SmartTime for Libero SoC v11.8 (Classic Constraint Flow)

SmartFusion2, IGLOO2, and RTG4

User Guide

NOTE: PDF files are intended to be viewed on the printed page; links and cross-references in this PDF file may point to external files and generate an error when clicked. **View the online help included with software to enable all linked content.**

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email:
sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at **www.microsemi.com.**

5-02-00629-2/01.17

Table of Contents

Welcome to SmartTime

About SmartTime

SmartTime is a gate-level static timing analysis (STA) tool for the SmartFusion2, IGLOO2, and RTG4 families. With SmartTime, you can enter timing constraints and perform complete timing analysis of your design to ensure that you meet all timing constraints and that your design operates at the desired speed with the right amount of margin across all operating conditions.

Note: SmartTime in the Enhanced Constraint Flow has changed. Creation and Editing of timing constraints are now handled in a separate Timing Constraints Editor. See the Timing Constraints Editor for help with creating and editing timing constraints in the Enhanced Constraints Flow.

Static Timing Analysis (STA)

Static timing analysis (STA) offers an efficient technique for identifying timing violations in your design and ensuring that it meets all your timing requirements. You can communicate timing requirements and timing exceptions to the system by setting timing constraints. A static timing analysis tool will then check and report setup and hold violations as well as violations on specific path requirements.

STA is particularly well suited for traditional synchronous designs. The main advantage of STA is that unlike dynamic simulation, it does not require input vectors. It covers all possible paths in the design and does all the above with relatively low run-time requirements.

The major disadvantage of STA is that the STA tools do not automatically detect false paths in their algorithms as it reports all possible paths, including false paths, in the design. False paths are timing paths in the design that do not propagate a signal. To get a true and useful timing analysis, you need to identify those false paths, if any, as false path constraints to the STA tool and exclude them from timing considerations.

Timing Constraints

SmartTime supports a range of timing constraints to provide useful analysis and efficient timing-driven layout. SmartTime also includes a constraint checker that validates the constraints in the database.

Timing Analysis

SmartTime provides a selection of analysis types that enable you to:

- Find the minimum cycle time that does not result in a timing violation
- Identify paths with timing violations
- Analyze delays of paths that have no timing constraints
- Perform inter-clock domain timing verification
- Perform maximum and minimum delay analysis for setup and hold checks

To improve the accuracy of the results, SmartTime evaluates clock skew during timing analysis by individually computing clock insertion delays for each register.

SmartTime checks the timing requirements for violations while evaluating timing exceptions (such as multicycle or false paths).

SmartTime and Place and Route

Timing constraints impact analysis and place and route the same way. As a result, adding and editing your timing constraints in SmartTime is the best way to achieve optimum performance.

See Also

Starting and Closing SmartTime

Starting and Closing SmartTime (Enhanced Constraint Flow)

SmartTime Components

Components of SmartTime Timing Analyzer

Changing SmartTime Preferences

Design Flows with SmartTime

You can access SmartTime in Libero SoC either implicitly or explicitly during the following phases of design implementation:

- After Compile – Run SmartTime to add or modify timing constraints or to perform pre-layout timing analysis. In the Libero SoC Design Flow window, expand **Implement Design > Place and Route > Timing Constraints**. Choose **Open Interactively** to enter Timing Constraints.
- During Place and Route – When you select timing-driven place-and-route, SmartTime runs in the background to provide accurate timing information.
- After Place and Route – Run SmartTime to perform post-layout timing analysis and adjust timing constraints. In the Libero SoC Design Flow window, expand **Implement Design > Verify Post-Layout Implementation**. Right-click **Verify Timing** and choose **Open Interactively**.
- During Back-Annotation – SmartTime runs in the background to generate the SDF file for timing simulation.

You can also run SmartTime whenever you need to generate timing reports, regardless of which design implementation phase you are in.

See Also

Compile

Layout

Back-Annotation

Starting and Closing SmartTime - SmartFusion2, IGLOO2, RTG4

You must complete the layout of your design before using SmartTime. If you have not completed layout, the software executes the layout step for you before starting SmartTime.

To edit timing constraints in SmartTime, in the Design Flow window under **Constrain Design** double-click **Edit Timing Constraints**.

To verify timing, in **Implement Design > Verify Post Layout Implementation** right-click **Open SmartTime** and choose **Open Interactively**.

SmartTime reads your design and displays post- or pre-layout timing information.

To close SmartTime, from the **File** menu, choose **Exit**.

To save changes to your design, from the **File** menu, choose **Commit**.

See Also

Importing Files

Compiling your design

Running Layout

SmartTime Components

SmartTime is composed of two main tools:

- The SmartTime Constraints Editor enables you to view and edit timing constraints in your design. Constraints are sorted by category (requirements and exceptions) and by constraint type.
- The Maximum Delay Analysis View  and the Minimum Delay Analysis View  enable you to analyze your design

With SmartTime, you can:

- Browse through your design's various clock domains to examine the timing paths and identify those that violate your timing requirements
- Add and modify timing requirements and exceptions
- Set constraints on a specific pin or a specific set of paths
- Create customizable timing reports
- Navigate directly to the paths responsible for violating your timing requirements

SmartTime Constraint Scenario - SmartFusion2, IGLOO2, RTG4 (Classic Constraint Flow Only)

A constraint scenario is an independent set of constraints. By default a scenario is created as *Primary Scenario* to hold all timing constraints defined by the user. This scenario is used during both analysis and TDPR. You can create multiple scenarios. The scenario used for analysis and the scenario used for TDPR can be selected from a list. Only one scenario can be used for analysis at a time. If multiple scenarios are created they are displayed in separate Constraint Editor windows.

The scenarios window lists all timing constraints scenarios available for the current design.

To create a new scenario, from the Constraints Editor, choose **Tools > Scenarios > New Scenarios**.

The icon  indicates it is the Primary Scenario. It is the default scenario when the Constraints Editor opens.

The new scenario option is also available from **Tools> Scenario > New Scenario**.

You may click the undock icon  at the upper right hand corner to undock a scenario window.

New scenarios are named Scenario_1, Scenario_2 and so on by default when they are first created. From the scenarios window you can select a scenario and from the right-click menu, select:

- **Use for Analysis:** : to use the selected scenario for Timing Analyzer. This command is also available from the Advanced tab in the SmartTime Options dialog box
- **Use for TDPR**: : to use the selected scenario for Timing-driven Layout. This command is also available from the Advanced tab in the SmartTime Options dialog box.
- **Clone scenario**: to create a new scenario with a set of constraints based on an existing scenario
- **Delete scenario**: to delete the selected scenario
- **Rename scenario**: to rename the selected scenario
- **New Scenario**: to create a new scenario

Setting SmartTime Options - SmartFusion2, IGLOO2, RTG4

You can modify SmartTime options for timing analysis by using the SmartTime Options dialog box.

To set SmartTime options:

1. From the SmartTime Maximum/Minimum Delay Analysis View window, choose **Tools> Options**.
   The **SmartTime Options** dialog box has three categories: **General**, **Analysis** and **Advanced**.
2. In the **General** category, select the settings for the operating conditions. SmartTime performs maximum or minimum delay analysis based on the Best, Typical, or Worst case.
3. Check or uncheck whether you want SmartTime to use inter-clock domains in calculations for timing analysis.
4. Click **Restore Defaults** only if you want the settings in the General pane to revert to their default settings.
5. Click **Analysis** to display the options you can modify in the Analysis view.
6. Enter a number greater than 1 to specify the maximum number of paths to include in a path set during timing analysis.

