Implementing Stratix III and Stratix IV Programmable I/O Delay Settings in the Quartus II Software

August 2013, ver. 1.3 Application Note 474

Introduction

Altera® Stratix® III and Stratix IV series devices have a very versatile I/O architecture. Included in the various features of the Stratix III I/O are options for skew management in your systems. The Stratix III I/O has various physical programmable delay elements in its architecture. Using the option to program these delay elements with different settings provides a method for skew management.

Skew management is an advanced technique that can be used in some cases to mitigate timing problems in your system. To take advantage of this technique, you must understand how programmable delay elements in Stratix III I/O devices are presented and assigned in the Quartus® II software. This document describes the programmable delay element settings, the methods to configure them, and some applications in which the programmable delay elements are reprogrammed to overcome a timing problem.

Stratix III I/O programmable delay elements are constructed and staggered to provide you with versatile solutions for coarse and fine tuning of delays. When you tune delays, you are managing the skew between various signals in your system. Circumstances in which you can use skew management with programmable delay elements include:

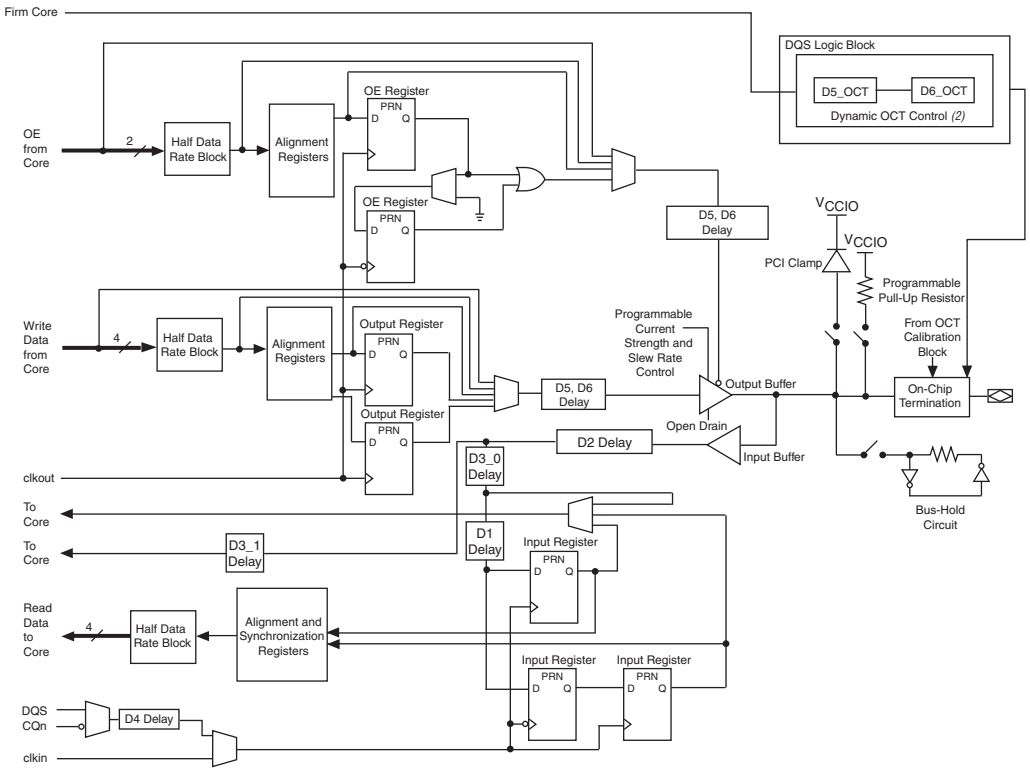
- To skew or deskew data signals to or from a bus of pins to compensate for process variations in the die, package, circuit board, and external interfacing devices.
- To skew simultaneous switching output (SSO) signals to mitigate any simultaneous switching noise (SSN) problems
- To skew or deskew data signals on DDR interfaces to mitigate any DDR read and write problems
- To skew or deskew input data signals to the device to meet timing performance
- To increase or decrease setup times (t_{SU}) within the core
- To increase or decrease clock-to-output (t_{CO}) times within the core timing

This document includes the following topics:

Delay Elements Overview

One of the primary purposes of delay elements in Stratix III I/O is to manage timing margins at the I/O. In today's high speed interfaces, timing margins need to be read and timed properly. Stratix III I/O has a versatile I/O architecture (Figure 1) in which the input and output paths consist of staggered delays to provide fine control of delays.

Figure 1. Delay Elements in Stratix III I/O *Notes (1), (2)*



Notes to Figure 1:

(1) D3_0 and D3_1 delays have the same available settings in the Quartus II software.

(2) One dynamic OCT control is available per DQ/DQS group.

The Quartus II software uses specific names for different programmable delay elements present in the Stratix III I/O. As seen in Figure 1, the input path consists of D1, D2, and D3 delay elements. The output path consists of D5 and D6 delay elements.

"Delay elements" can also be referred to as "delay chains" in this application note and in other Altera documentation. These terms have the same meaning and can be used interchangeably.

Table 1 shows the nomenclature used for the delay elements in the Quartus II software.

Table 1. Nomenclature of Stratix III I/O Delay Elements in the Quartus II Software

Programmable Delay (Parameter)	Description	Settings (Integer)	Delay Range (ps)
D1 *(1)*	I/O Buffer-to-Input Register Delay	0 to 15	150 to 900
D2 *(1)*	I/O Buffer-to-Input Register or Internal Cells Delay	0 to 7	330 to 700
D3 *(1)*, *(4)*	I/O Buffer-to-Internal Cells Delay and I/O Buffer-to-Input Register Path	0 to 7	155 to 2581
D4 *(2)*	DQS Delay Chain Delay	0 to 15	115 to 750
D5 *(2)*, *(5)*	Output Register-to-I/O Buffer Delay	0 to 15	123 to 897
D6 *(5)*	Output Register-to-I/O Buffer Delay	0 to 6	118 to 377
D5_OCT *(3)*	On-Chip Termination (OCT) Calibration Block to I/O Buffer Delay	0 to 15	115 to 865
D6_OCT *(3)*	On-Chip Termination Calibration Block to I/O Buffer Delay	0 to 6	115 to 415

Notes to Table 1:

(1) Applicable for inputs or bidirectional pins.

