High-Performance FPGA PLL Analysis with TimeQuest

August 2007, ver. 1.0 Application Note 471

Introduction

Phase-locked loops (PLLs) provide robust clock management and clock synthesis capabilities for maximum total system performance. Altera's high-density Stratix device families provide many highly versatile PLLs, and each PLL can be customized as a zero delay buffer, jitter attenuator, low skew fan-out buffer, or as a frequency synthesizer. To take advantage of the numerous features and capabilities provided by the Stratix device families, you should have a full understanding of all PLL-related reports generated by the Quartus® II software and the TimeQuest Timing Analyzer. This application note guides you through constraining PLLs and performing a timing analysis on the PLLs. Each of these steps includes examples and guidelines on how to read and understand the various reports relating to PLLs, and how to analyze and constrain PLLs in the TimeQuest Timing Analyzer.

Refer to the respective Stratix device handbook for specifications of the PLL features available on your selected device. Also, refer to the *altpll Megafunction User Guide* for information about how to customize the Stratix PLLs.

This document includes the following topics:

- "PLL Overview"
- "PLL Reports" on page 3
- "Constraining PLLs" on page 7
- "TimeQuest Reports" on page 9
- "PLL Features" on page 13

PLL Overview

One of the primary uses of a PLL is to synchronize the phase and frequency of an internal or external clock to an input reference clock. Numerous PLL components must work together to achieve this phase alignment.

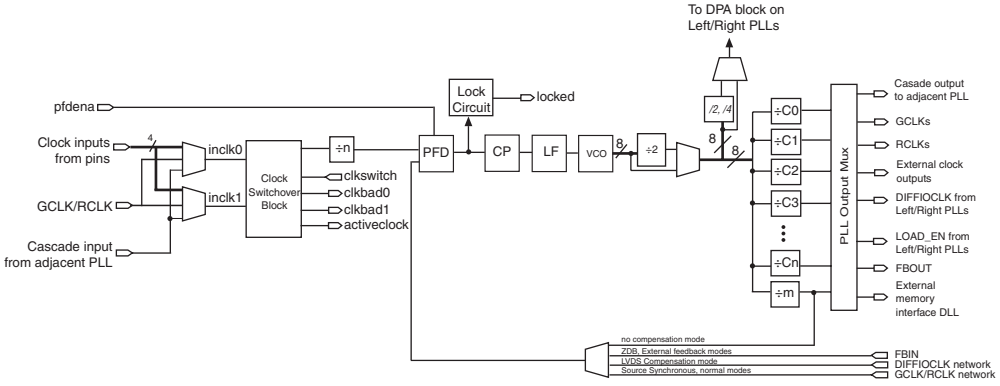
PLLs in the Stratix device families align the rising edge of the reference input clock to a feedback clock using a phase frequency detector (PFD) (Figure 1). The falling edges are determined by the duty cycle specifications. The PFD produces an up or down signal that determines whether the voltage controlled oscillator (VCO) should operate at a higher or lower frequency. The PFD output is applied to the charge pump and loop filter, which produce a control voltage for setting the frequency of the VCO. If the PFD produces an up signal, the VCO frequency

increases, while a down signal causes the VCO frequency to decrease. The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if the charge pump receives a down signal, current is drawn from the loop filter. The loop filter converts these up and down signals to a voltage that is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage overshoot, which minimizes the jitter on the VCO.

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

Figure 1 is a simplified block diagram that shows the major components of the Stratix III PLL.

Figure 1. Stratix III PLL Block Diagram



The following section provides an overview of the altpll megafunction.

altpll Megafunction Overview

The altpll megafunction provides support for the customization of the PLLs in the Stratix device families. The altpll megafunction implements different PLL configurations to generate and customize clock signals, distribute clock signals to different devices in a design, reduce clock skew between devices, and generate internal clock signals.

For more information about the altpll megafunction, refer to the *altpll Megafunction User Guide*.

PLL Reports

PLL reports play an important part in PLL analysis. Both the Quartus II software and the TimeQuest Timing Analyzer generate reports that allow you to verify the functionality and implemenation of all PLLs in your design.

This section describes the various reports available.

Fitter Reports

After a full compilation in the Quartus II software, the Fitter report, contained in the Compilation Report, provides details that show the various configurations of all the PLLs implemented in the target Stratix device. There are two reports that provide information on the PLL implementation in the device: the PLL Summary report and the PLL Usage report. Both reports are located under Resource Section in the Fitter folder of the Compilation Report.

These reports are not generated if the design does not include PLLs.

PLL Summary Report

The PLL Summary report lists information about the specific type of PLL chosen in the design. Table 1 describes the different PLL properties shown in the PLL Summary report.

Not all PLL properties shown in Table 1 are available for all PLLs in the Stratix device families.

Table 1. PLL Summary Report Values (Part 1 of 3)

PLL Property	Description
PLL type	Shows the type of PLL.
PLL mode	Shows the PLL operating mode.