7. Check or uncheck whether to filter the paths by slack value. If you check this box, you must then specify the slack range between minimum slack and maximum slack.
8. Check or uncheck whether to include clock network details.
9. Enter a number greater than 1 to specify the number of parallel paths in the expanded path.
10. Click **Restore Defaults** only if you want the settings in the Analysis View pane to revert to their default settings.
11. Click **Advanced** to display advanced options.
12. Check or uncheck whether to use loopback in bidirectional buffers (bibufs) and/or break paths at asynchronous pins. Check or uncheck whether to disable non-unate arcs in the clock path. If using **Scenarios**, pick the appropriate scenario for timing analysis and timing driven place- and- route.
13. Click **Restore Defaults** only if you want the settings in the Advanced pane to revert to their default settings.
14. Click **OK.**

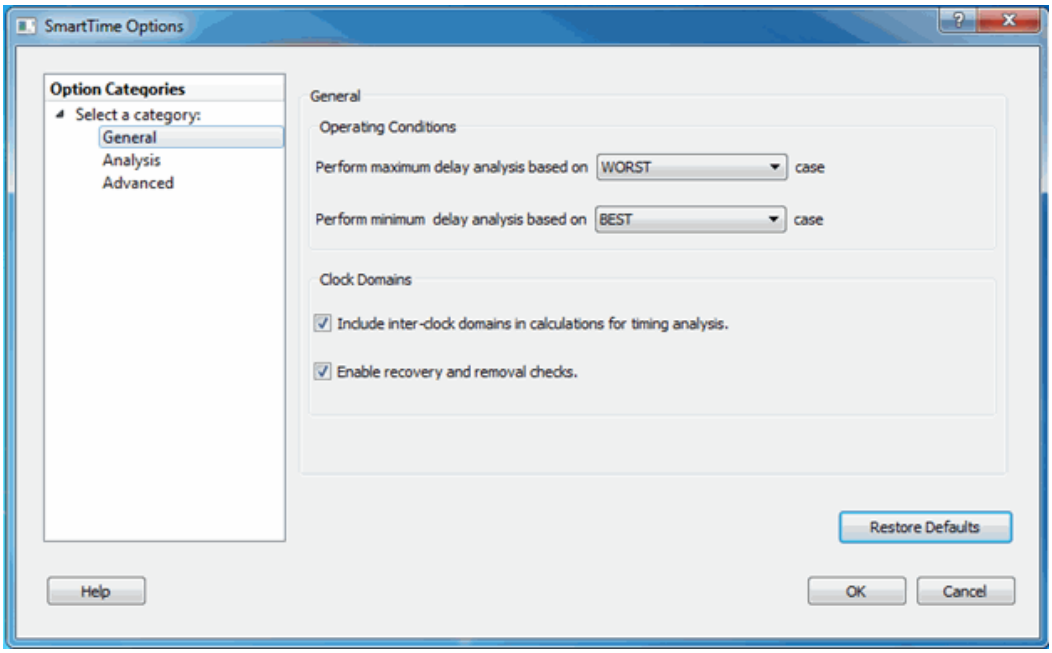


Figure 1 · SmartTime Options Dialog Box – General Options

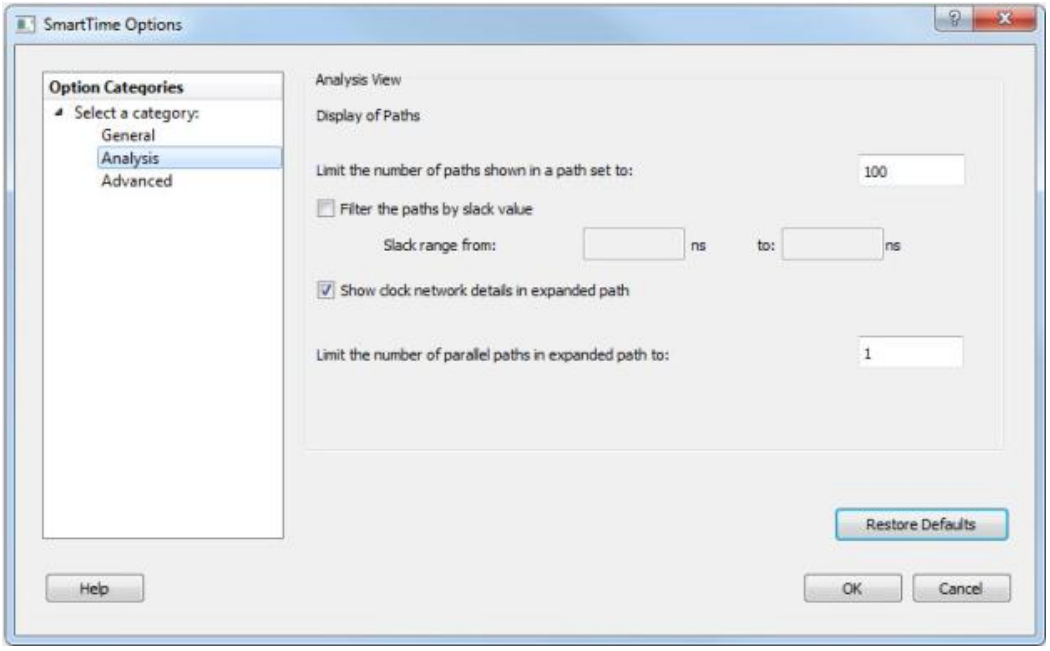


Figure 2 · SmartTime Options Dialog Box – Analysis Options

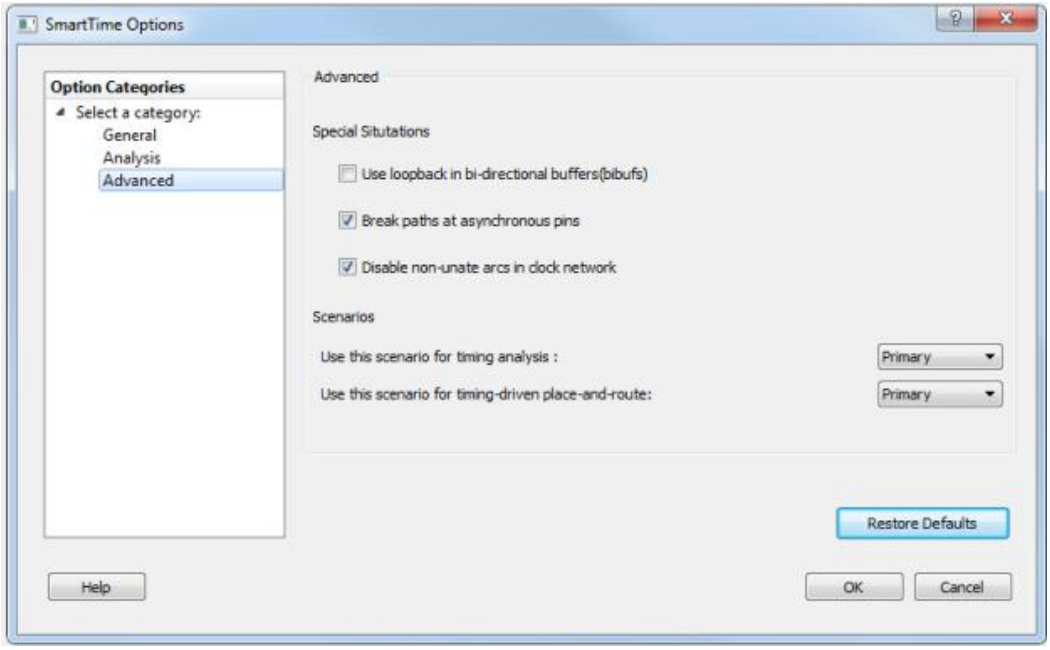


Figure 3 · SmartTime Options Dialog Box – Advanced Options

See Also

SmartTime Options Dialog Box

SmartTime Tutorial - Libero SoC for SmartFusion2, IGLOO2, and RTG4

The following tutorials explore common SmartTime features with example designs:

- **Tutorial 1 - 32-Bit Shift Register with Clock Enable**

Goal: Learn how to apply a clock constraint, perform maximum delay analysis & minimum delay analysis. Also, use the feature to dynamically update timing analysis by changing constraints in the constraints editor.

- **Tutorial 2 – Count16 Counter**

Goal: Import timing constraints file (SDC)/Add Clock Constraint, add input delay and output delay constraints. Create filters abd user sets to isolate design paths for analysis in SmartTime.