(2) Applicable for outputs or bidirectional pins.

(3) Applicable for bidirectional pins and only when the dynamic OCT scheme is used. D5_OCT and D6_OCT programmable delays are associated with the OCT control block in Stratix III devices. For any bidirectional I/Os, if your design uses dynamic OCT feature in Stratix III I/Os, the D5 and D6 delay elements settings must match the D5_OCT and D6_OCT delay elements settings. If you alter the D5_OCT or D6_OCT delay settings, you must alter the D5 element, the D6 element, or both elements to match them with D5_OCT and/or D6_OCT delay settings for proper functioning of the I/O as input and output.

(4) In Stratix III devices, there are two delay elements, D3_0 and D3_1, which are parallel and are part of the input buffer to core path. These two chains are represented in Quartus II software reports and the Resource Property Editor tool as D3_0 and D3_1. When creating an assignment using the Assignment Editor, both delay elements are represented as a D3 assignment. The settings and allowable range is identical for both D3_0 and D3_1. When you make D3 delay assignments for a pin, after a successful compilation, you will see both delay elements D3_0 and D3_1 configured with the same settings.

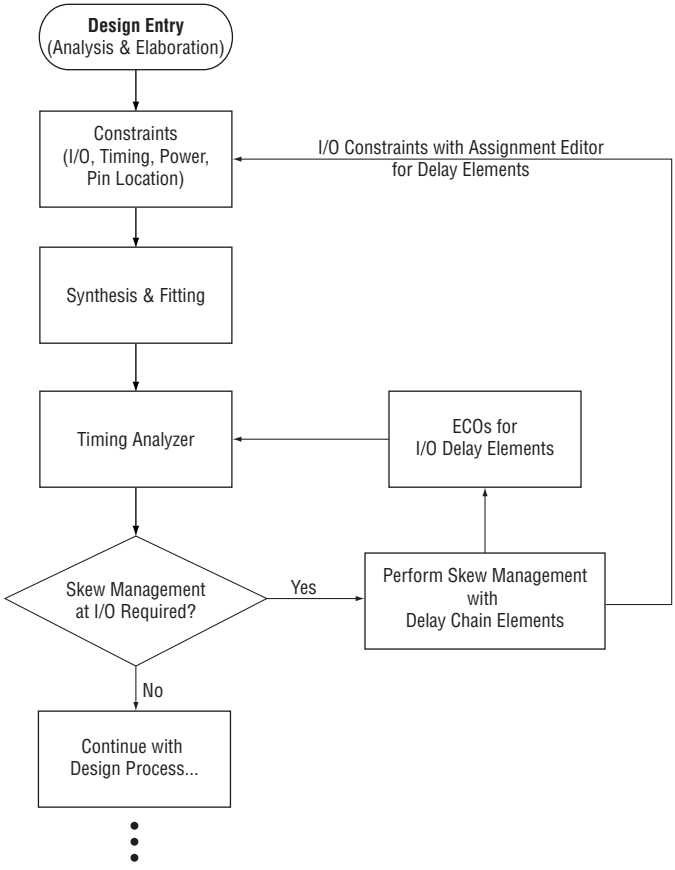
(5) These delays are applied to Output Enable (OE) control, thus affecting switching between input and output.

For details about the Stratix III I/O architecture and its features, refer to the *Stratix III Device Handbook*.

Skew Management in the Design Cycle

In a typical Quartus II design flow (as shown in Figure 2), you would compile your design with timing constraints, pin constraints, output loading constraints, and location constraints. The timing constraints specify the performance of your design; that is, how fast you can run your FPGA circuit. Timing constraints include clock constraints, I/O timing constraints, clock uncertainty margins, and other timing requirements such as false and multicycle path information. Based on the timing constraints you provide, the Quartus II software compiles your design and attempts to meet the timing constraints during the synthesis and fitting stages of compilation. During this process it also configures the programmable delay elements in Stratix III I/Os, with various settings best suited to meet the I/O timing of your design.

Figure 2. Typical Quartus II Design Flow



After you have compiled your design, you may meet timing requirements in your design, but might still want to skew or deskew some I/O signals in your FPGA. In this case, you may want to perform skew management at the I/Os in your design.

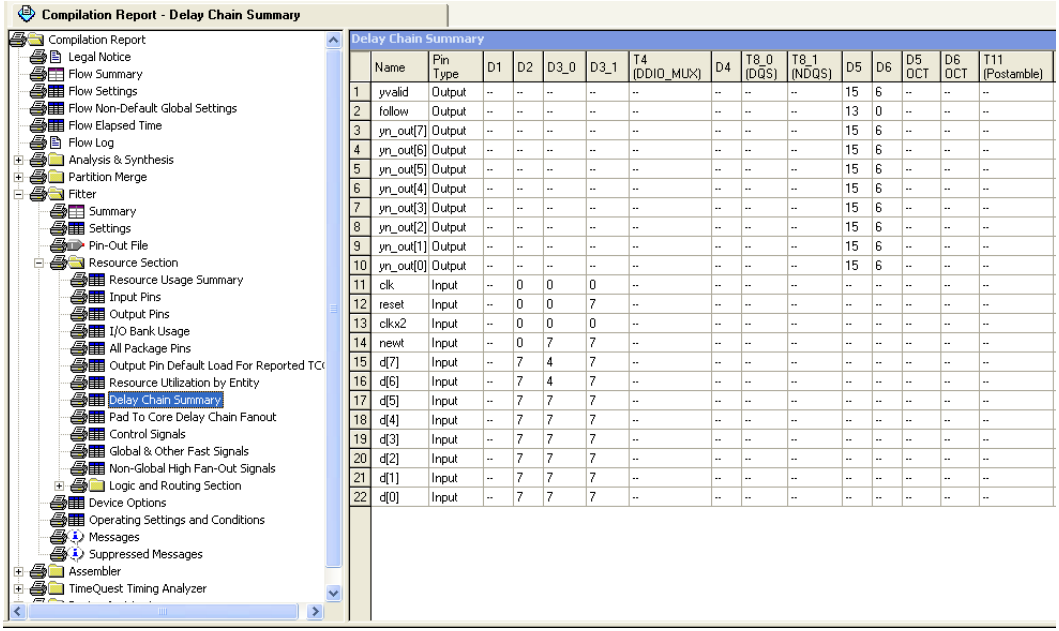
There are various methods to achieve skew management by changing the settings of programmable delays, as detailed in "Changing the Programmable Delay Settings" on page 7. Before using any of these methods, however, you must perform detailed timing analysis and constraint analysis of your design. It is important to understand how timing constraints can affect the delay settings of programmable delay elements during compilation.