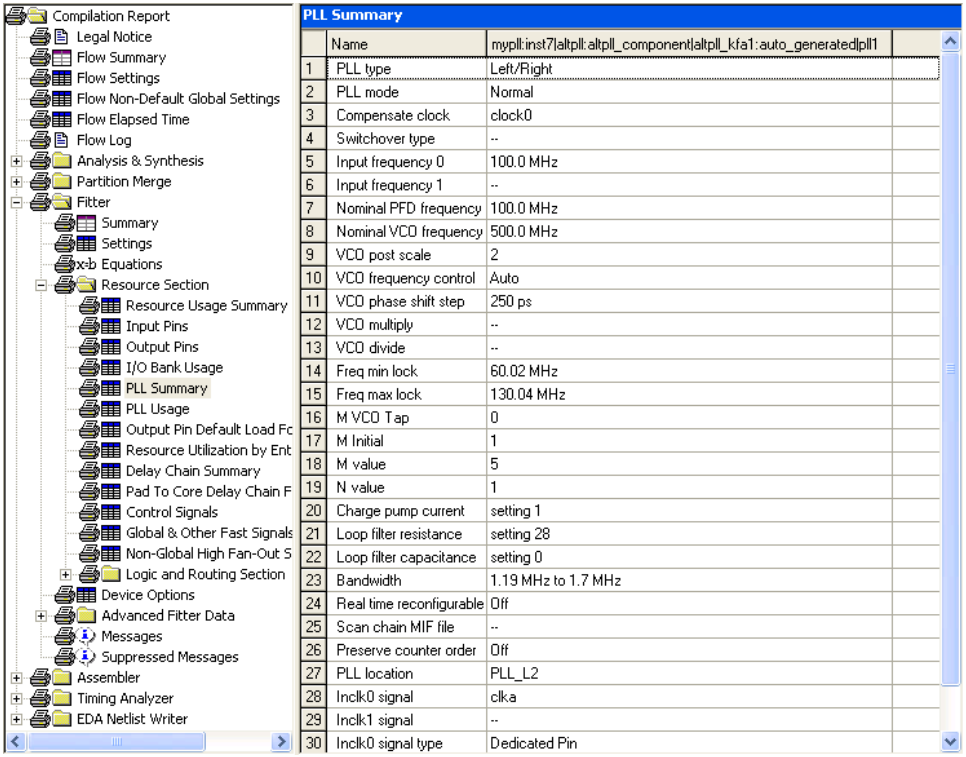
Table 1. PLL Summary Report Values (Part 2 of 3)

PLL Property	Description
Feedback source	Shows which output clock has a board level connection to the external feedback input pin to the PLL.
Compensate clock	Shows the PLL compensate clock source.
Switchover type	Shows the clock switchover type selected by the user.
Switchover on loss of clock	Shows whether or not the switchover on loss of clock was turned on for the PLL.
Switchover counter	Shows the value of the PLL switchover counter.
Self reset on gated loss of lock	Shows whether a self reset is asserted on loss of lock.
Gate lock counter	Shows the value of the 20-bit counter (0–1048575) that gates the locked output from the PLL.
Input frequency 0	Shows the input frequency for input clock 0 in MHz.
Input frequency 1	Shows the input frequency for input clock 1 in MHz.
Nominal PFD frequency	Shows the value of the nominal PFD frequency for the PLL in MHz.
Nominal VCO frequency	Shows the value of the nominal VCO frequency for the PLL in MHz.
VCO post scale	Shows the value of the VCO post-scale counter.
VCO frequency control	Shows the VCO frequency control mode.
VCO multiply	Shows the multiplication factor for the VCO output clock.
VCO divide	Shows the division factor for the VCO output clock.
Freq min lock	Shows the minimum lock input frequency, in MHz, for the PLL.
Freq max lock	Shows the maximum lock input frequency, in MHz, for the PLL.
M VCO Tap	Shows the value of the M VCO tap.
M initial	Shows the number of initial VCO cycles before the M counter starts.
M value	Shows the value of the M counter.
N value	Shows the value of the N counter.
M2 value	Shows the spread spectrum modulus for the M counter.
N2 value	Shows the spread spectrum modulus for the N counter.
SS counter	Shows the value for the spread spectrum counter for the PLL.
Downspread	Shows the downspread percentage for the PLL.
Spread frequency	Shows the spread frequency for the PLL in MHz.
Charge pump current	Shows the charge pump current value for the PLL in microamperes.
Loop filter resistance	Shows the value for the loop filter R resistor in Ohms.
Loop filter capacitance	Shows the value for the loop filter C capacitor in picoFarads.
Bandwidth	Shows the typical and minimum to maximum bandwidth value for the PLL in MHz or KHz.
Real time configurable	Shows whether the real time configuration option was turned on for the PLL.