- **Tutorial 3 - Design Using Both Clock Edges**

Goal: Learn how to apply a clock constraint, perform maximum delay analysis for a design using both edges of the clock (rising & falling). Generate a custom timing report using SmartTime.

- **Tutorial 4 - False Path Constraints**

Goal: Add false path constraints to identify non-timing critical design paths.

- **Tutorial 5 - Cross Clock Domain Analysis**

Goal: Analyze the timing results for a design with cross clock domain paths.

Tutorial 1 - 32-Bit Shift Register with Clock Enable

This tutorial section describes how to enter a clock constraint for the 32-bit shift register shown. You will use the SmartTime Constraints Editor and perform post-layout timing analysis using the SmartTime Timing Analyzer.

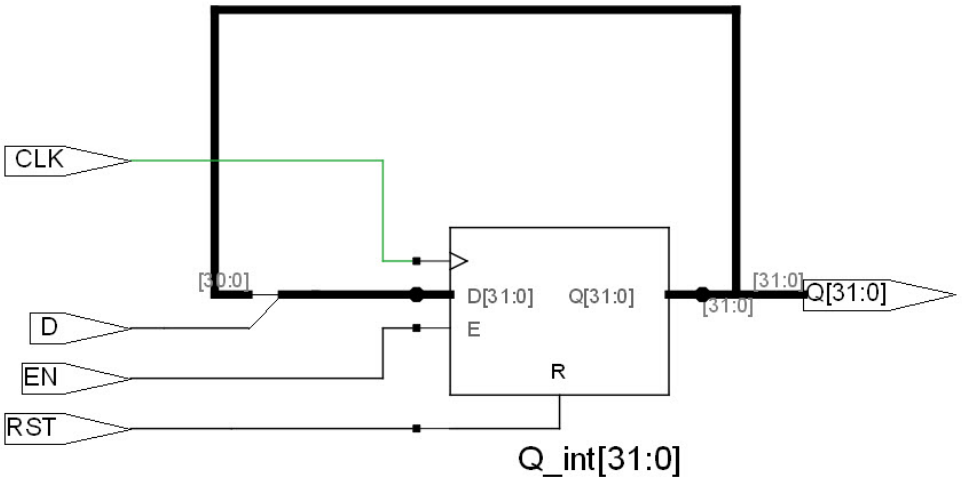


Figure 4 · 32-bit Shift Register

Use the links below to go directly to a topic:

- Add a Clock Constraint
- Run Place and Route

- Maximum Delay Analysis with Timing Analyzer
- Minimum Delay Analysis with Timing Analyzer
- Changing Constraints and Observing Results

To set up your project:

1. Invoke Libero SoC. From the **Project** menu, choose **New Project**.
2. Enter **shift32** for your new project name and browse to a folder for your project location.
3. Select **Verilog** as the Preferred HDL Type.
4. Leave all other settings at the default values.

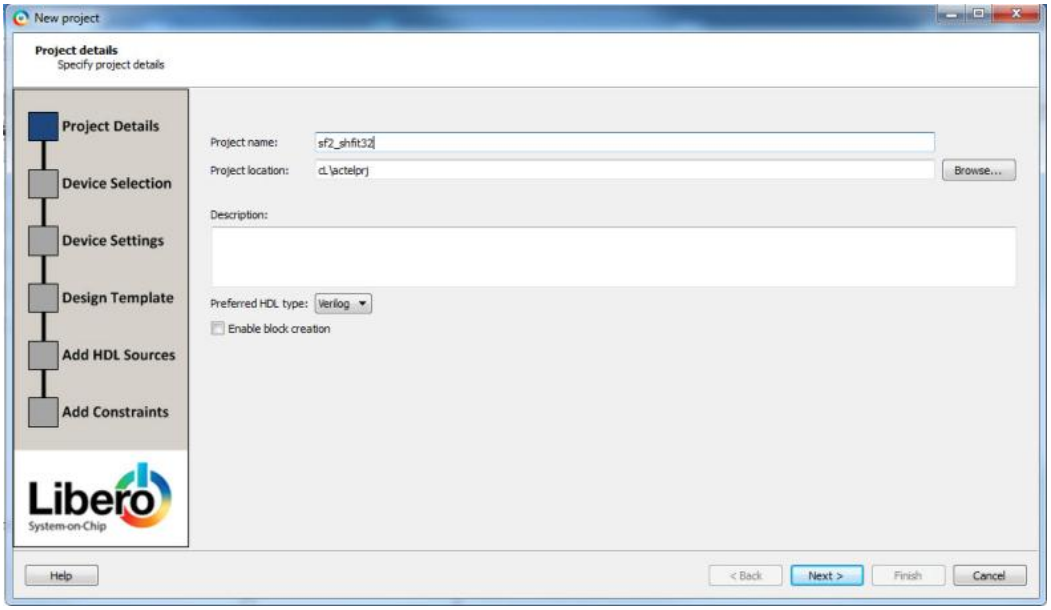
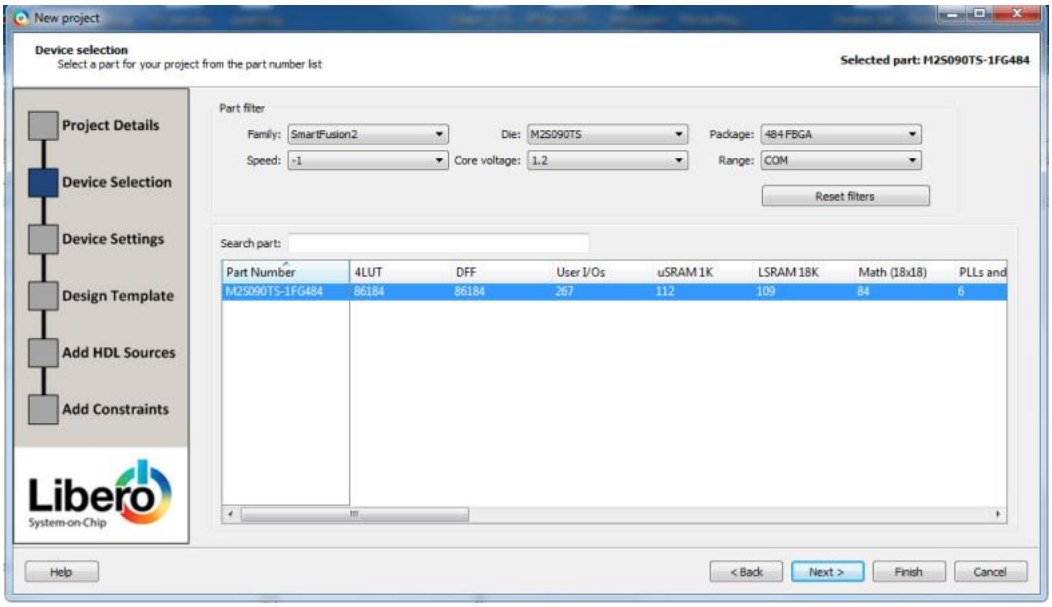
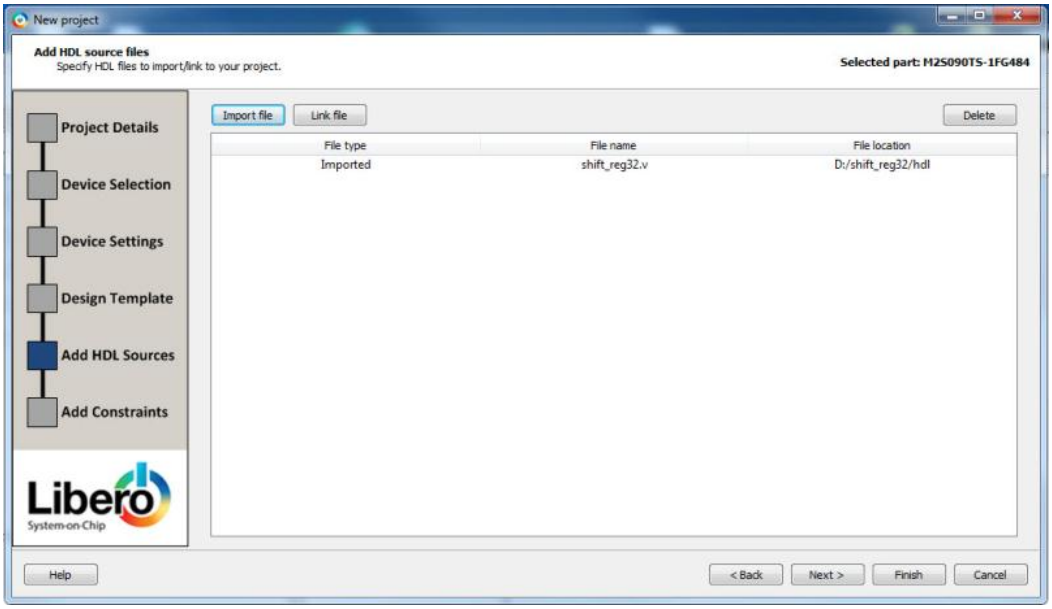