To understand how Quartus II software honors various I/O timing constraints during a compilation, refer to delay chain summary reports in Figure 3 on page 6 and Figure 4 on page 6 obtained from running an example design with relaxed and tight timing constraints. Figure 3 on page 6 shows the results after first compilation of the design. Figure 4 on page 6 shows the results after the second compilation, during which the timing constraints have been tightened. In this example design, the locations of all the internal registers in both compilations are fixed by making location assignments in the Assignment Editor, so the I/O timing paths from internal registers to the output pins are fixed.

Because the location of the internal registers was fixed, the fitter chose different (lower) delay settings for the output pins during the second compilation to meet the I/O timing, as shown in Figure 4 on page 6. During the second compilation, the timing constraints have been tightened by increasing the clock frequency. By choosing a lower value of the delay elements, the delay from the output of those fixed internal registers to the I/O buffer was decreased.

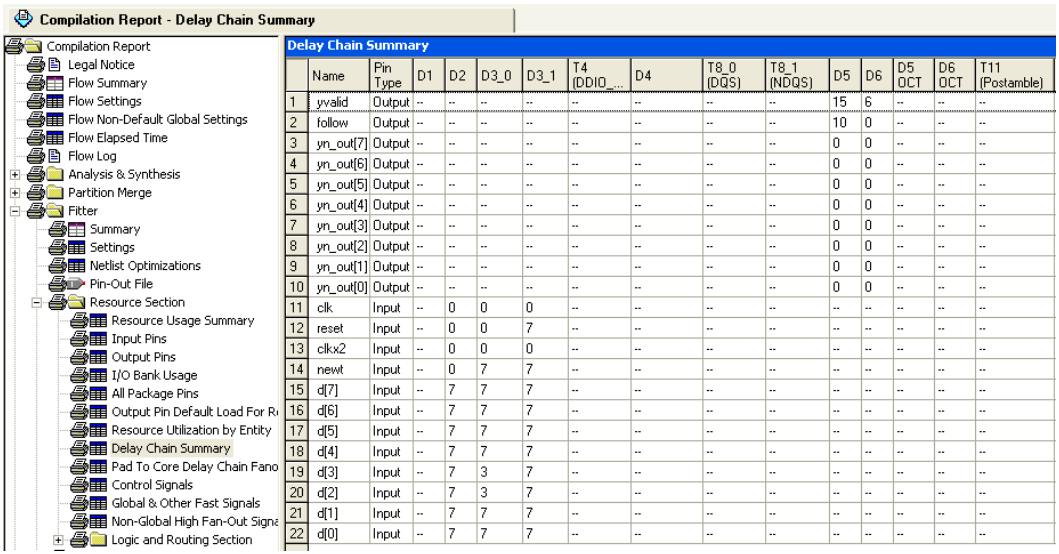
Because the Quartus II software chooses different delay settings based on the timing constraints, it is important that your timing constraints are complete and accurate. Furthermore, during the analyses of your design, you must recognize and understand all of the delay element settings used during a compilation by analyzing the detailed reports printed in the Quartus II software after the compilation.

Figure 3. Delay Chain Summary Report



	Name	Pin Type	D1	D2	D3_0	D3_1	T4 (DDIO_MUX)	D4	T8_0 (DQS)	T8_1 (NDQS)	D5	D6	D5 OCT	D6 OCT	T11 (Postamble)
1	yvalid	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
2	follow	Output	--	--	--	--	--	--	--	--	13	0	--	--	--
3	yn_out[7]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
4	yn_out[6]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
5	yn_out[5]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
6	yn_out[4]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
7	yn_out[3]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
8	yn_out[2]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
9	yn_out[1]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
10	yn_out[0]	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
11	clk	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
12	reset	Input	--	0	0	7	--	--	--	--	--	--	--	--	--
13	clkx2	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
14	newt	Input	--	0	7	7	--	--	--	--	--	--	--	--	--
15	d[7]	Input	--	7	4	7	--	--	--	--	--	--	--	--	--
16	d[6]	Input	--	7	4	7	--	--	--	--	--	--	--	--	--
17	d[5]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
18	d[4]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
19	d[3]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
20	d[2]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
21	d[1]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
22	d[0]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--

Figure 4. Delay Chain Summary Report with Tight Constraints



	Name	Pin Type	D1	D2	D3_0	D3_1	T4 (DDIO_...	D4	T8_0 (DQS)	T8_1 (NDQS)	D5	D6	D5 OCT	D6 OCT	T11 (Postamble)
1	yvalid	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
2	follow	Output	--	--	--	--	--	--	--	--	10	0	--	--	--
3	yn_out[7]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
4	yn_out[6]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
5	yn_out[5]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
6	yn_out[4]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
7	yn_out[3]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
8	yn_out[2]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
9	yn_out[1]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
10	yn_out[0]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
11	clk	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
12	reset	Input	--	0	0	7	--	--	--	--	--	--	--	--	--
13	clkx2	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
14	newt	Input	--	0	7	7	--	--	--	--	--	--	--	--	--
15	d[7]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
16	d[6]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
17	d[5]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
18	d[4]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
19	d[3]	Input	--	7	3	7	--	--	--	--	--	--	--	--	--
20	d[2]	Input	--	7	3	7	--	--	--	--	--	--	--	--	--
21	d[1]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--
22	d[0]	Input	--	7	7	7	--	--	--	--	--	--	--	--	--