Table 1. PLL Summary Report Values (Part 3 of 3)	
PLL Property	**Description**
Scan chain MIF file	Shows the Memory Initialization File (**.mif**) generated by the compiler to represent the initial state of the scan chain. Use with the altpll_reconfig megafunction to reconfigure the PLL.
Preserve counter order	Allows you to use specific counters with specific clock outputs.
PLL location	Shows the location of the PLL.
Inclk0 signal	Shows the primary clock input to the PLL.
Inclk1 signal	Shows the secondary clock input to the PLL.
Inclk0 signal type	Shows the primary clock input to the PLL type.
Inclk1 signal type	Shows the secondary clock input to the PLL type

Figure 2 shows a portion of the PLL Summary report generated for a sample design.

Figure 2. Sample PLL Summary Report



PLL Summary

	Name	mypll:inst7|altpll:altpll_component|altpll_kfa1:auto_generated|pll1
1	PLL type	Left/Right
2	PLL mode	Normal
3	Compensate clock	clock0
4	Switchover type	--
5	Input frequency 0	100.0 MHz
6	Input frequency 1	--
7	Nominal PFD frequency	100.0 MHz
8	Nominal VCO frequency	500.0 MHz
9	VCO post scale	2
10	VCO frequency control	Auto
11	VCO phase shift step	250 ps
12	VCO multiply	--
13	VCO divide	--
14	Freq min lock	60.02 MHz
15	Freq max lock	130.04 MHz
16	M VCO Tap	0
17	M Initial	1
18	M value	5
19	N value	1
20	Charge pump current	setting 1
21	Loop filter resistance	setting 28
22	Loop filter capacitance	setting 0
23	Bandwidth	1.19 MHz to 1.7 MHz
24	Real time reconfigurable	Off
25	Scan chain MIF file	--
26	Preserve counter order	Off
27	PLL location	PLL_L2
28	Inclk0 signal	clka
29	Inclk1 signal	--
30	Inclk0 signal type	Dedicated Pin

PLL Usage Report

The PLL Usage report shows the values of the specific output clock(s) for the PLLs in the design.

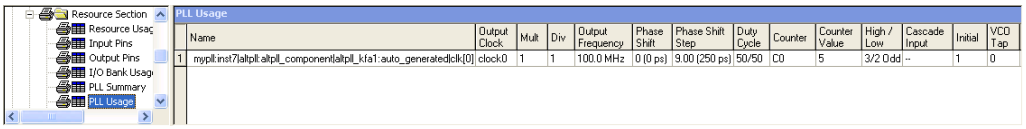
Table 2 describes the PLL properties shown in the PLL Usage report.

Table 2. PLL Usage Report Values

Column Heading	Description
Name	Shows the instance name of the PLL.
Output clock	Shows the output clock that was specified for the PLL.
Mult	Shows the multiplication factor for the PLL output clock.
Div	Shows the division factor for the PLL output clock.
Output Frequency	Shows the clock frequency value for the PLL output port. This value is equal to the input frequency multiplied by the multiplication factor for the PLL output port, divided by the division factor for the PLL output port.
Phase Shift	Shows the value specified for the phase shift of the PLL output clock, measured in degrees and picoseconds.
Duty Cycle	Shows the duty cycle of the output clock.
Counter	Shows the counter specified for the PLL output clock.
Counter Value	Shows the counter value for the PLL. This value is equal to the sum of the high and low cycles specified for the PLL. Bypass is shown if there are no high or low cycles specified for the PLL.
High/Low	Shows the number of high and low cycles for the PLL.
Cascade Input	Specifies the name of the preceding counter that is feeding the cascade input of the current counter.
Initial	Shows the number of initial VCO cycles before the counter starts.
VCO Tap	Shows the VCO tap value for the counter.

Figure 3 shows a portion of the PLL Usage report generated for a sample design.

Figure 3. Sample PLL Usage Report



Constraining PLLs

All clocks of the Stratix PLLs must be constrained to obtain a thorough static timing analysis with the TimeQuest Timing Analyzer. Without the proper clock constraints on the input and outupt clocks of the PLL, the TimeQuest Timing Analyzer will not analyze any clock transfers from these clocks. The TimeQuest Timing Analyzer supports three clock constraints for constraining PLL clocks:

- `derive_pll_clocks`—Used to automatically create generated clocks on the PLL output clocks.
- `create_clock`—Used to create a base clock.
- `create_generated_clock`—Used to create a derived/generated clock.

Perform the following steps to constrain all PLLs:

1. Constrain all input clocks to the PLL. Use the `create_clock` command. For example, `create_clock -period 10 -name \ clk_sys [get_ports inclk_pll]`

2. Constrain all PLL output clocks in one of the following ways:

   - Use `derive_pll_clocks` to automatically constrain the PLL output clocks.

   or

   - Use `create_generated_clock` to constrain the PLL output clocks individually.

