open the top-level **shift_clk.qpf**. 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**.

   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 wizard, 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.

Figure 2–23 shows page 2a after you have made these selections.

Figure 2–23. MegaWizard Plug-In Manager —altpll [Page 2a]



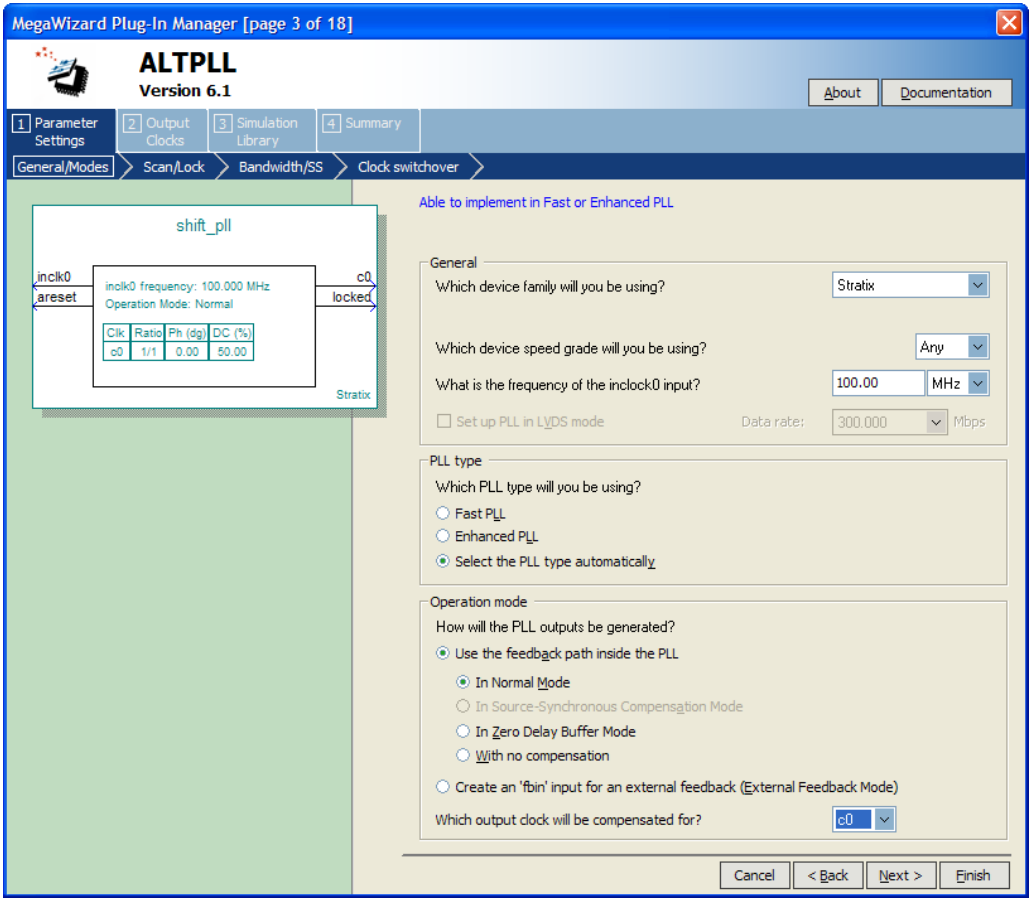
7. Name the output file **shift_pll**.
8. Click **Next**. Page 3 appears.

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

9. In the General section, for **What is the frequency of the inclock0 input?** type `100` and select **MHz**. Leave the other options as default in this section.
10. In the PLL type section, for **Which PLL type will you be using?**, click **Select the PLL type automatically**.
11. In the **Operation mode** section, make sure the **Use the feedback inside the PLL** and **In Normal Mode** options are turned on.
12. For **Which output clock will be compensated for?**, select **c0**.

Figure 2–24 shows page 3 after you have made these selections.

Figure 2–24. MegaWizard Plug-In Manager—altpll [Page 3 of 18]



13. Click **Next**. Page 4 appears.
14. In the Dynamic configuration section, make sure that **Create optional inputs for dynamic reconfiguration** is turned off.
15. In the Optional inputs selection:
    a. Turn on **Create an 'pllena' input to selectively enable the PLL**.