Delay Elements Report

Delay elements settings used for all of the I/O pin's delay elements are shown under the **Fitter** report section as the **Delay Chain Summary** report. To view this report, in the **Compilation Report**, click the "+" icon to expand the Fitter folder; beneath it, expand the Resource Section folder, and click **Delay Chain Summary**

The delay chain summary report shows all of the I/O signals in your design and the settings used for each of the I/O's delay elements. As shown in Figure 3, `D1`, `D2`, `D3`, `D4`, `D5`, and `D6` delay elements appear for each of the I/Os. Note that other delay elements, such as `T8` and `T11`, are also present in the **Delay Chain Summary** report. These delays are part of Stratix III I/O architecture, but are preconfigured with settings based on characterization of Stratix III devices. You cannot change these preconfigured delay elements settings.

Also notice that for some I/Os, the delay element column has a "—". This means that the I/O is used in a configuration such that the delay elements settings with a "—" do not apply to it. For example, in the instance of the `d[0]` input pin in Figure 3, the `D1` delay element setting has an em-dash ("—") in its column. This means that because the input register is not used in that I/O, the `D1` delay elements setting, defined as "Input buffer to Input register", is not applicable.

If you have compiled your design and face timing problems that can be mitigated by setting the delay elements settings in the Stratix III I/O, use the methods described in "Changing the Programmable Delay Settings" to change the delay settings.

Changing the Programmable Delay Settings

Although the Quartus II software configures the delay elements settings automatically, there are methods for you to configure or reconfigure the programmable I/O elements. You can change the settings in the Quartus II software during compilation or after compilation. Another option allows you to reconfigure the delays in real time operation.

The following methods can be used to configure or reconfigure the Programmable I/O elements:

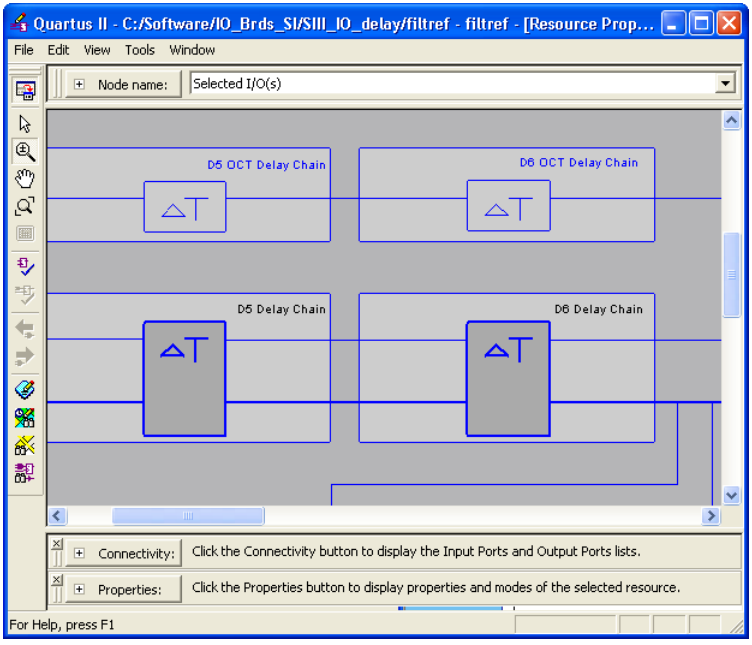
- Using the Resource Property Editor to create an Engineering Change Order (ECO)
- Using the Assignment Editor to create assignments
- Using the ALTIOBUF Megafunction in your RTL code

With the first two methods, you can change the settings in the Quartus II software before or after the compilation. The third option allows you to reconfigure the delays in real-time operation.

Using the Resource Property Editor

You can change the delay elements settings using the Resource Property Editor (RPE) tool in the Quartus II software. This method involves an ECO. You can use the RPE tool to view and edit properties for various resources. You can also change the settings of the delay element or elements in an I/O, after locating the I/O in the RPE tool. Figure 5 shows the view of an I/O delay element using the RPE tool.

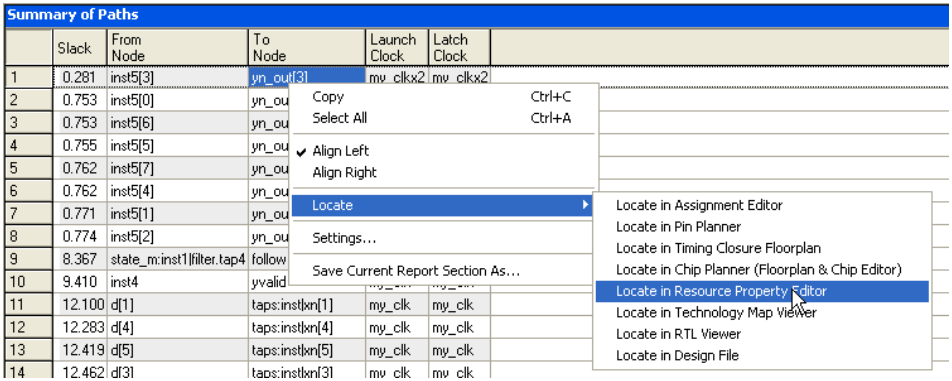
Figure 5. Delay Elements Viewed in the RPE Tool



For more information about how to locate a resource using the RPE tool, refer to the *Engineering Change Management with the Chip Planner* chapter in the *Quartus II Handbook*.