Figure 5 · New Project Creation - 32 Bit Shift Register

5. Click Next to go to Device Selection page. Make the following selection from the pull-down menu:
   - **Family**: SmartFusion2
   - **Die**: M2S090TS
   - **Package**: 484FBGA
   - **Speed**:STD
   - **Core Voltage**: 1.2 V
   - **Range**: COM
6. Click the M2S090TS-1FG484 part number and click **Next**.



7. Accept the default settings in the Device Settings page and click **Next**.
8. Accept the default settings in the Design Template page and click **Next**.
9. In the Add HDL source files page, click **Import file** to import the source file, Navigate to the location of the source Verilog file for the 32-bit shift register you have downloaded from the Microsemi website. Click to select the source file and click **Open**. After project creation, the source Verilog file you import will appear in the project's hdl folder under the File tab.



10. Click **Next** to go to the Add Constraints Page.
11. We are not adding any constraints. Click **Finish** to exit the New Project Creation wizard.
12. After you have created the project, confirm that the imported Verilog source file appears in the Files window, as shown in the figure below.

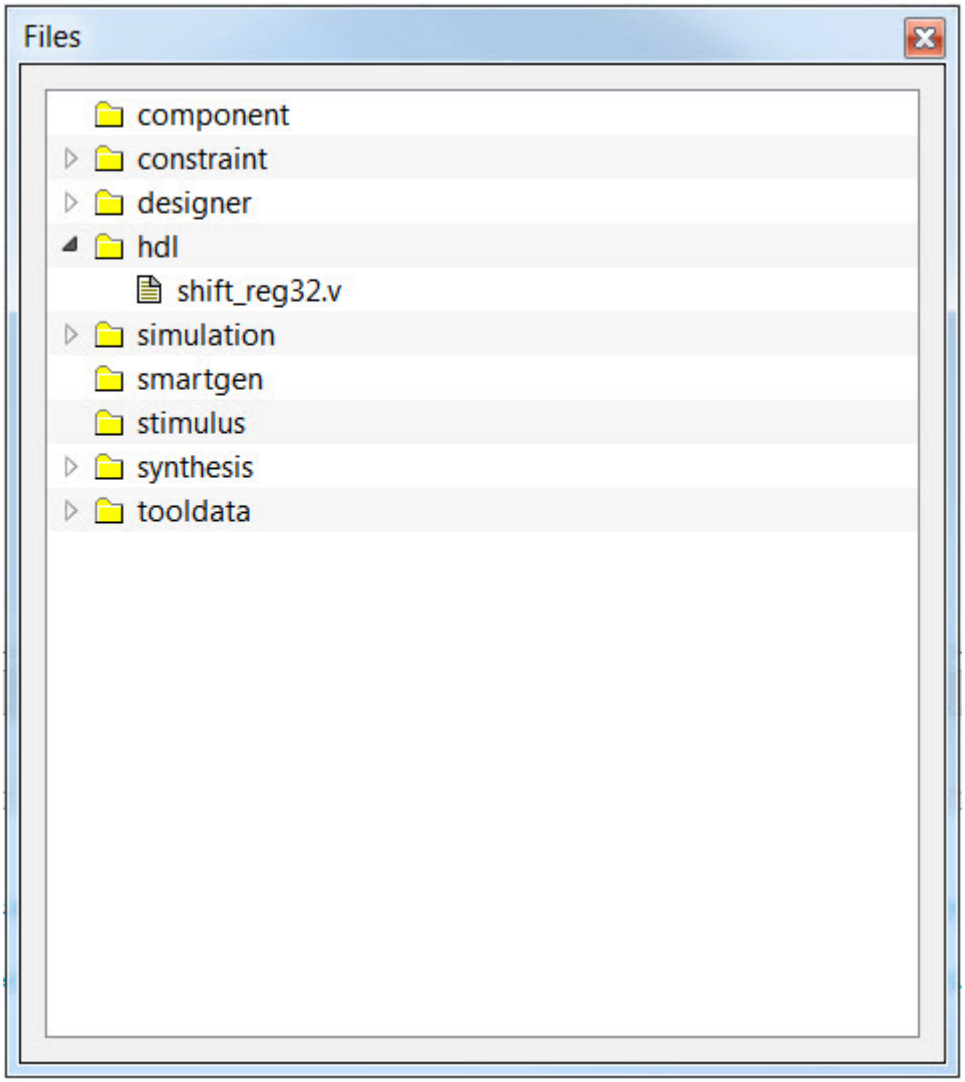


Figure 6 · HDL File shift_reg32.v in the Libero SoC File Window