b. Turn on **Create an 'areset' input to asynchronously reset the PLL.**

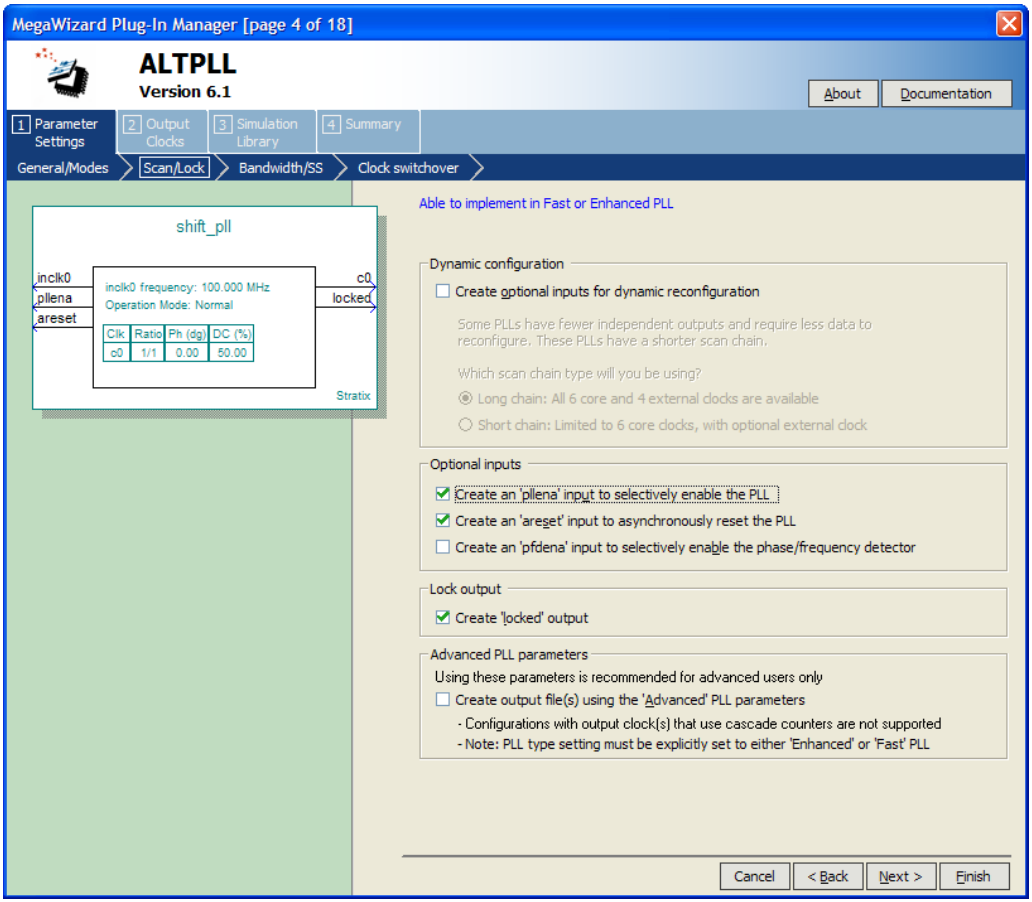
c. Turn off **Create an 'pfdena' input to selectively enable the phase/frequency detector**.

16. In the Lock output section, turn on **Create 'locked' output**.

17. Leave the Advanced PLL parameters as the default.

Figure 2–25 shows page 4 after you have made these selections.

Figure 2–25. MegaWizard Plug-In Manager—altpll [Page 4of 18]