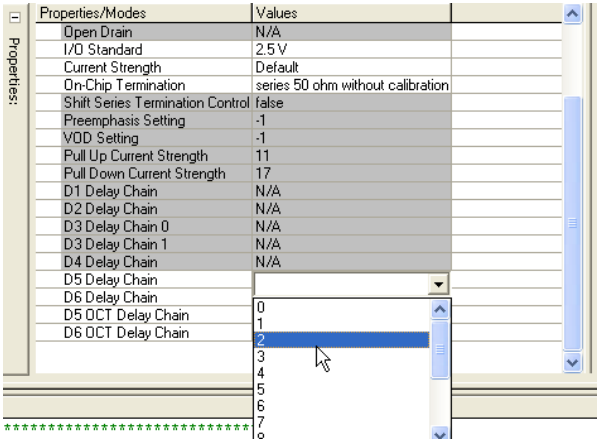
One way to locate an I/O in the RPE tool is to use the cross-probing technique available in the Quartus II software. Cross-probing from timing reports and other tools within the Quartus II software to the RPE tool allows you to easily locate I/Os in the design. For example, you can locate the I/O pins directly from the TimeQuest Timing Analyzer reports. To do this, in the timing report, right-click the I/O pin, point to **Locate**, then click **Locate in Resource Property Editor**, as shown in Figure 6.

Figure 6. Locating an Output Pin in the Resource Property Editor from the TimeQuest Timing Analyzer Report



When using the RPE tool, ensure that the **Properties** window is open. From the **Properties** window, you can see the delay elements and their available settings, as shown in Figure 7.

Figure 7. I/O Delay Element Settings in the Resource Property Editor Tool



You can select different settings for the delay elements as per your requirement. Each setting results in an increase or decrease in the delay. For settings and delay range for each of the delay elements, refer to Table 1 on page 3. When you change these settings, the changes are stored in the **Change Manager** window. To implement these changes as ECOs, click the **Save & Compile All Netlist Changes** menu button in the RPE

tool. When ECOs are compiled, other parts of your design are preserved. ECO compilation is faster, because it compiles the changes in the Change Manager only.

For step-by-step instructions, refer to the *Engineering Change Management with the Chip Planner* chapter in the *Quartus II Handbook*.

When using the RPE tool, some of the delay element settings might be grayed out. This is due to the fact that if the I/O is used as an output, the delay element settings, which are part of the input path, are grayed out. You cannot change the properties of grayed-out delay elements.

You can also enter constraints for delay elements using the Assignment Editor. For more information, refer to "Using the Assignment Editor" on page 13.

If you have assignments made for delay settings in the Assignment Editor or with the Quartus Settings File (**.qsf**), you can still use the RPE tool to change the settings. The changes you make with the RPE tool are not written to the **.qsf** file. These changes are stored in the Change Manager. If you recompile your design, you may have to reapply the stored changes in the Change Manager to implement the ECOs.

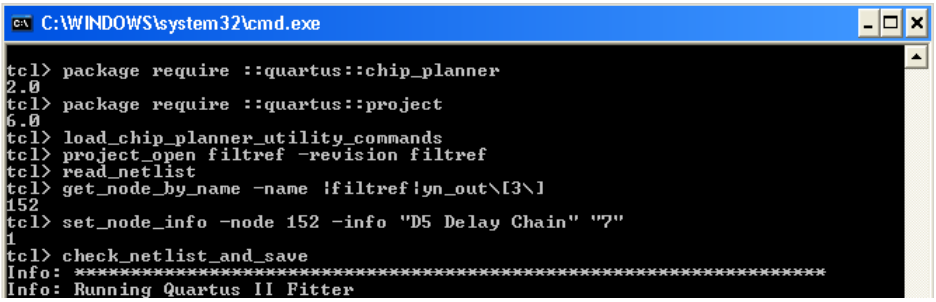
Tcl Support

You can perform the ECO changes for delay element settings with command line operation. To do so, you must use the `quartus_cdb` executable provided with the Quartus II software. After invoking the `quartus_cdb` executable in shell mode, you can issue the commands at the command prompt to perform the ECO. For example, the following procedure shows how to change the delay settings for the `D5` element of an output pin `yn_out[3]` with the `quartus_cdb`:

1. To bring the `quartus_cdb` executable in shell mode, at the windows command prompt, type the following command: `><`*Project_directory*`>/quartus_cdb –s`
2. Load the required packages.
3. Load the project with the proper revision.
4. Read the netlist.
5. Create the ECO.
6. Apply the ECO changes.

Figure 8 shows the previous sequence of commands performed in a windows console to change the delay element setting D5 of output pin yn_out[3].

Figure 8. ECOs in Command Mode to Change the Delay Element Settings In Tcl



You can also write all the commands in a script to perform ECOs and run it with the quartus_cdb executable, as shown in Example 1.

Example 1. Performing ECOs in Tcl Script Format

```
><Project_directory>/quartus_cdb -t <eco_script.tcl>
```

The example script in Example 2 shows steps to perform an ECO for changing the D5 and D6 delay elements setting of the output pin yn_out[3].

Example 2. Changing the D5 Delay Element Setting of the yn_out[3] Output Pin

```
package require ::quartus::chip_planner
package require ::quartus::project
load_chip_planner_utility_commands

project_open filtref -revision filtref
read_netlist
set had_failure 0

set node_id [ get_node_by_name -name |filtref|yn_out\[3\] ]
if { $node_id == -1 } {
     puts "FAIL: get_node_by_name -name |filtref|yn_out\[3\]"
     set had_failure 1
} else {
     set result [ set_node_info -node $node_id -info "D5 Delay Chain" "0" ]
     if { $result == 0 } {
         puts "FAIL (|filtref|yn_out\[3\]): set_node_info -node $node_id -info \"D5 Delay
Chain\" \"0\""
         set had_failure 1
     } else {
         puts "SET  (|filtref|yn_out\[3\]): set_node_info -node $node_id -info \"D5 Delay
Chain\" \"0\""
     }
}

set node_id [ get_node_by_name -name |filtref|yn_out\[3\] ]
if { $node_id == -1 } {
     puts "FAIL: get_node_by_name -name |filtref|yn_out\[3\]"
     set had_failure 1
} else {
     set result [ set_node_info -node $node_id -info "D6 Delay Chain" "0" ]
     if { $result == 0 } {
         puts "FAIL (|filtref|yn_out\[3\]): set_node_info -node $node_id -info \"D6 Delay
Chain\" \"0\""
         set had_failure 1
     } else {
         puts "SET  (|filtref|yn_out\[3\]): set_node_info -node $node_id -info \"D6 Delay
Chain\" \"0\""
     }
}

puts ""
set drc_result [check_netlist_and_save]
if { $drc_result == 1 } {
    puts "check_netlist_and_save: SUCCESS"
} else {
    puts "check_netlist_and_save: FAIL"
}
if { $had_failure == 1 } {
    puts "Not all set operations were successful"
}
project_close
```