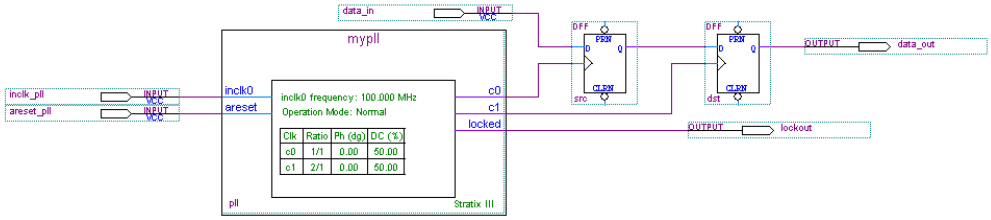
Altera recommends that you use the `derive_pll_clocks` command to automatically constrain the PLL output clocks. This allows the PLL output clock properties to be automatically updated if any of the clock properties are changed by the PLL MegaWizard® plug-in. Without the `derive_pll_clocks` command, each individual `create_generated_clock` constraint needs to be updated if any PLL output clock properties change.

If the `derive_pll_clocks` command is used, any input clocks to the PLL must still be constrained. There is no need to issue the `derive_pll_clocks` command multiple times in the same analysis. The first time the command is issued, all PLL output clocks are constrained.

If individual `create_generated_clock` constraints are used to constrain the PLL output clocks, the TimeQuest Timing Analyzer issues a warning if there are differences between the parameters specified in the `create_generated_clock` constraints and the parameters specified for the PLL output clocks in the megafunction.

Figure 4 shows a typical PLL design in a Stratix III device.

Figure 4. Stratix III PLL Design



To constrain the clocks shown in Figure 4, the following constraints can be applied:

```
# create a base clock for the input of the PLL
create_clock -period 10 -name clk_sys [get_ports inclk_pll]

# automatically create generated clocks on the output clocks of the PLL
derive_pll_clocks
```

Without the use of the `derive_pll_clocks` command, the PLL output clocks can be constrained as shown in the following example:

```
# create a generated clock on the first output clock of the PLL
create_generated_clock -source \
    pll|altpll_component|auto_generated|pll1|inclk[0] \
    -name my_pll_clk[0] pll|altpll_component|auto_generated|pll1|clk[0]

# create a generated clock on the second output clock of the PLL
create_generated_clock -source \
    pll|altpll_component|auto_generated|pll1|inclk[0] \
    -multiply_by 2 -name my_pll_clk[1] \
    pll|altpll_component|auto_generated|pll1|clk[1]

# create a base clock for the input of the PLL
create_clock -period 10 -name clk_sys [get_ports inclk_pll]
```

TimeQuest Reports

The TimeQuest Timing Analyzer provides various reports that can be used to verify that all PLL parameters have been specified correctly.

Report Clocks Report

The Report Clocks report summarizes all clocks that have been created in the design. This report details all parameters specified for all base clocks and all generated clocks in the design.

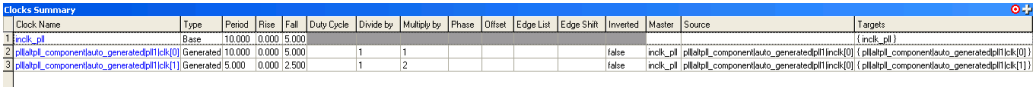
Table 3 describes the clock properties shown in the Report Clock report.

Table 3. Report Clocks Report Properties

Column Heading	Description
Clock Name	Shows the name of the clock.
Type	Shows the type of clock—either Base or Generated.
Period	Shows the period for the clock.
Rise	Shows the initial rise edge of the clock.
Fall	Shows the initial fall edge of the clock.
Duty Cycle	Shows the duty cycle of the clock.
Divide by	Shows any divide by factor for the clock.
Multiply by	Shows any multiply by factor for the clock.
Phase	Shows any phase shift of the clock.
Offset	Shows any offset of the clock.
Edge List	Shows the edge list for the clock.
Edge Shift	Shows the edge shift for the clock.
Inverted	Shows whether the clock is inverted with respect to the source clock.
Master	Shows the master clock.
Slave	Shows the source node for the clock.
Targets	Shows the node that the clock has been applied to.

Figure 5 shows a sample Report Clocks report.

Figure 5. Report Clocks Report


Clocks Summary

	Clock Name	Type	Period	Rise	Fall	Duty Cycle	Divide by	Multiply by	Phase	Offset	Edge List	Edge Shift	Inverted	Master	Source	Targets
1	inclk_pll	Base	10.000	0.000	5.000											{ inclk_pll }
2	pllaltpll_component|auto_generatedpll1|clk[0]	Generated	10.000	0.000	5.000		1	1					false	inclk_pll	pllaltpll_component|auto_generatedpll1|inclk[0]	{ pllaltpll_component|auto_generatedpll1|clk[0] }
3	pllaltpll_component|auto_generatedpll1|clk[1]	Generated	5.000	0.000	2.500		1	2					false	inclk_pll	pllaltpll_component|auto_generatedpll1|inclk[0]	{ pllaltpll_component|auto_generatedpll1|clk[1] }

The Report Clocks report contains various columns that show the properties of each clock in the design. Use the `report_clocks` command to generate a Report Clocks report.

