35. PLL Cores

NII53002-9.1.0

Core Overview

The PLL cores, Avalon ALTPLL and PLL, provide a means of accessing the dedicated on-chip PLL circuitry in the Altera® Stratix® and Cyclone® series FPGAs. Both cores are a component wrapper around the Altera ALTPLL megafunction.

The Avalon ALTPLL core is a newer generation of the PLL cores. Altera recommends that you use this new core in your design as the older PLL core will be phased out in the near future.

The core takes an SOPC Builder system clock as its input and generates PLL output clocks locked to that reference clock.

The PLL cores support the following features:

- All PLL features provided by Altera's ALTPLL megafunction. The exact feature set depends on the device family.
- Access to status and control signals via Avalon Memory-Mapped (Avalon-MM) registers or top-level signals on the SOPC Builder system module.
- Dynamic phase reconfiguration in Stratix III and Stratix IV device families.

The PLL output clocks are made available in two ways:

- As sources to system-wide clocks in your SOPC Builder system.
- As output signals on your SOPC Builder system module.

For details about the ALTPLL megafunction, refer to the *ALTPLL Megafunction User Guide*.

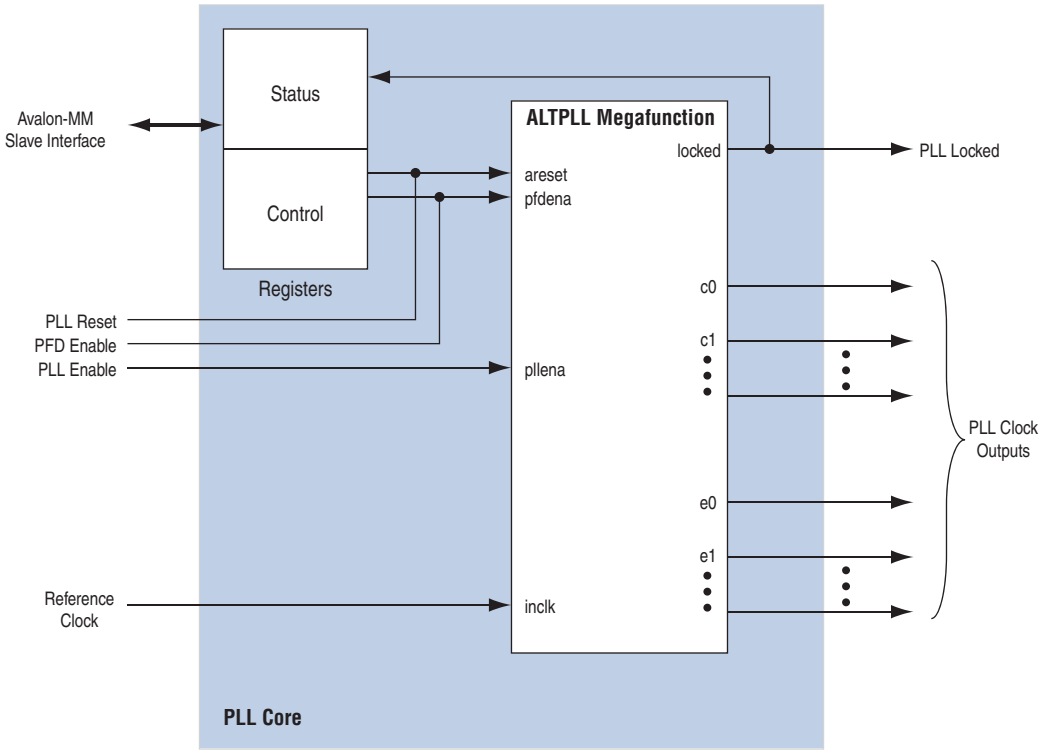
The PLL core is SOPC Builder-ready and integrates easily into any SOPC Builder-generated system. This chapter contains the following sections:

- "Functional Description"
- "Device Support" on page 35–3
- "Instantiating the Cores in SOPC Builder" on page 35–3
- "Hardware Simulation Considerations" on page 35–5
- "Register Definitions and Bit List" on page 35–5

Functional Description

Figure 35–1 shows a block diagram of the PLL cores and their connection to the PLL circuitry inside an Altera FPGA. The following sections describe the components of the core.

Figure 35–1. PLL Core Block Diagram



ALTPLL Megafunction

The PLL cores consist of an ALTPLL megafunction instantiation and an Avalon-MM slave interface. This interface can optionally provide access to status and control registers within the cores. The ALTPLL megafunction takes an SOPC Builder system clock as its reference, and generates one or more phase-locked loop output clocks.

Clock Outputs

Depending on the target device family, the ALTPLL megafunction can produce two types of output clock:

- internal (c)—clock outputs that can drive logic either inside or outside the SOPC Builder system module. Internal clock outputs can also be mapped to top-level FPGA pins. Internal clock outputs are available on all device families.
- external (e)—clock outputs that can only drive dedicated FPGA pins. They cannot be used as on-chip clock sources. External clock outputs are not available on all device families.