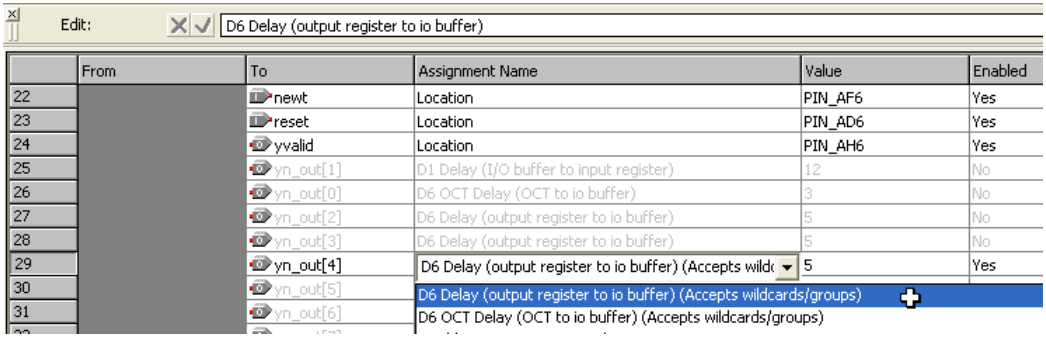
For more information about scripting, command line operation, and Quartus II command line operation, refer to the *Quartus II Scripting Reference Manual*.

Using the Assignment Editor

This method is recommended only after you have performed detailed analyses of your I/O timing paths. When you make assignments, you must recompile your design and run timing analyses to see if it helps in mitigating the timing problem or skew.

You can use the Assignment Editor in the Quartus II software to create assignments. You must choose a particular delay element for a particular I/O pin in the Assignment Editor and select an available setting, as shown in Figure 9. The assignments are represented by **D1** to **D6** assignments.

Figure 9. Delay Element Assignments in the Quartus II Software



To make an assignment using the Assignment Editor, perform the following steps:

1. In the **To** column, enter the I/O pin name. (To find an I/O pin, you can use the **Node Finder** utility.)
2. In the **Assignment Name** column, enter a valid delay assignment. For an input pin, only input delays are available, and vice versa.
3. In the **Value** column, choose a legal value. Refer to Table 1 on page 3 for values allowed for different delays.
4. In the **Enable** column, select **On**. This enables the assignment.

When you use the Assignment Editor in the Quartus II software to create assignments, the assignments are written to the **.qsf** file. To save the assignments, on the File menu, click **Save**. To implement the assignments, you must recompile the design.

Each setting results in an increase or decrease in the delay. For settings and delay range for each of the delay elements, refer to Table 1 on page 3. The value you use for delay element assignment depends on the delay you want in an I/O timing path. When you perform the timing analysis of your I/O path, you can decide which values to choose while making assignments.

In addition to making an assignment for an I/O pin, you can also make assignments for groups of I/O pins. I/O groups are defined as collections of I/O pins, such as input buses and output buses. You can find groups using the **Node Finder** utility. When you make delay assignments for a group, each pin associated with that group gets that assignment.

Tcl Support

In addition to using the Assignment Editor, you can also specify the assignments using Tcl commands in the **.qsf** file.

To set various settings on these delay elements, create the assignment in the ***.qsf** file, as shown in Example 3.

Example 3. Creating Assignments Using Tcl Commands

```
set_instance_assignment -name <delay_element> <available setting> -to <pin_name in design>
```

For example, the assignment shown in Example 4 sets the `D5` delay element for the output pin `yn_out[1]` with the setting `12`.

Example 4. Setting the D5 Delay Element for Output Pin yn_out[1]

```
set_instance_assignment -name D5_DELAY 12 -to yn_out[1]
```

Example 5 shows the assignment made to a group named `d`, which is a 7-bit input bus.

Example 5. Setting the Assignment Made to Group d

```
set_instance_assignment -name D3_DELAY 4 –to d
```

After editing and saving the **.qsf** file, you must recompile the design. After the compilation, the delay chain report in the **Fitter Resource** section of the compilation report shows the values of these delay elements.

You must set the delay settings carefully if you are using the Assignment Editor, because when you set assignments for the delays, the Quartus II software cannot set them during the fitting (compilation) process. The Quartus II software assigns your assigned values to the programmable delays. This is an advanced technique and you must perform detailed timing analyses before assigning any delay values to the programmable delay chains of an I/O.

Using the ALTIOBUF Megafunction

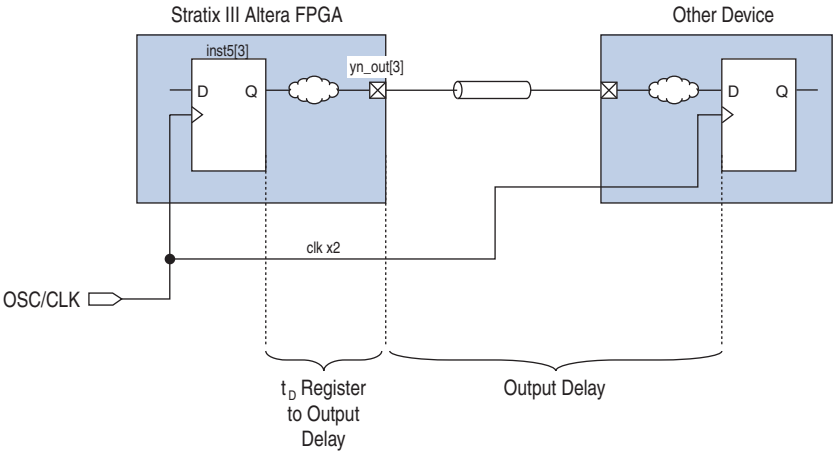
Another way to reconfigure the I/O delay element settings is to use the ALTIOBUF megafunction. You must use the megafunction in your HDL code; the megafunction then becomes part of your design. The ALTIOBUF megafunction allows you to reconfigure the delay elements while the device is in operation.