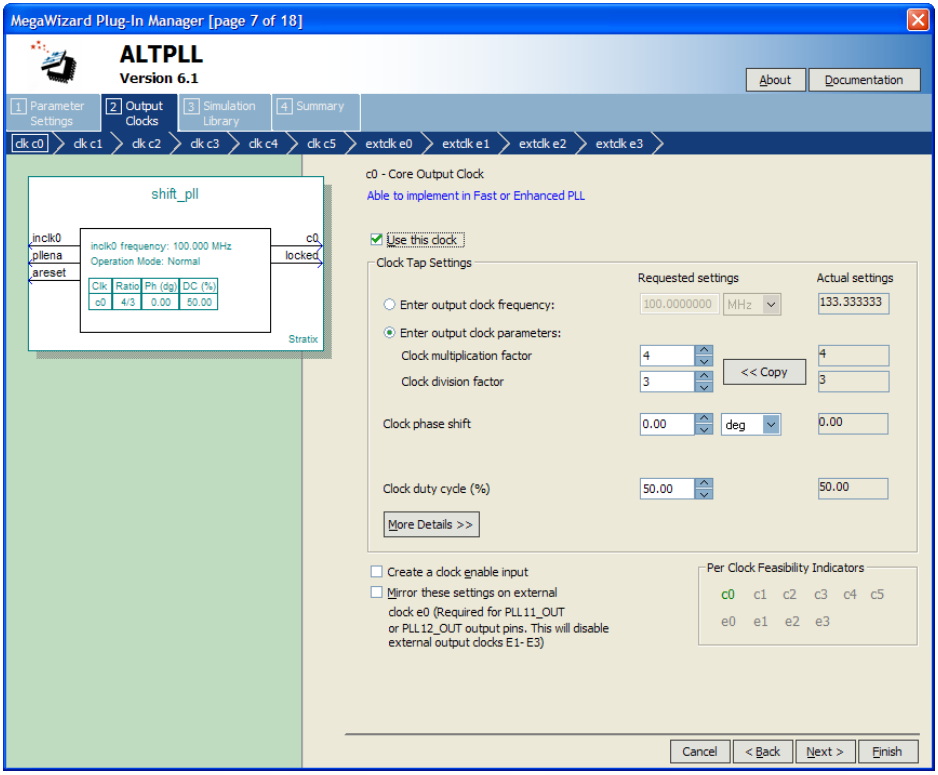


In the following steps, you specify the 133 MHz internal clock (c0).

18. Click on the Output Clocks tab to access configuration for all of the output clocks in the PLL. Page 7 appears.
19. Under Core Output Clock, turn on **Use this clock**.
20. 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`.
21. Leave the other options as the default.

    Figure 2–26 shows page 7 after you have made these selections.

Figure 2–26. MegaWizard Plug-In Manager —altpll [Page 7 of 18]



22. Click **Next**. Page 8 appears.

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

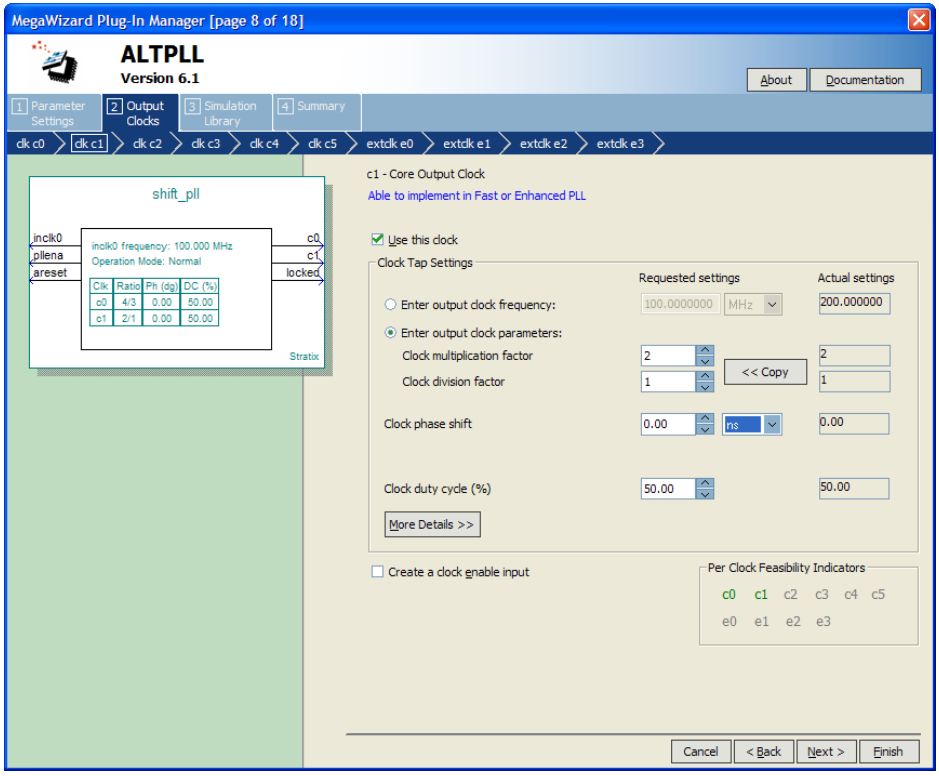
23. Under Core Output Clock, turn on **Use this clock**.
24. In the Clock Tap Settings section:
    a. Turn off **Enter output clock frequency:**.
    b. Turn on **Enter output clock parameters:**.
    c. For **Clock multiplication factor**, type `2`.
    d. For **Clock division factor**, type `1`.
    e. For **Clock phase shift**, type `0.00` and select **ns**.

f. For **Clock duty cycle (%)**, type `50.00`.

25. Leave the other options as the default.

26. Figure 2–27 shows page 8 after you have made these selections.

Figure 2–27. MegaWizard Plug-In Manager—altpll [Page 8 of 18]



27. Click **Next**. Page 9 appears.

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

28. Under Core Output Clock, turn on **Use this clock**.

29. 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`.