The Avalon ALTPLL core, however, does not differentiate the internal and external clock outputs and allows the external clock outputs to be used as on-chip clock sources.

To determine the exact number and type of output clocks available on your target device, refer to the *ALTPLL Megafunction User Guide*.

PLL Status and Control Signals

Depending on how the ALTPLL megafunction is parameterized, there can be a variable number of status and control signals. You can choose to export certain status and control signals to the top-level SOPC Builder system module. Alternatively, Avalon-MM registers can provide access to the signals. Any status or control signals which are not mapped to registers are exported to the top-level module. For details, refer to the "Instantiating the Cores in SOPC Builder" on page 35–3.

System Reset Considerations

At FPGA configuration, the PLL cores reset automatically. PLL-specific reset circuitry guarantees that the PLL locks before releasing reset for the overall SOPC Builder system module.

Resetting the PLL resets the entire SOPC Builder system module.

Device Support

The PLL cores support all Altera device families.

Instantiating the Cores in SOPC Builder

The PLL cores contain an instantiation of the ALTPLL megafunction. The MegaWizard™ interface for the PLL cores allows you to configure the ALTPLL megafunction, and specify connections to selected status and control signals of the megafunction.

For details about using the ALTPLL MegaWizard Plug-In Manager, refer to the *ALTPLL Megafunction User Guide*.

Instantiating the Avalon ALTPLL Core

When you instantiate the Avalon ALTPLL core, the MegaWizard Plug-In Manager is automatically launched for you to parameterize the ALTPLL megafunction. There are no additional parameters that you can configure in SOPC Builder.

The `pfdena` signal of the ALTPLL megafunction is not exported to the top level of the SOPC Builder module. You can drive this port by writing to the `PFDENA` bit in the `control` register.

The `locked`, `pllena/extclkena`, and `areset` signals of the megafunction are always exported to the top level of the SOPC Builder module. You can read the `locked` signal and reset the core by manipulating respective bits in the registers. See "Register Definitions and Bit List" on page 35–5 for more information on the registers.

Instantiating the PLL Core

This section describes the options available in the MegaWizard interface for the PLL core in SOPC Builder.

PLL Settings Page

The **PLL Settings** page contains a button that launches the ALTPLL MegaWizard Plug-In Manager. Use the MegaWizard Plug-In Manager to parameterize the ALTPLL megafunction. The set of available parameters depends on the target device family.

You cannot click **Finish** in the PLL wizard nor configure the PLL interface until you parameterize the ALTPLL megafunction.

Interface Page

The **Interface** page configures the access modes for the optional advanced PLL status and control signals.

For each advanced signal present on the ALTPLL megafunction, you can select one of the following access modes:

- **Export**—Exports the signal to the top level of the SOPC builder system module.
- **Register**—Maps the signal to a bit in a status or control register.

The advanced signals are optional. If you choose not to create any of them in the ALTPLL MegaWizard Plug-In, the PLL's default behavior is as shown in Table 35–1.

You can specify the access mode for the advanced signals shown in Table 35–1. The ALTPLL core signals, not displayed in this table, are automatically exported to the top level of the SOPC Builder system module.

Table 35–1. ALTPLL Advanced Signal

ALTPLL Name	Input / Output	Avalon-MM PLL Wizard Name	Default Behavior	Description
`areset`	input	PLL Reset Input	The PLL is reset only at device configuration.	This signal resets the entire SOPC Builder system module, and restores the PLL to its initial settings.
`pllena`	input	PLL Enable Input	The PLL is enabled.	This signal enables the PLL. `pllena` is always exported.
`pfdena`	input	PFD Enable Input	The phase-frequency detector is enabled.	This signal enables the phase-frequency detector in the PLL, allowing it to lock on to changes in the clock reference.
`locked`	output	PLL Locked Output	—	This signal is asserted when the PLL is locked to the input clock.

Asserting `areset` resets the entire SOPC Builder system module, not just the PLL.

Finish

Click **Finish** to insert the PLL into the SOPC Builder system. The PLL clock output(s) appear in the clock settings table on the SOPC Builder **System Contents** tab.

If the PLL has external output clocks, they appear in the clock settings table like other clocks; however, you cannot use them to drive components within the SOPC Builder system.

For details about using external output clocks, refer to the *ALTPLL Megafunction User Guide*.

The SOPC Builder automatically connects the PLL's reference clock input to the first available clock in the clock settings table.

If there is more than one SOPC Builder system clock available, verify that the PLL is connected to the appropriate reference clock.

Hardware Simulation Considerations

The HDL files generated by SOPC Builder for the PLL cores are suitable for both synthesis and simulation. The PLL cores support the standard SOPC Builder simulation flow, so there are no special considerations for hardware simulation.