Usage:

```
report_clocks [-append] [-desc] [-file <name>]
[-panel_name <name>] [-stdout] [-summary]
```

`-append`	If output is sent to a file, this option appends the result to that file. Otherwise, the file is overwritten.
`-desc`	Sorts the clocks by name in descending order (ascending order is the default).
`-file` <*name*>	Sends the results to a file.
`-panel_name` <*name*>	Sends the results to the panel and specifies the name of the new panel.
`-stdout`	Sends the output to `stdout`, via messages. Use this option if you have selected another output format, such as a file, and would also like to receive messages.
`-summary`	Creates a single table with a summary of each clock.

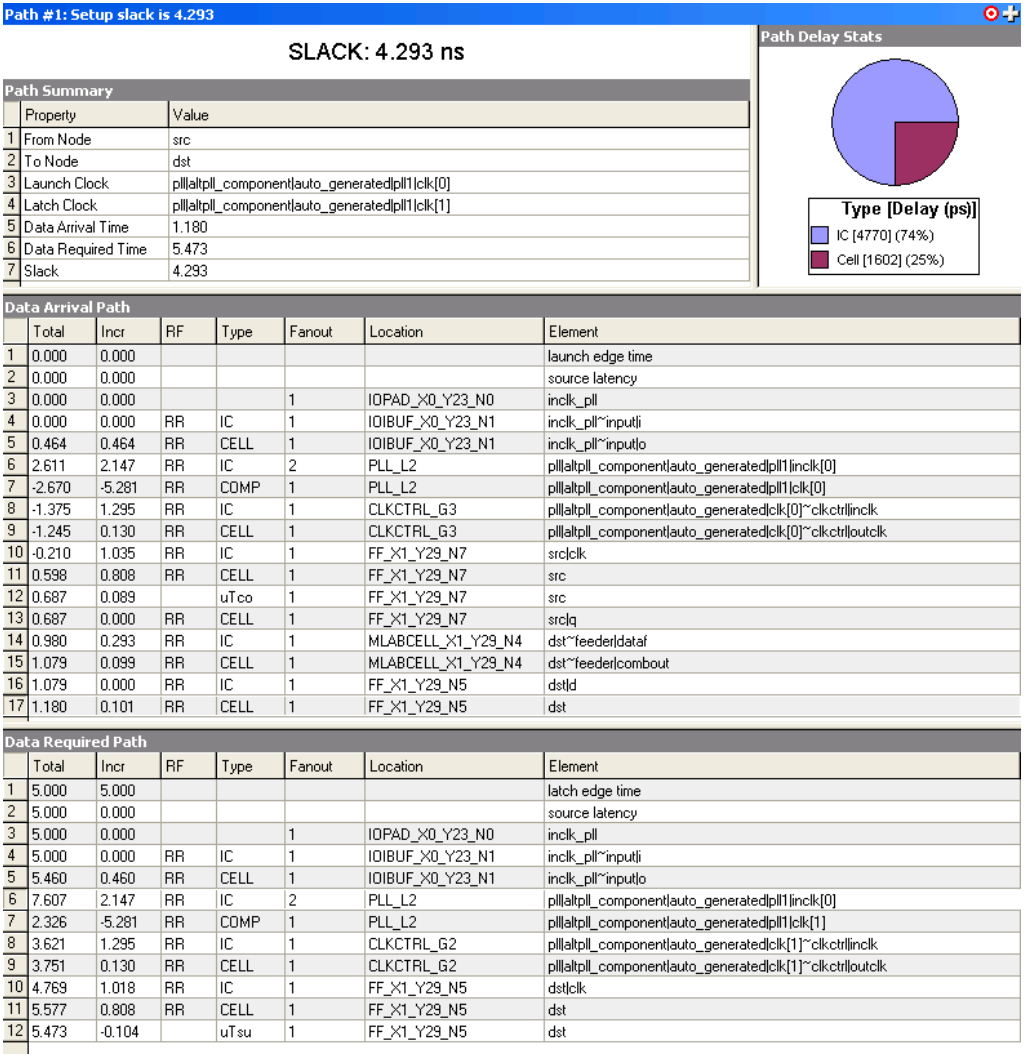
Report Timing Report

The Report Timing report shows the data arrival and data required times, and the associated slack for any register-to-register path. Various analysis types can be performed, including setup, hold, recovery, and removal. The Report Timing report also shows the timing delays, including compensation delay, for any PLLs that are part of the clock path.