13. Confirm that the shift_reg32 design appears in the Design Hierarchy window, as shown in the figure below.

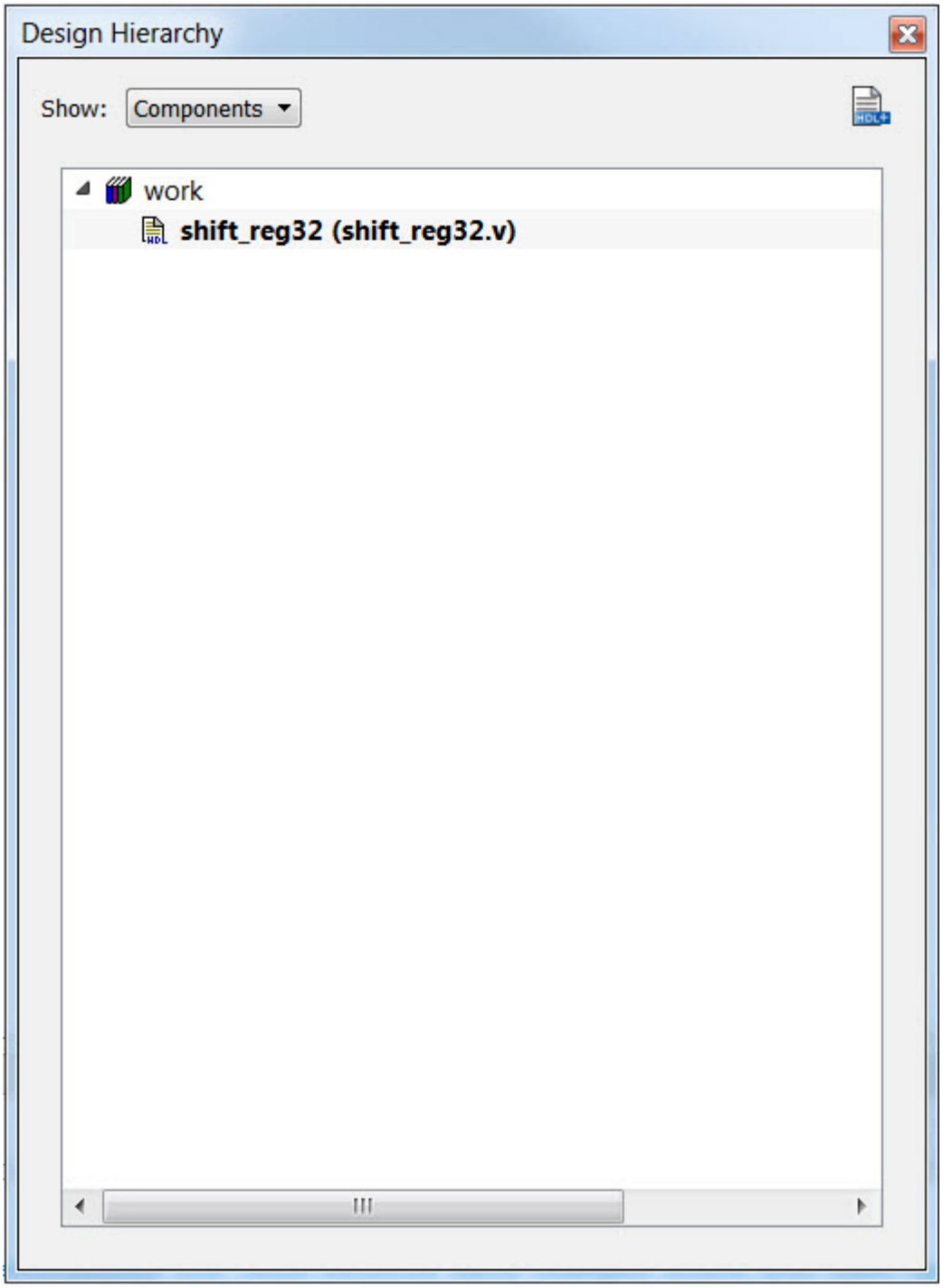


Figure 7 · shift_reg32 in the Design Hierarchy Window

14. In the Design Flow window, double-click **Synthesize** to run Synplify Pro with default settings. A green check marks appears next to Synthesize when Synthesis is successful (as shown in the figure below).
15. Double-click **Compile** in the Design Flow window to run Compile with default settings. A green check mark appears next to Compile when it completes successfully (as shown in the figure below).

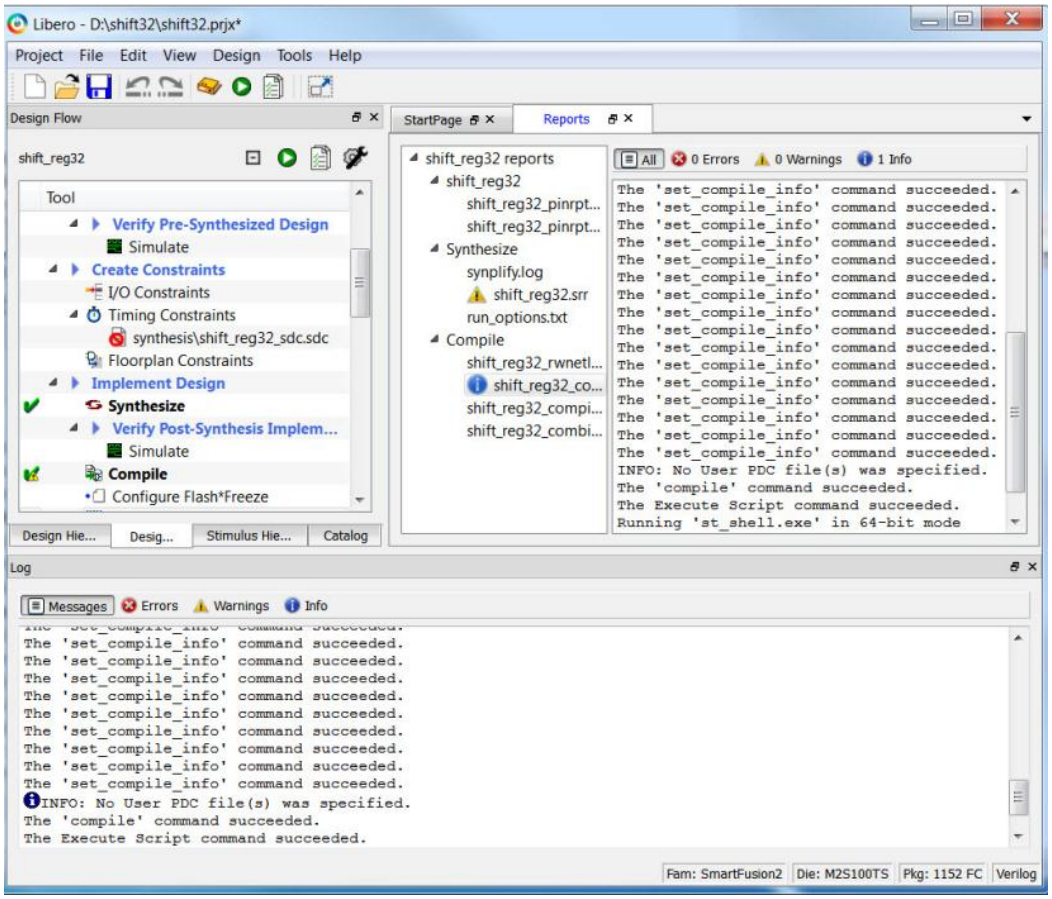


Figure 8 · Synthesis and Compile Complete - 32-Bit Shift Register with Clock Enable

Add a Clock Constraint - 32 Bit Shift Register

To add a clock constraint to your design:

1. In the Design Flow window, expand **Edit Constraints** and double-click **Timing Constraints** to open the Constraints Editor (as shown in the figure below.)

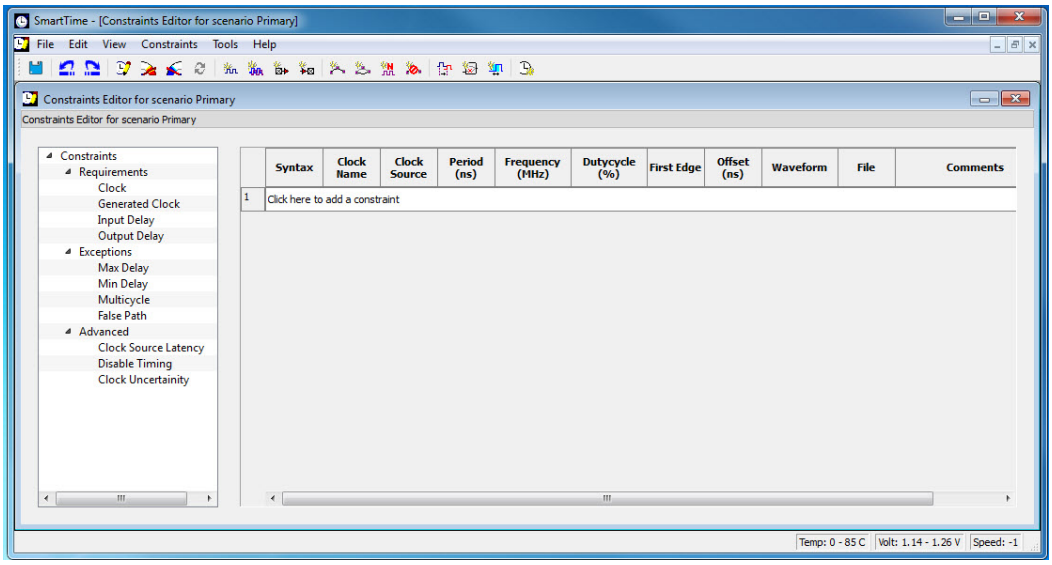


Figure 9 · SmartTime Constraints Editor

2. In the left pane, under Constraints> Requirements, right-click **Clock** and choose **Add Clock Constraint** to open the Create Clock Constraint dialog box (as shown in the figure below).