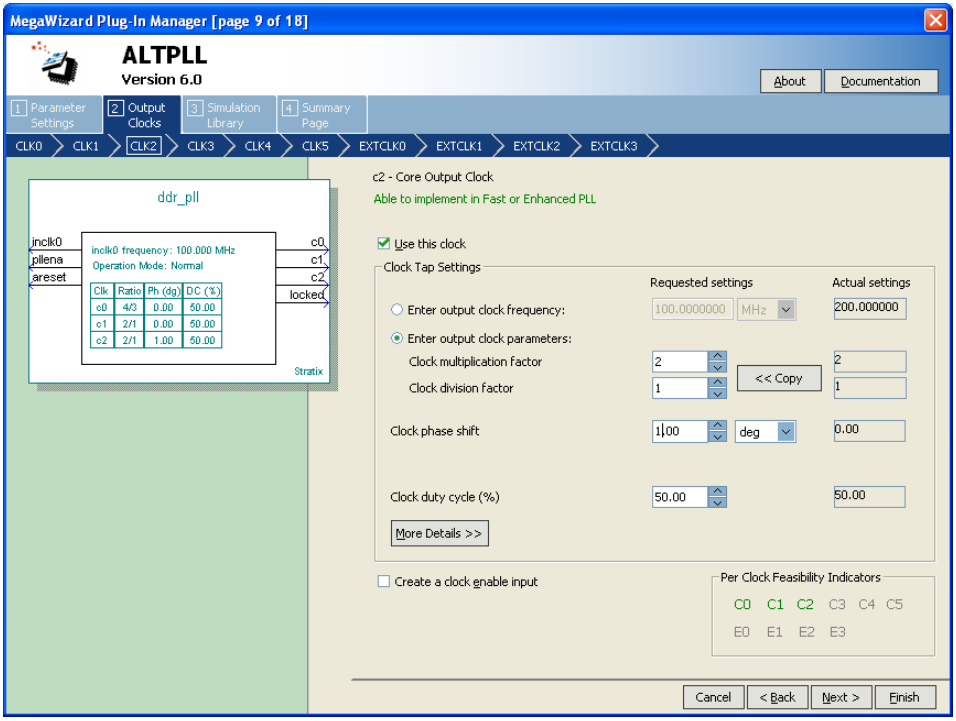
e. For **Clock phase shift**, type `1.00` and select **deg**.

f. For **Clock duty cycle (%)**, type `50.00`.

30. Leave the other options as the default.

Figure 2–28 shows page 9 after you have made these selections.

Figure 2–28. MegaWizard Plug-In Manager —altpll [Page 9 of 18]



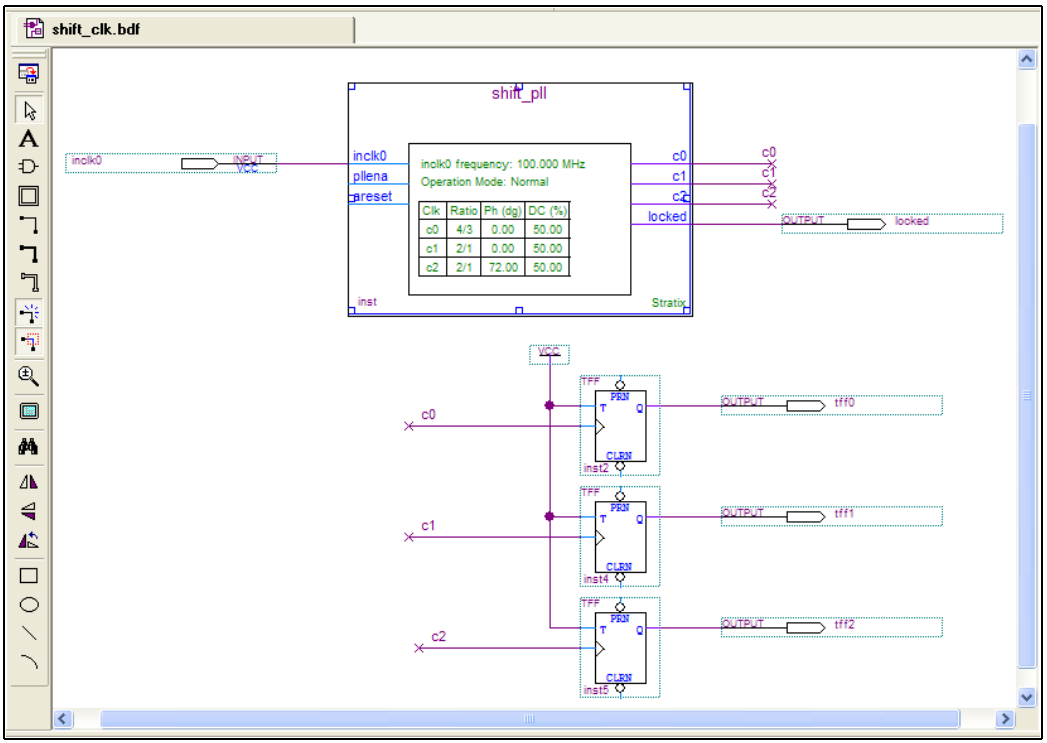
31. Click **Finish**. The `shift_pll` module is built.