Figure 6 shows a sample Report Timing report, with the detail level set to Full Path, for a register-to-register path.

For more information about the Report Timing detail levels, refer to the *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Figure 6. Report Timing Report



Path #1: Setup slack is 4.293

SLACK: 4.293 ns

Path Delay Stats

Path Summary

	Property	Value
1	From Node	src
2	To Node	dst
3	Launch Clock	pll|altpll_component|auto_generated|pll1|clk[0]
4	Latch Clock	pll|altpll_component|auto_generated|pll1|clk[1]
5	Data Arrival Time	1.180
6	Data Required Time	5.473
7	Slack	4.293

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	0.000	0.000					source latency
3	0.000	0.000			1	IOPAD_X0_Y23_N0	inclk_pll
4	0.000	0.000	RR	IC	1	IOIBUF_X0_Y23_N1	inclk_pll~input|i
5	0.464	0.464	RR	CELL	1	IOIBUF_X0_Y23_N1	inclk_pll~input|o
6	2.611	2.147	RR	IC	2	PLL_L2	pll|altpll_component|auto_generated|pll1|inclk[0]
7	-2.670	-5.281	RR	COMP	1	PLL_L2	pll|altpll_component|auto_generated|pll1|clk[0]
8	-1.375	1.295	RR	IC	1	CLKCTRL_G3	pll|altpll_component|auto_generated|clk[0]~clkctrl|inclk
9	-1.245	0.130	RR	CELL	1	CLKCTRL_G3	pll|altpll_component|auto_generated|clk[0]~clkctrl|outclk
10	-0.210	1.035	RR	IC	1	FF_X1_Y29_N7	src|clk
11	0.598	0.808	RR	CELL	1	FF_X1_Y29_N7	src
12	0.687	0.089		uTco	1	FF_X1_Y29_N7	src
13	0.687	0.000	RR	CELL	1	FF_X1_Y29_N7	src|q
14	0.980	0.293	RR	IC	1	MLABCELL_X1_Y29_N4	dst~feeder|dataf
15	1.079	0.099	RR	CELL	1	MLABCELL_X1_Y29_N4	dst~feeder|combout
16	1.079	0.000	RR	IC	1	FF_X1_Y29_N5	dst|d
17	1.180	0.101	RR	CELL	1	FF_X1_Y29_N5	dst

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	5.000	5.000					latch edge time
2	5.000	0.000					source latency
3	5.000	0.000			1	IOPAD_X0_Y23_N0	inclk_pll
4	5.000	0.000	RR	IC	1	IOIBUF_X0_Y23_N1	inclk_pll~input|i
5	5.460	0.460	RR	CELL	1	IOIBUF_X0_Y23_N1	inclk_pll~input|o
6	7.607	2.147	RR	IC	2	PLL_L2	pll|altpll_component|auto_generated|pll1|inclk[0]
7	2.326	-5.281	RR	COMP	1	PLL_L2	pll|altpll_component|auto_generated|pll1|clk[1]
8	3.621	1.295	RR	IC	1	CLKCTRL_G2	pll|altpll_component|auto_generated|clk[1]~clkctrl|inclk
9	3.751	0.130	RR	CELL	1	CLKCTRL_G2	pll|altpll_component|auto_generated|clk[1]~clkctrl|outclk
10	4.769	1.018	RR	IC	1	FF_X1_Y29_N5	dst|clk
11	5.577	0.808	RR	CELL	1	FF_X1_Y29_N5	dst
12	5.473	-0.104		uTsu	1	FF_X1_Y29_N5	dst

With the detail level set to Full Path, the entire clock path is shown in the Data Arrival Path and Data Required Path sections of the report. The report includes the columns shown in Table 4

Table 4. Report Timing Report Description

Column Heading	Description
Incr	Shows the time delay increment by the element.
RF	Shows the input and output transitions for the element.
Type	Shows the type of resource of the element.
Fanout	Shows the number of fanouts from the element.
Location	Shows the location on the Stratix device.
Element	Shows the name of the element.

For the Data Arrival Path, lines 3 to 9 detail the clock path to the source register, as shown in Table 5.

Table 5. Clock Path Description

Line Number	Description
Line 3	Shows the input clock port named `inclk_pll` located at IOPAD X0 Y23.
Lines 4 and 5	Show the path through the input buffer of IOPAD X0 Y23.
Line 6	Shows the interconnect delay to the input pin of the PLL.
Line 7	Shows the compensation delay due to the PLL. This is indicated by the COMP entry in the Type column.
Lines 8 and 9	Show the interconnect and global buffer delays, respectively.

Line 7 shows the total PLL compensation if the PLL is in normal mode, source-synchronous mode, or zero delay buffer mode.

The TimeQuest Timing Analyzer does not report which mode the PLL has been set to. The PLL mode can be determined in the PLL Summary report.