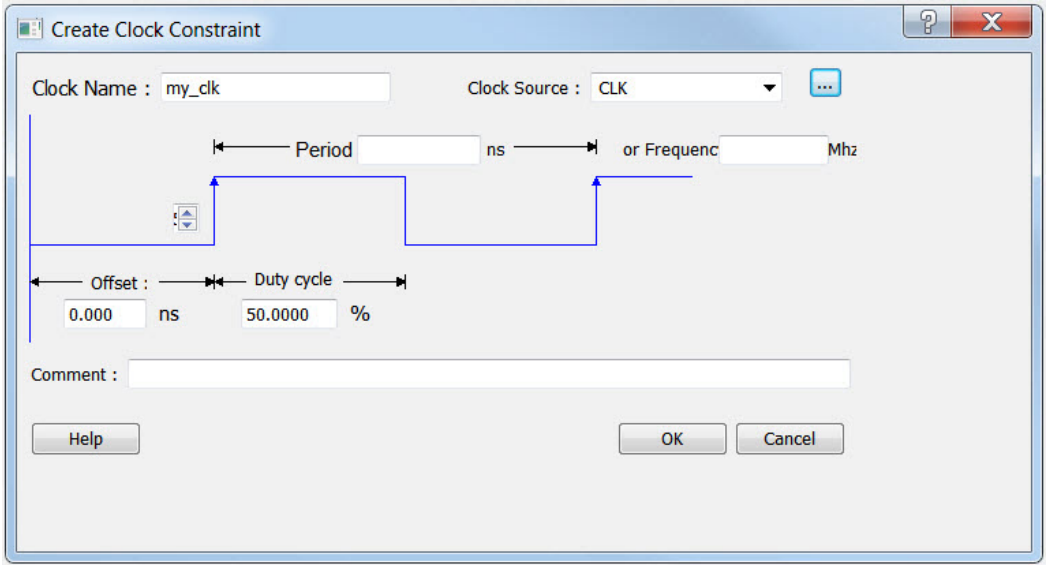


Figure 10 · Create Clock Constraint Dialog Box

3. From the **Clock Source** drop-down menu, choose the CLK pin.
4. Enter **my_clk** in the Clock Name field.
5. Set the Frequency to 800 MHz (as shown in the figure below) and leave all other values at the default settings. Click **OK** to continue.

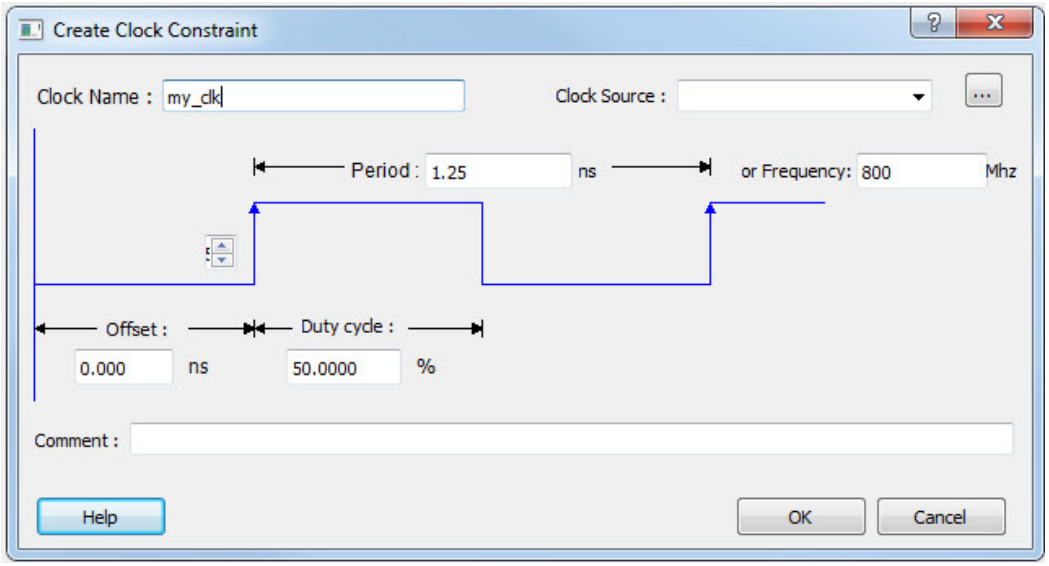


Figure 11 · Add a 800 MHz Clock Constraint

The clock constraint appears in the SmartTime Constraints Editor (as shown in the figure below).

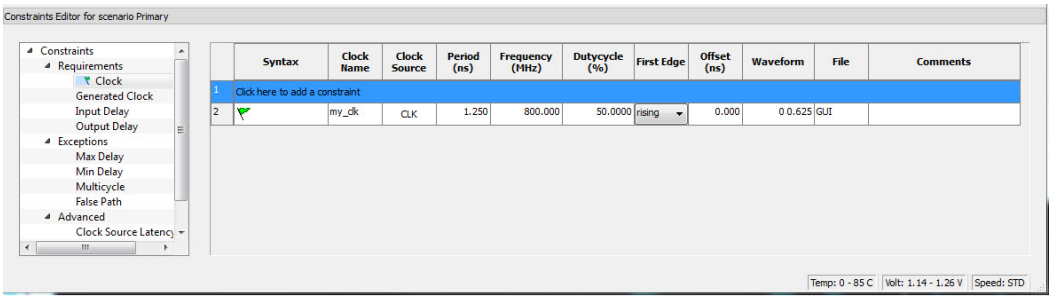


Figure 12 · 800 MHz Clock Constraint in the Constraint Editor

6. From the **File** menu, choose **Save** to save the constraints.
7. From the SmartTime **File** menu, choose **Exit** to exit SmartTime.

Run Place and Route

1. Right-click **Place and Route** and choose **Configure Options**.
2. Click the checkbox to enable Timing-Driven layout in Layout Options and leave the other values at the default settings (as shown in the figure below). Click **OK** to continue.

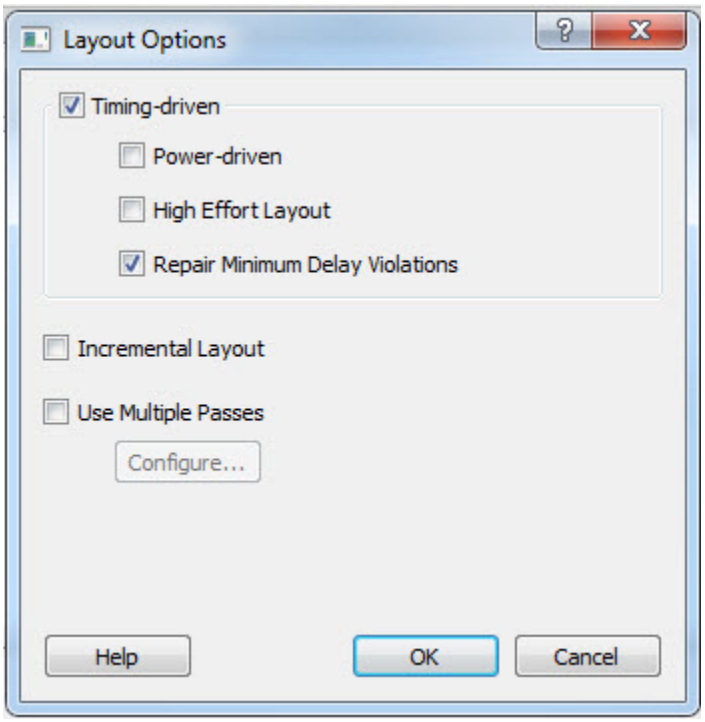


Figure 13 · Layout Options Dialog Box

3. Double-click **Place and Route** inside the Design Flow window to start the Place and Route.

A green check mark appears next to Place and Route after successful completion of Place and Route.

Maximum Delay Analysis with Timing Analyzer- 32-Bit Shift Register Example

The SmartTime Maximum Delay Analysis window displays the design maximum operating frequency and lists any setup violations.

To perform Maximum Delay Analysis:

1. Right-click **Verify Timing** in the Design Flow window and choose **Open Interactively** to open SmartTime. The Maximum Delay analysis window appears. A green check next to the clock name indicates there are no timing violations for that clock domain. The Summary page displays a summary of the clock domain timing performance.

   The Maximum Delay Analysis Summary displays:

- Maximum operating frequency for the design
- External setup and hold requirements
- Maximum and minimum clock-to-out times. In this example, the maximum clock frequency for CLK is 609.75 MHz.