32. In the Symbol dialog box, click **OK**.

33. 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 as shown in Figure 2–29.

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

Figure 2–29. 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.

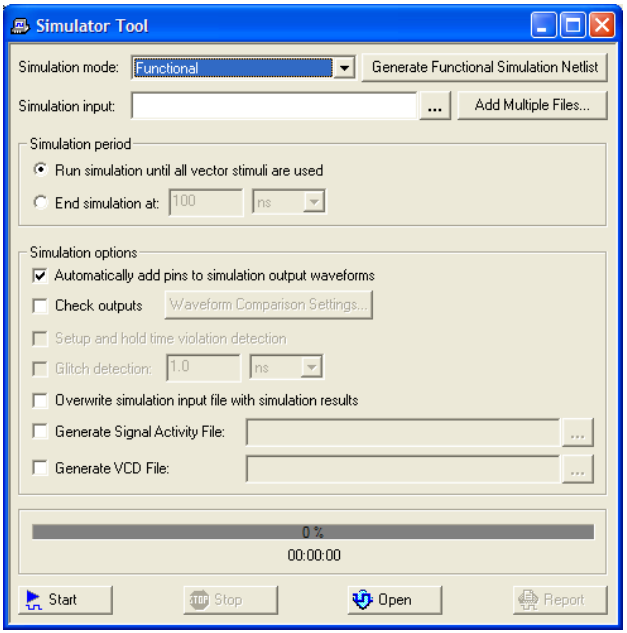
1. On the Assignments menu, click **Settings**.
2. In the **Category** list, click **Device**.
3. In the Target device section, under the Available devices list, select **EP1S10F780C5**.
4. Leave all other selections as the default.
5. Click **OK**.
6. Click **Start**, or on the Processing menu, click **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**.

Functional Results—Simulate the shift_clk Design in Quartus

In this section, you will simulate the design to verify the results. Set up the Quartus II Simulator by performing the following steps:

1. On the Processing menu, click **Simulator Tool** to open the **Simulator Tool** dialog box (Figure 2–30).

Figure 2–30. Simulator Tool Dialog Box



2. For Simulation mode, select **Functional**.
3. In the **Simulation input** field, browse to select the simulation input file **shift_clk.vwf**.
4. In the Simulation period section, select **Run simulation until all vector stimuli are used**.
5. In the Simulation options section, turn on **Automatically add pins to simulation output waveforms**. Leave the other options in this section un-checked.
6. Click the **Generate Functional Simulation Netlist** button.

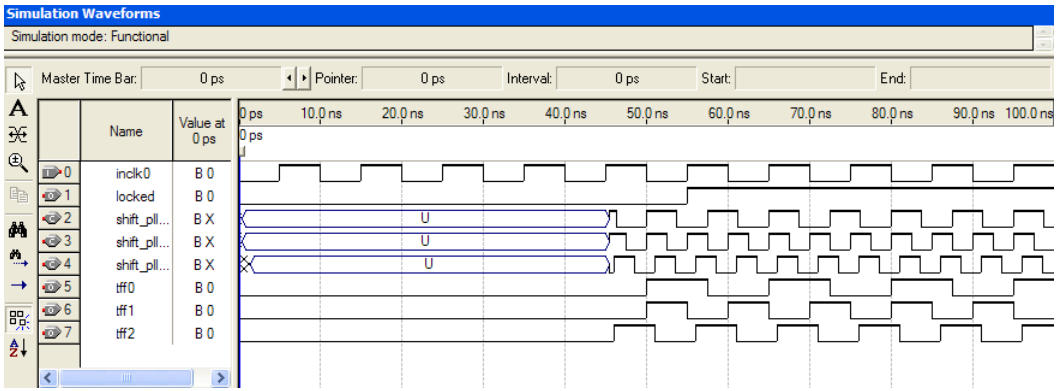
7. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.

8. Click **Start**.

9. When the **Simulation was successful** message box appears, click **OK**.

10. To verify the results of the simulation output waveform, click **Report**. Figure 2–31 shows the expected simulation results.

Figure 2–31. Functional Waveform for shift_clk Design



Simulate the shift_clk Design in ModelSim-Altera

Simulate the design in ModelSim to compare the results of both simulators. This User Guide assumes that you are familiar with using ModelSim-Altera before trying out the design example. If you are unfamiliar with ModelSim-Altera, refer to the support web page for ModelSim-Altera. There are various links to various topics, such 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 **ModelSim-Altera**.