Any specified phase shift of the PLL output clocks appears as part of the Launch Time or Latch Time, depending on the clock path to either the source register or destination register, respectively.

PLL Features

The following sections describe some of the PLL features available in the Stratix device family.

Refer to the respective Stratix device handbook for specifications of the PLL features available on your selected device.

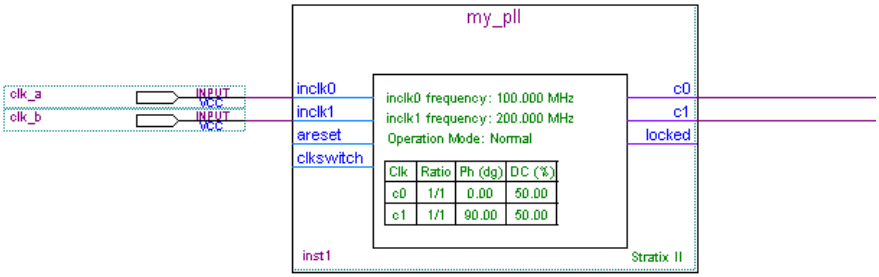
Clock Switchover

Clock switchover is a feature that allows the PLL to switch between two input reference clocks. You can use clock switchover for clock redundancy, or for a dual clock domain application. This feature of the Stratix PLLs can be used to develop a highly reliable system in which you can use a redundant clock in case the primary clock fails. Clock switchover can be performed automatically, when the clock is no longer toggling, or manually through a user control signal.

You must create clocks on the inputs of the PLL clock pins in the TimeQuest Timing Analyzer. The PLL output clocks can be either created individually or with the `derive_pll_clocks` command.

Figure 7 shows a typical PLL with clock switchover enabled.

Figure 7. Typical PLL with Clock Switchover Enabled



The PLL input clock 0 is set to 100 MHz and the input clock 1 is set to 200 MHz. To constrain the clocks in this design you can specify the following constraints:

```
# create input clock 0
create_clock -period 10 -name clk_100 [get_ports clk_a]

# create input clock 1
create_clock -period 5 -name clk_200 [get_ports clk_b]
```

```
# create the PLL output clocks
derive_pll_clocks
```

The `derive_pll_clocks` command will constrain the two PLL output clocks with respect to the two input clocks. For the above design, the following constraints are created:

```
create_generated_clock -source inst1|altpll_component|pll|inclk[0] -name \
    inst1|altpll_component|pll|clk[0] inst1|altpll_component|pll|clk[0]

create_generated_clock -source inst1|altpll_component|pll|inclk[1] -name \
    inst1|altpll_component|pll|clk[0]~1 inst1|altpll_component|pll|clk[0]

create_generated_clock -source inst1|altpll_component|pll|inclk[0] -phase 90.00 -name \
    inst1|altpll_component|pll|clk[1] inst1|altpll_component|pll|clk[1]

create_generated_clock -source inst1|altpll_component|pll|inclk[1] -phase 90.00 -name \
    inst1|altpll_component|pll|clk[1]~1 inst1|altpll_component|pll|clk[1]
```

Alternatively, to manually constrain the PLL in Figure 7, you can specify the following constraints:

```
# create input clock 0
create_clock -period 10 -name clk_100 [get_ports clk_a]

# create input clock 1
create_clock -period 5 -name clk_200 [get_ports clk_b]

# create the PLL output clocks
derive_pll_clocks

# create the generated clocks on the output of PLL clk0
# sourced from inclk0
create_generated_clock -source inst1|altpll_component|pll|inclk[0] -name pll_inclk0_clk0 \
    inst1|altpll_component|pll|clk[0]

# sourced from inclk1
create_generated_clock -source inst1|altpll_component|pll|inclk[1] -name pll_inclk1_clk0 \
    inst1|altpll_component|pll|clk[0] -add

# create the generated clocks on the output of PLL clk1
# sourced from inclk0
create_generated_clock -source inst1|altpll_component|pll|inclk[0] -phase 90.00 -name \
    pll_inclk0_clk1  inst1|altpll_component|pll|clk[1]

# sourced from inclk1
create_generated_clock -source inst1|altpll_component|pll|inclk[1] -phase 90.00 -name \
    pll_inclk1_clk1  inst1|altpll_component|pll|clk[1] -add
```

When manually applying two or more clocks to the same node, `-add` must be included.

Clock Uncertainty

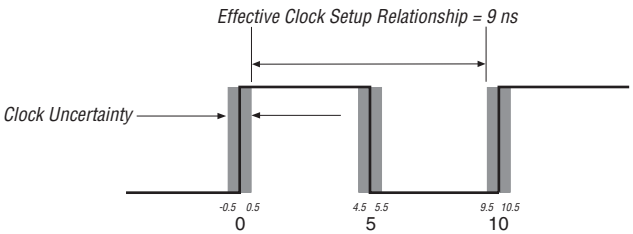
Clock uncertainty is defined as the interval of confidence around the ideal clock value such that the measured value is always inside this stated interval.