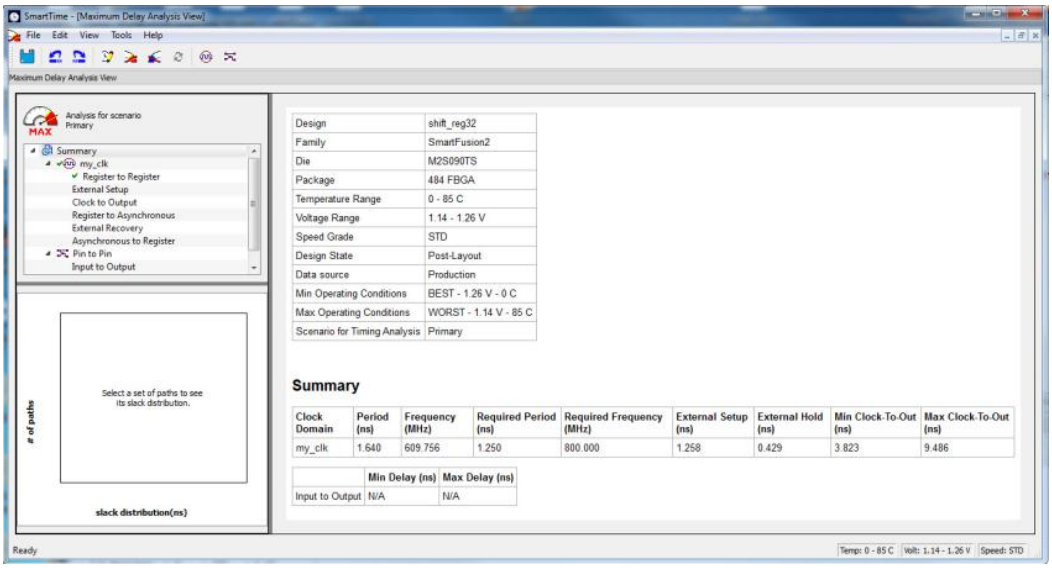


Figure 14 · Maximum Delay Analysis - Summary

2. Expand **my_clk** to display the Register to Register, External Setup and Clock to Output path sets.
3. Select **Register to Register** to display the register-to-register paths. The window displays a list of register-to-register paths and detailed timing analysis for the selected path (as shown in the figure below). Note that all the slack values are positive, indicating that there are no setup time violations

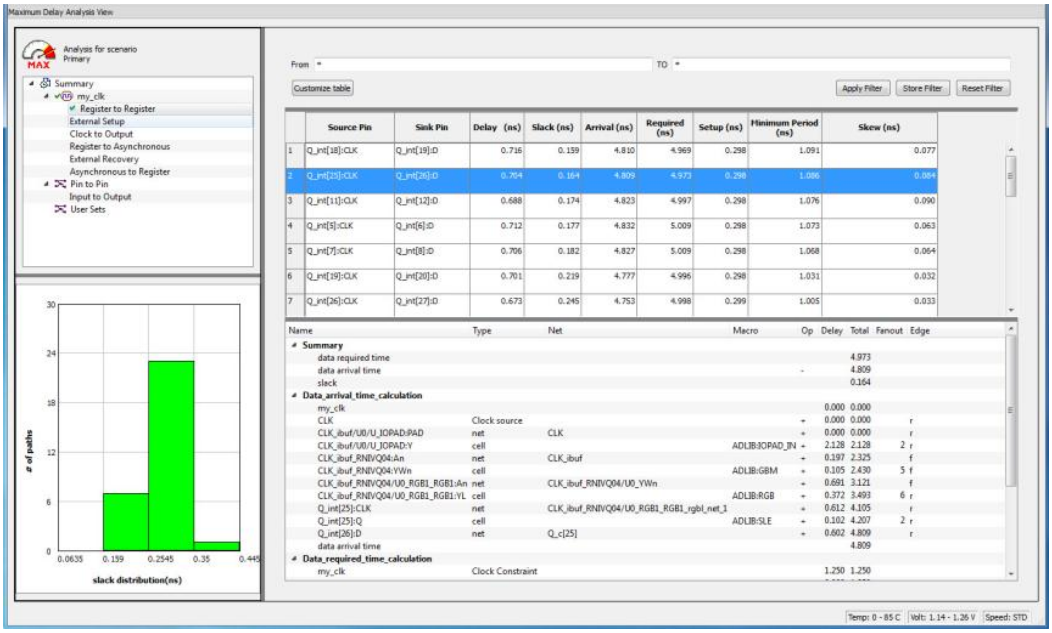


Figure 15 · SmartTime Register to Register Delay

4. Double-click a path row to open the Expanded Path window. The window shows a calculation of the data arrival and required times along with a schematic of the path (as shown in the figure below).

Note: The Timing Numbers in these reports may vary slightly with different versions of the Libero Software, and may not be exactly the same as what you will see when you run the tutorial.

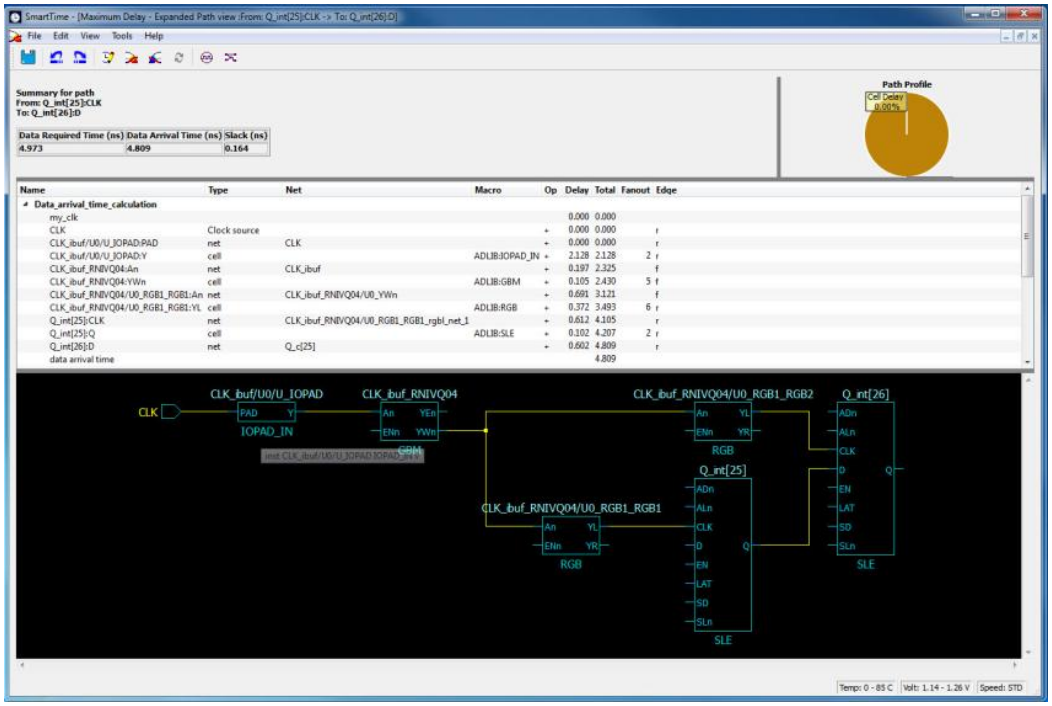


Figure 16 · Register-to-Register Expanded Path View

5. Select **External Setup** to display the Input to Register timing. Select **Path 3**. The Input Arrival time from the EN pin to Q_int[27]:EN is 4.547 ns (as shown in the figure below).