6. On the File menu, click **Change Directory**.

7. Select the folder in which you unzipped the files. Click **OK**.

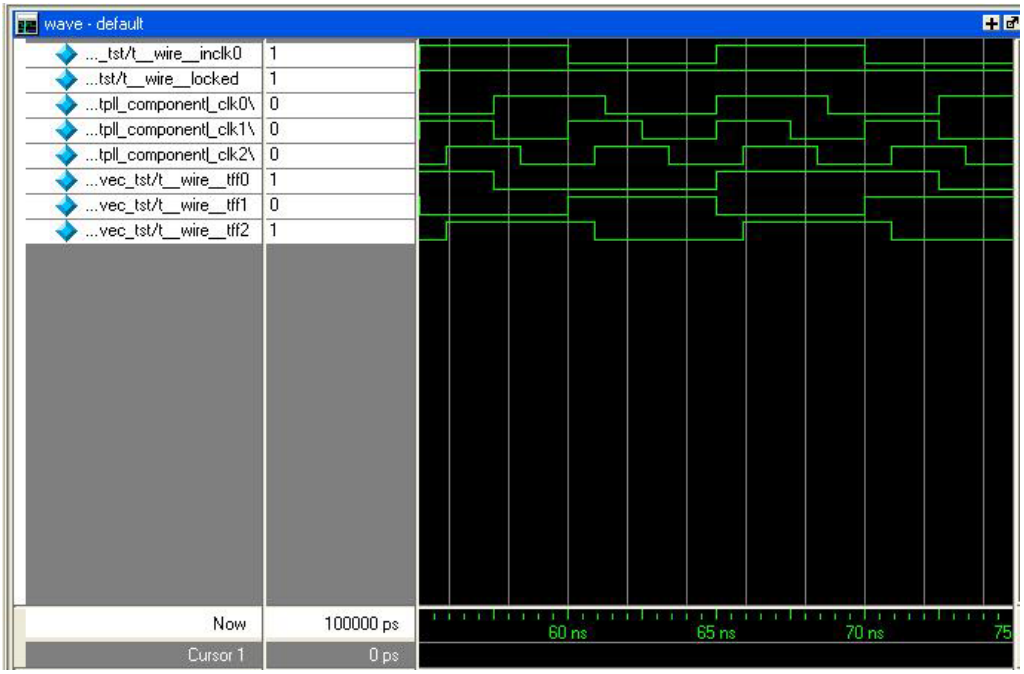
8. On the Tools menu, click **Execute Macro**.

9. Select **shift_clk.do**, and click **Open**. This is a script file for ModelSim which automates 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 2–32 shows the expected simulation results in ModelSim.

Figure 2–32. ModelSim Simulation Results



Conclusion

The Quartus II software provides parameterizable megafunctions ranging from simple arithmetic units, such as adders and counters, to advanced phase-locked loop (PLL) blocks, multipliers, and memory structures. These megafunctions are performance-optimized for Altera devices and therefore, provide more efficient logic synthesis and device implementation, because they automate the coding process and save valuable design time. You should use these functions during design implementation so you can consistently meet your design goals.

3. Specifications

Ports & Parameters

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.

Figure 3–1 shows the enhanced phase-locked loop (PLL) ports and parameters for the altpll megafunction. Table 3–1 shows the input ports, Table 3–2 shows the output ports, and Table 3–3 shows the altpll megafunction parameters.