For more information about the ALTIOBUF megafunction and its design examples, refer to the *I/O Buffer Megafunction User Guide (ALTIOBUF)*.

Example: Managing Skew at Outputs

One application for skew management could be to overcome an unforeseen timing delay problem on the board. In this example, you will see when output timing margins change, and how you can use the Quartus II software to manage the output skew to account for that change without recompiling the design. Figure 10 shows an output pin's delay requirement for one of the design's pins, `yn_out[3]`.

Figure 10. Output Timing Requirement at the FPGA



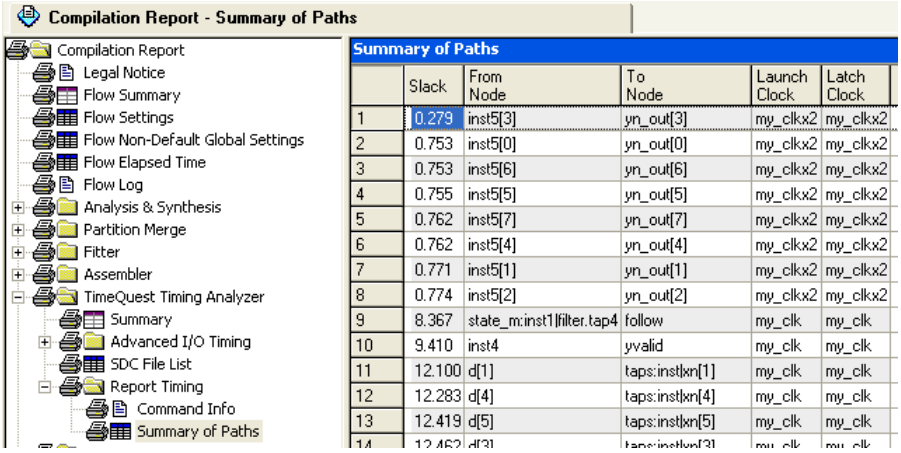
In Figure 10, the output timing path is formed from the `inst5[3]` register to the `yn_out[3]` pin. The output delay for pin `yn_out[3]` is 3 ns. The frequency of clock `clkx2` is 100 Mhz, resulting in a time period of 10 ns. The clock `clkx2` and the output timing requirements for pin `yn_out[3]` were provided to the Quartus II software using the SDC command, shown in Example 6.

Example 6.

```
create_clock -name my_clkx2 -period 10 -waveform [list 0 5] \
   [get_ports clkx2]
set_output_delay 3 -clock my_clkx2 [get_ports yn_out[3]]
```

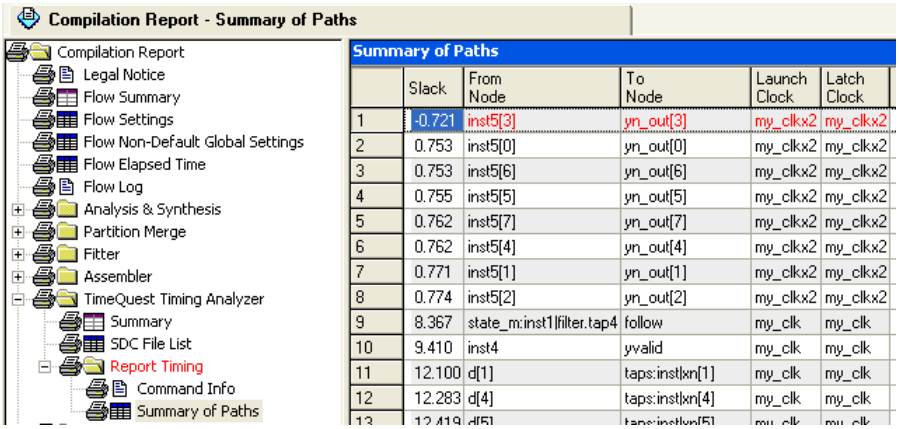
After compiling the design, the timing is met, as shown in Figure 11. The timing path from the internal register `inst5[3]` to the output pin `yn_out[3]` is met with a positive slack.

Figure 11. I/O Timing Path from Internal Register inst5[3] to yn_out[3]



Assume that during the design and debug process the timing requirement for pin `yn_out[3]` has changed to 3.5 ns from 3.0 ns. After running timing analysis with the new timing requirement of 3.5 ns, the timing path will not meet the new timing requirement and fails with a negative slack, as shown in Figure 12.

Figure 12. I/O Timing Path Violation with New Timing Requirement



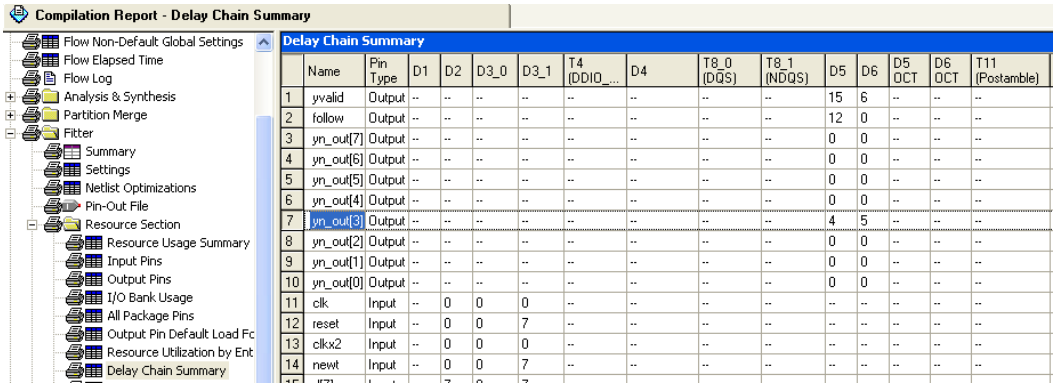
One way to fix this problem and meet the new timing requirement within the FPGA is to decrease the data delay (t_D) from the `inst5[3]` register to the `yn_out[3]` output pin. One method is to enter a new SDC timing

constraint for the yn_out[3] output pin and recompile the design. Another method is to use the RPE tool to change the delay element settings of yn_out[3] output, such that the data delay (t_D) shown in Figure 12 is decreased.