Register Definitions and Bit List

Table 35–2 shows the register map for the PLL cores. Device drivers can control and communicate with the cores through two memory-mapped registers, `status` and `control`. The width of these registers are 32 bits in the Avalon ALTPLL core but only 16 bits in the PLL core.

In the PLL core, the `status` and `control` bits shown in Table 35–2 are present only if they have been created in the ALTPLL MegaWizard Plug-In Manager, and set to **Register** on the **Interface** page in the PLL wizard. These registers are always created in the Avalon ALTPLL core.

Table 35–2. PLL Cores Register Map

Offset	Register Name	R/W	Bit Description													
			31/15 *(2)*	30	29	...	9	8	7	6	5	4	3	2	1	0
0	`status`	R/O	*(1)*												`phasedone`	`locked`
1	`control`	R/W	*(1)*												`pfdena`	`areset`
2	`phase reconfig control`	R/W	`phase`		*(1)*			`counter_number`								
3	—	—	Undefined													

Notes to Table 35–2:

(1) Reserved. Read values are undefined. When writing, set reserved bits to zero.

(2) The registers are 32-bit wide in the Avalon ALTPLL core and 16-bit wide in the PLL core.

Status Register

Embedded software can access the PLL status via the `status` register. Writing to `status` has no effect. Table 35–3 describes the function of each bit.

Table 35–3. Status Register Bits

Bit Number	Bit Name	Value after reset	Description
0	locked *(2)*	1	Connects to the locked signal on the ALTPLL megafunction. The locked bit is high when valid clocks are present on the output of the PLL.
1	phasedone *(2)*	0	Connects to the phasedone signal on the ALTPLL megafunction. The phasedone output of the ALTPLL is synchronized to the system clock.
2:15/31 *(1)*	—	—	Reserved. Read values are undefined.

Note to Table 35–3:

(1) The status register is 32-bit wide in the Avalon ALTPLL core and 16-bit wide in the PLL core.

(2) Both the locked and phasedone outputs from the Avalon ALTPLL component are available as conduits and reflect the non-synchronized outputs from the ALTPLL.

Control Register

Embedded software can control the PLL via the control register. Software can also read back the status of control bits. Table 35–4 describes the function of each bit.

Table 35–4. Control Register Bits

Bit Number	Bit Name	Value after reset	Description
0	areset	0	Connects to the areset signal on the ALTPLL megafunction. Writing a 1 to this bit initiates a PLL reset.
1	pfdena	1	Connects to the pfdena signal on the ALTPLL megafunction. Writing a 0 to this bit disables the phase frequency detection.
2:15/31 *(1)*	—	—	Reserved. Read values are undefined. When writing, set reserved bits to zero.

Note to Table 35–4:

(1) The controlregister is 32-bit wide in the Avalon ALTPLL core and 16-bit wide in the PLL core.

Phase Reconfig Control Register

Embedded software can control the dynamic phase reconfiguration via the phase reconfig control register. Table 35–5 describes the function of each bit.

Table 35–5. Phase Reconfig Control Register Bits

Bit Number	Bit Name	Value after reset	Description
0:8	counter_number	—	A binary 9-bit representation of the counter that needs to be reconfigured. Refer to Table 35–6 for the counter selection.
9:29	—	—	Reserved. Read values are undefined. When writing, set reserved bits to zero.

Table 35–5. Phase Reconfig Control Register Bits

Bit Number	Bit Name	Value after reset	Description
30:31	`phase` *(1)*	—	01: Step up phase of `counter_number` 10: Step down phase of `counter_number` 00 and 11: No operation

Note to Table 35–5:

(1) Phase step up or down when set to 1 (only applicable to the Avalon ALTPLL core).

Table 35–6 lists the counter number and selection. For example, 100 000 000 selects counter C0 and 100 000 001 selects counter C1.

Table 35–6. Counter_Number Bits and Selection

Counter_Number [0:8]	Counter Selection
0 0000 0000	All output counters
0 0000 0001	M counter
> 0 0000 0001	Undefined
1 0000 0000	C0
1 0000 0001	C1
1 0000 0010	C2
...	...
1 0000 1000	C8
1 0000 1001	C9
> 1 0000 1001	Undefined

Referenced Documents

This chapter references the *ALTPLL Megafunction User Guide*.

Document Revision History

Table 35–7 shows the revision history for this chapter.

Table 35–7. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	Revised descriptions of register fields and bits.	Features added to the register map.
March 2009 v9.0.0	Added information on the new Avalon ALTPLL core.	A new PLL core, Avalon ALTPLL, is released and the chapter is updated accordingly to include the new core.

Table 35–7. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2008 v8.1.0	Changed to 8-1/2 x 11 page size. No change to content.	—
May 2008 v8.0.0	No change from previous release.	—

For previous versions of the *Quartus II Handbook*, refer to the Quartus II Handbook Archive.