The following are the common sources of clock uncertainty:

- Clock jitter
- Clock skew
- Duty cycle distortion
- Phase shift error (when using non-zero PLL phase shifts)

Use the `set_clock_uncertainty` constraint to model jitter, skew, and a guard band associated with clock signals. When a clock uncertainty assignment exists for a clock signal, the TimeQuest Timing Analyzer performs the most conservative setup and hold checks. An uncertainty value of 0.5 ns for `clk0` (with a clock period of 10 ns) means the first rising edge of `clk0` can arrive at any time between –0.5 and 0.5 ns, and the first falling edge of `clk0` can arrive at any time between 4.5 and 5.5 ns. This uncertainty interval effectively reduces the available launch and latch time period to 10 – 1 = 9 ns (Figure 8).

Figure 8. Clock Uncertainty



The `set_clock_uncertainty` constraint does not change the micro t_{SU} and t_H values of the internal registers of the device, but decreases the clock setup and clock hold relationships for any register-to-register paths.

Use `set_clock_uncertainty -setup` to apply a setup uncertainty value or `set_clock_uncertainty -hold` to apply a hold uncertainty value to a clock signal, or as a point-to-point assignment between two clocks. For a setup check, the TimeQuest Timing Analyzer subtracts the specified clock setup uncertainty from the data required time. For a hold check, the TimeQuest Timing Analyzer adds the specified clock hold uncertainty to the data required time.

Figure 9 shows the Data Arrival Path and Data Required Path reports for a setup check performed on a single clock domain register-to-register path. A 10 ns clock drives the clock port of the registers and no uncertainty constraints have been applied to the clock. The data arrival time is 3.331 ns, and the data required time is 12.639 ns. The slack time for this path is 9.308.

Figure 9. Data Arrival and Data Required Times without Clock Uncertainty

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	2.747	2.747	R				clock network delay
3	2.836	0.089		uTco	1	FF_X1_Y43_N39	src_reg
4	2.836	0.000	RR	CELL	2	FF_X1_Y43_N39	src_reg|q
5	3.131	0.295	RR	IC	1	MLABCELL_X1_Y43_N6	dst_reg~feeder|dataf
6	3.230	0.099	RR	CELL	1	MLABCELL_X1_Y43_N6	dst_reg~feeder|combout
7	3.230	0.000	RR	IC	1	FF_X1_Y43_N7	dst_reg|d
8	3.331	0.101	RR	CELL	1	FF_X1_Y43_N7	dst_reg

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.743	2.743	R				clock network delay
3	12.639	-0.104		uTsu	1	FF_X1_Y43_N7	dst_reg

Figure 10 shows the Data Arrival Path and Data Required Path reports for a setup check performed on a single clock domain register-to-register path. A 10 ns clock drives the clock port of the registers and a setup uncertainty of 0.2 ns (Data Required Path line 2) has been applied to the clock. The data arrival time is 3.331 ns (Data Arrival Path line 8), and the data required time is 12.439 ns (Data Required Path line 3). The slack time for this path is 9.108 ns.

Figure 10. Data Arrival and Data Required Times with Clock Uncertainty

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	2.747	2.747	R				clock network delay
3	2.836	0.089		uTco	1	FF_X1_Y43_N39	src_reg
4	2.836	0.000	RR	CELL	2	FF_X1_Y43_N39	src_reglq
5	3.131	0.295	RR	IC	1	MLABCELL_X1_Y43_N6	dst_reg~feeder|dataf
6	3.230	0.099	RR	CELL	1	MLABCELL_X1_Y43_N6	dst_reg~feeder|combout
7	3.230	0.000	RR	IC	1	FF_X1_Y43_N7	dst_regld
8	3.331	0.101	RR	CELL	1	FF_X1_Y43_N7	dst_reg

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.743	2.743	R				clock network delay
3	12.543	-0.200					clock uncertainty
4	12.439	-0.104		uTsu	1	FF_X1_Y43_N7	dst_reg

Conclusion

The Stratix device family PLLs provide complete control of device clocks and system timing. This application note enables you to take advantage of the numerous features and capabilities provided by the Stratix PLLs. Detailed information about all reports and analysis performed by the Quartus II software and the TimeQuest Timing Analyzer is presented, along with examples and guidelines on how to read and understand the various Timing Analysis reports relating to PLLs.

Referenced Documents

This application note references the following documents:

- *altpll Megafunction User Guide*
- *Cyclone III Device Handbook*
- *Cyclone II Device Handbook*
- *Cyclone Device Handbook*
- *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*
- *Stratix III Device Handbook*
- *Stratix II Device Handbook*
- *Stratix Device Handbook*

Document Revision History

Table 6 shows the revision history for this application note.

Table 6. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
August 2007 v1.0	Initial release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com