Figure 3–1. Enhanced PLL

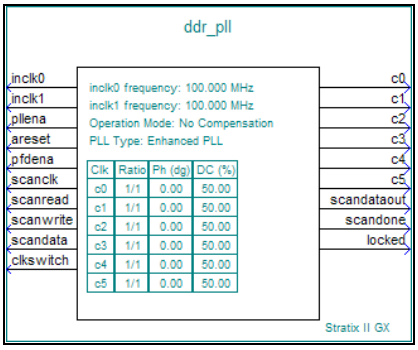


Figure 3–2. Fast PLL

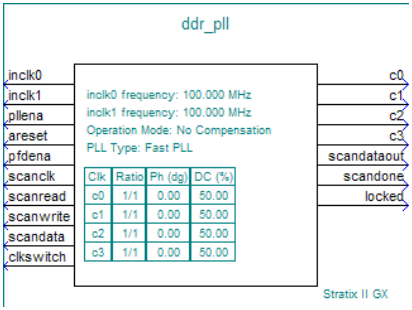


Figure 3–3. Top-Bottom PLL, Stratix III Devices

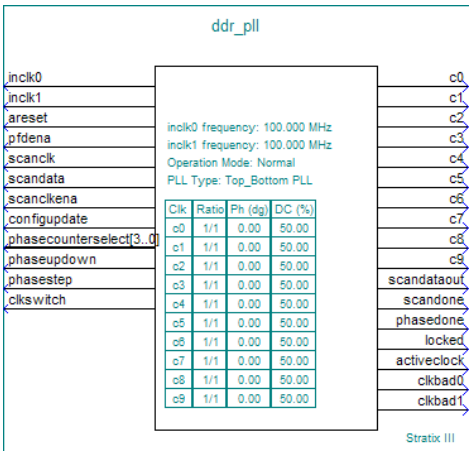


Figure 3–4. Left-Right PLL, Stratix III Devices

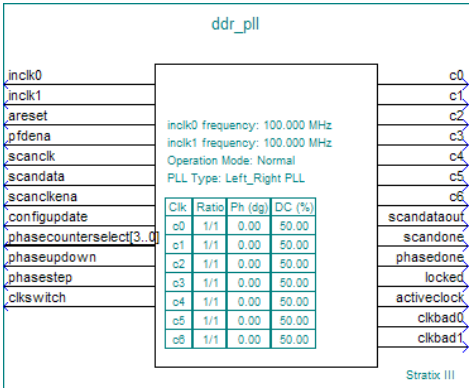


Table 3–1 shows the altpll megafunction input ports.

Table 3–1. altpll Megafunction Input Ports (Part 1 of 3)

Port Name	Required?	Description	Comments	Supported by Stratix III?
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, the fbout pin would feed the dedicated extclk pin, and through a board connection feed the fbin pin. The fbin pin is also required for zero delay buffer (ZDB) mode to connect to mimic the bidir I/O.	✓

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

Port Name	Required?	Description	Comments	Supported by Stratix III?
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 goes out of lock. The pllena port acts as a combined enable and reset pin. When the pin is reasserted, the PLL has to re-lock. 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. The PLL can programmed after the device has been configured.	✓
pfdena	—	Enables the phase frequency detector (PFD).	Allows the VCO pin to continue to operate. When the 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 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 Cyclone II and Stratix II devices.	—
extclkena[]	—	Enables the extclk[] ports to the PLL.	The extclkena[]port is not applicable to Cyclone II and Stratix 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 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 (Part 3 of 3)

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

Table 3–2 shows the altpll megafunction output ports.

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

Port Name	Required?	Description	Comments	Supported by Stratix III?
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 Cyclone II and Stratix II devices	—
clkbad[]	—	Specifies which signal goes high.	If the inclk0 stops toggling, the clkbad0 is high. If the inclk1 stops toggling, the clkbad1 is high.	✓
activeclock	—	Specifies when the clock switchover circuit initiates.	The clock switchover circuit initiates when the primary reference clock is not toggling correctly, or if you initiate a switch with the clkswitch input port.	✓
clkloss	—	Specifies when the clock switchover circuit initiates. Not used in Stratix III.	The clock switchover circuit initiates when the primary reference clock is not toggling correctly or if you specify with the clkswitch input port.	—

Table 3–2. altpll Megafunction Output Ports (Part 2 of 2)

Port Name	Required?	Description	Comments	Supported by Stratix III?
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 while the PLL is in the process of achieving lock.	✓
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 M counter and drives the output MUX. If a feedback path is not connected, the compiler automatically connects fbout to fbin. Additionally, a clkbuf is added to specify the resource type used, similar to other clock networks.	✓
enable0	—	Enable pulse output port.	This port is available only when the altpll megafunction is in LVDS mode.	—
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 shows the altpll megafunction parameters.

Table 3–3. altpll Megafunction Parameters (Part 1 of 8)

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. • The 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 will be 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. The maximum input delay assignment on the fbin port can be used to specify external board delay.
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.

Table 3–3. altpll Megafunction Parameters (Part 2 of 8)

Parameter	Type	Required?	Comments
COMPENSATE_CLOCK	String	—	Specifies the output clock port which should be compensated. If the OPERATION_MODE parameter is specified to NORMAL, values are CLK[], GCLK[], LCLK[], or LVDSCLK[]. • If the OPERATION_MODE parameter is specified to ZERO_DELAY_BUFFER, value is EXTCLK[]. • If the OPERATION_MODE parameter is specified to SOURCE_SYNCHRONOUS, values are CLK[], LCLK[], GCLK[], or LVDSCLK[]. This clock cannot offset with respect to the reference clock, and this relationship is preserved closely even upon temperature and frequency changes. • If the OPERATION_MODE parameter is specified to NORMAL, 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 specified 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. You can use the clock switch scheme to switch between clocks; however, you can only switch back to the primary clock with a user-initiated switch.
INCLK0_INPUT_FREQUENCY	Integer	✓	Specifies the input frequency for the inclk0 clock. The Compiler uses the frequency of 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 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 (Part 3 of 8)

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 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 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 switch over 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 is AUTO. For low bandwidth option, the PLL will have a better jitter rejection but slower lock time. For the high bandwidth option, it instead 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.