Instead of recompiling the design, the change in delay is achieved by changing the programmable delay settings D5 and D6 for the pin yn_out[3]. The available range for D5 or D6 are shown in Table 1 on page 3.

Figure 13 shows the D5 and D6 delay element settings initially used for pin yn_out[3].

Figure 13. Delay Chain Summary for yn_out[3]

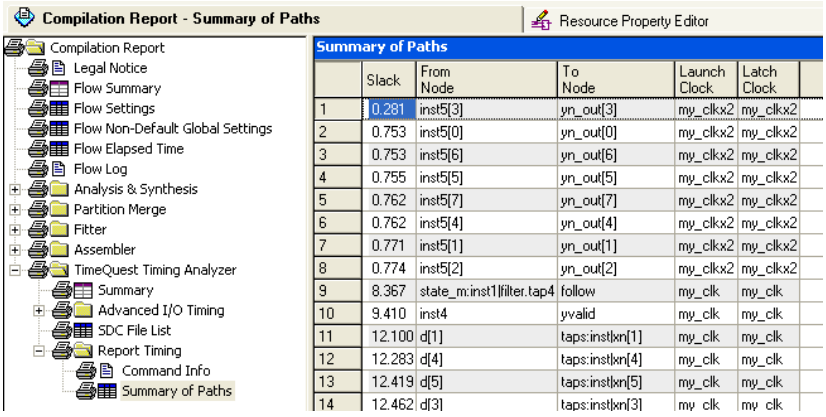


Delay Chain Summary

	Name	Pin Type	D1	D2	D3_0	D3_1	T4 (DDIO_...	D4	T8_0 (DQS)	T8_1 (NDQS)	D5	D6	D5 OCT	D6 OCT	T11 (Postamble)
1	yvalid	Output	--	--	--	--	--	--	--	--	15	6	--	--	--
2	follow	Output	--	--	--	--	--	--	--	--	12	0	--	--	--
3	yn_out[7]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
4	yn_out[6]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
5	yn_out[5]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
6	yn_out[4]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
7	yn_out[3]	Output	--	--	--	--	--	--	--	--	4	5	--	--	--
8	yn_out[2]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
9	yn_out[1]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
10	yn_out[0]	Output	--	--	--	--	--	--	--	--	0	0	--	--	--
11	clk	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
12	reset	Input	--	0	0	7	--	--	--	--	--	--	--	--	--
13	clkx2	Input	--	0	0	0	--	--	--	--	--	--	--	--	--
14	newt	Input	--	0	0	7	--	--	--	--	--	--	--	--	--

In the RPE tool, a setting of 0 is set for D5 and D6 delay elements for the yn_out[3] output pin. After implementing this change as an ECO, the new timing requirement for yn_out[3] is met, as shown in Figure 14.

Figure 14. Delay Chain Summary with the New Timing Requirement Met



	Slack	From Node	To Node	Launch Clock	Latch Clock
1	0.281	inst5[3]	yn_out[3]	my_clkx2	my_clkx2
2	0.753	inst5[0]	yn_out[0]	my_clkx2	my_clkx2
3	0.753	inst5[6]	yn_out[6]	my_clkx2	my_clkx2
4	0.755	inst5[5]	yn_out[5]	my_clkx2	my_clkx2
5	0.762	inst5[7]	yn_out[7]	my_clkx2	my_clkx2
6	0.762	inst5[4]	yn_out[4]	my_clkx2	my_clkx2
7	0.771	inst5[1]	yn_out[1]	my_clkx2	my_clkx2
8	0.774	inst5[2]	yn_out[2]	my_clkx2	my_clkx2
9	8.367	state_m:inst1|filter.tap4	follow	my_clk	my_clk
10	9.410	inst4	yvalid	my_clk	my_clk
11	12.100	d[1]	taps:inst|xn[1]	my_clk	my_clk
12	12.283	d[4]	taps:inst|xn[4]	my_clk	my_clk
13	12.419	d[5]	taps:inst|xn[5]	my_clk	my_clk
14	12.462	d[3]	taps:inst|xn[3]	my_clk	my_clk

The example illustrates how you can manage unforeseen timing problems or manage skew in your FPGA designs.

If you decide to use any of the methods to alter the programmable delay element settings during the design process as described in this application note, it is very important to plan ahead. If you use the Assignment Editor or the Resource Property Editor, you do not have to modify your HDL code. If you use the dynamic method to alter your design, you must embed the ALTIOBUF megafunction in your HDL code. If you use the ALTIOBUF megafunction, you can still use the methods described here to achieve your desired settings.

Conclusion

Using programmable delays in Stratix III I/Os with the Quartus II software gives you flexibility to manage skew. Reconfiguring the programmable delays is an advanced technique and must be completed after careful analysis of your constraints and design. Various tools in the Quartus II software can be used to reconfigure the programmable delays.

Referenced Documents

This application note references the following documents:

- *I/O Buffer Megafunction User Guide (ALTIOBUF)*
- *Engineering Change Management with the Chip Planner* chapter in the *Quartus II Handbook*
- *Quartus II Scripting Reference Manual*
- *Stratix III Device Handbook*

Document Revision History

Table 2 shows the revision history for this application note.

Table 2. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
August 2013 v1.3	Added Stratix IV devices to this application note.	—
March 2008 v1.2	Fixed broken hypertext link	—
February 2008 v1.1	Changed the title of this application note.	—
November 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