Table 3–3. altpll Megafunction Parameters (Part 4 of 8)

Parameter	Type	Required?	Comments
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.
LOCK_WINDOW_UI	String	—	Specifies the value of the LOCK_WINDOW_UI setting. If omitted, default is 0.05.
C[]_HIGH	Integer	—	Parameter [9..0]. Specifies the high period count for the corresponding C[9..0] counter. If omitted, default is 1.
C[]_LOW	Integer	—	Parameter [9..0]. Specifies the low period count for the corresponding C[9..0] counter. If omitted, default is 1.
C[]_INITIAL	Integer	—	Parameter [9..0]. Specifies the initial value for the corresponding C[9..0] counter. If omitted, default is 1.
C[]_PH	Integer	—	Parameter [9..0]. Specifies the phase tap for the C[9..0] counter. If omitted, default is 0.
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.
C[]_TEST_SOURCE	Integer	—	Parameter [9..0]. Specifies the test source for the corresponding C[9..0] counter. If omitted, default is 0.
C[]_USE_CASC_IN	String	—	Parameter [9..0]. Specifies whether to use cascade input for corresponding C[9..0] counter. Values are ON and OFF. If omitted, default is OFF.
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. This parameter can be specified 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.
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. This parameter can be specified 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.
CLK[]_PHASE_SHIFT	Integer	—	Specifies, in picoseconds (ps), phase shift for the corresponding clk[9..0] port. If omitted, the default is 0.

Table 3–3. altpll Megafunction Parameters (Part 5 of 8)

Parameter	Type	Required?	Comments
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, the two ports can be used simultaneously. If no units are specified, default is picoseconds (ps). Legal time delay values range from -3 ns through 6 ns in increments of 0.25 ns. These values should not be used as parameters except when reprogramming the PLL via the real-time programming interface.
CLK[]_DUTY_CYCLE	Integer	—	Specifies duty cycle for the corresponding clk[9..0] port by providing the percentage of high time. If omitted, the default is 50.
CLK[]_USE_EVEN_COUNTER_MODE	String	—	Specifies whether the clock output should be forced to be implemented using even counter mode for the corresponding CLK[9..0] port. If omitted, the default is OFF.
CLK[]_USE_EVEN_COUNTER_VALUE	String	—	Specifies whether the clock output should be forced to be implemented using even counter values for the corresponding CLK[9..0] port. If omitted, the default is OFF.
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. This parameter can only be specified 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.
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. This parameter can only be specified 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.
EXTCLK[]_PHASE_SHIFT	Integer	—	Specifies the phase shift for the corresponding extclk[3..0] port. This parameter is not available for Stratix II devices.

Table 3–3. altpll Megafunction Parameters (Part 6 of 8)

Parameter	Type	Required?	Comments
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, the two port can be used 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. These values should normally not be used as parameters except when reprogramming the PLL via 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, then 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. 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.

Table 3–3. altpll Megafunction Parameters (Part 7 of 8)

Parameter	Type	Required?	Comments
WIDTH_CLOCK	Integer	—	Specifies the clock width. Values are 10 for Stratix III devices and 6 for all other supported device families. If omitted, the default is 6. For Stratix III device designs, the WIDTH_CLOCK parameter is required for both clearbox and non-clearbox implementation to reflect the correct width.
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.

Table 3–3. altpll Megafunction Parameters (Part 8 of 8)

Parameter	Type	Required?	Comments
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.

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.

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 (Part 1 of 8)

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.

Table 3–4. Advanced altpll Megafunction Parameters (Part 2 of 8)

Parameter	Type	Required?	Comments
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.
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. Of 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.
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.

Table 3–4. Advanced altpll Megafunction Parameters (Part 3 of 8)

Parameter	Type	Required?	Comments
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 (Part 4 of 8)

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.

Table 3–4. Advanced altpll Megafunction Parameters (Part 5 of 8)

Parameter	Type	Required?	Comments
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.
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.

Table 3–4. Advanced altpll Megafunction Parameters (Part 6 of 8)

Parameter	Type	Required?	Comments
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.
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.

Table 3–4. Advanced altpll Megafunction Parameters (Part 7 of 8)

Parameter	Type	Required?	Comments
EXTCLK[]_COUNTER	String	—	Parameter [9..0]. 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 not available for Cyclone II and Stratix II 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.
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 (A), 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 (Part 8 of 8)

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 (i.e., 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 (i.e., phase amount that is less than one VCO period). In LVDS mode, this parameter defaults to a phase of 0.