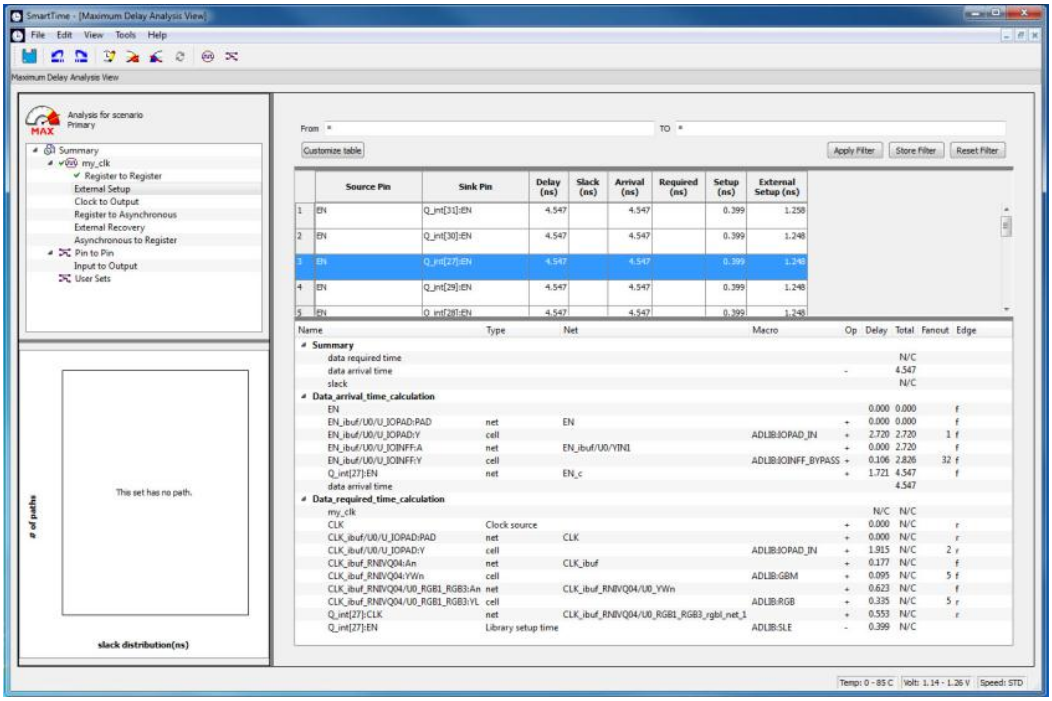


Figure 17 · SmartTime - Input to Register Path Analysis

6. Select **Clock to Output** to display the register to output timing. Select Path 1. The maximum clock to output time from Q_int[16]:CLK to Q[16] is 9.486ns .

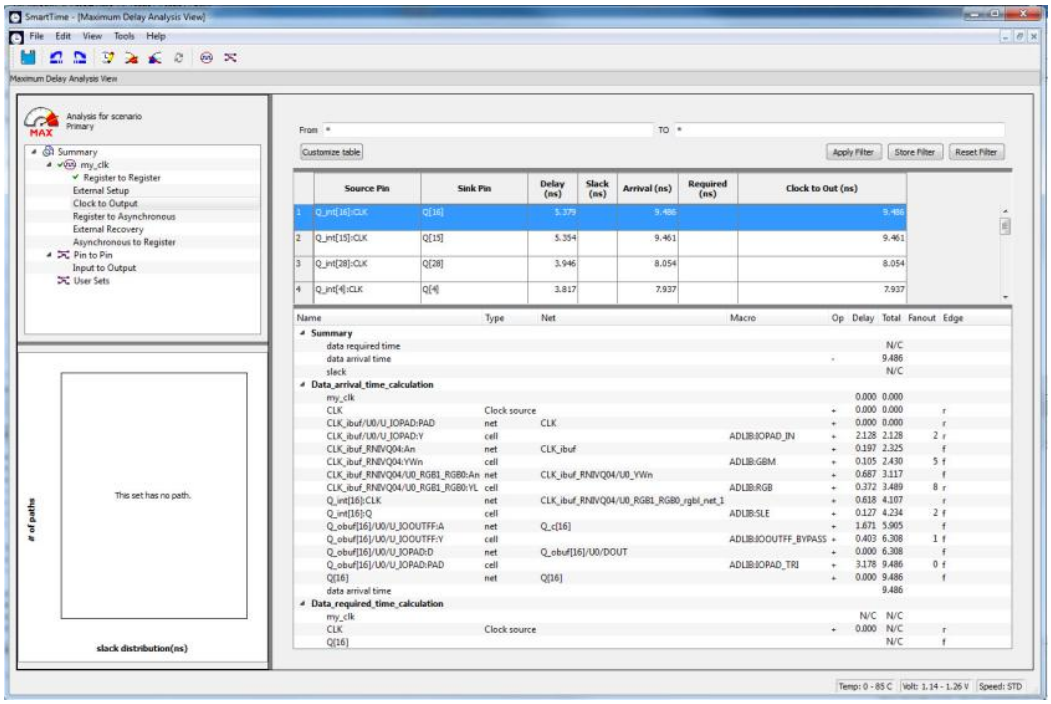


Figure 18 · SmartTime Clock to Output Path Analysis

Minimum Delay Analysis with Timing Analyzer - 32-Bit Shift Register Example

The SmartTime Minimum Delay Analysis window identifies any hold violations that exist in the design.

To perform Minimum Delay Analysis:

1. From the SmartTime Constraints Editor Tools menu, choose **Minimum Delay A**
2. **Analysis**. The Minimum Delay Analysis View appears, as shown in the figure below.

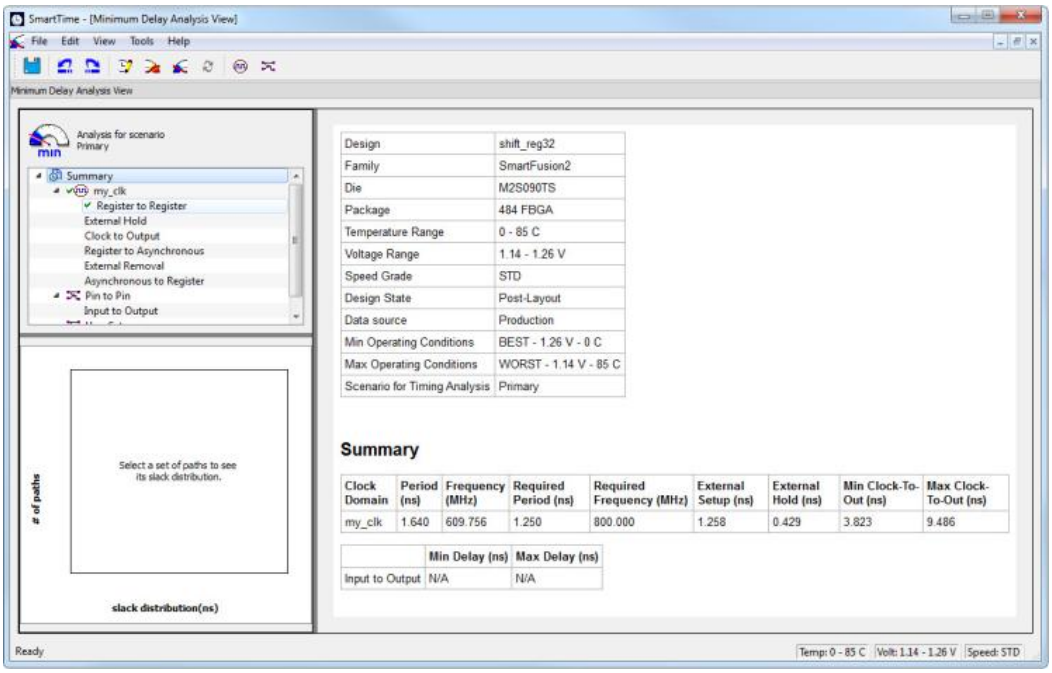


Figure 19 · SmartTime Minimum Delay Analysis View- Summary

3. Expand **my_clk** to display Register to Register, External Hold, Clock to Output, Register to Asynchronous, External Removal and Asynchronous to Register path sets.
4. Click **Register to Register** to display the reg to reg paths. The window displays a list of register to register paths and detailed timing analysis for the selected path. Note that all slack value are positive, indicating that there are no hold time violations.
5. Click to select the first path and observe the hold analysis calculation details, as shown in the figure below.

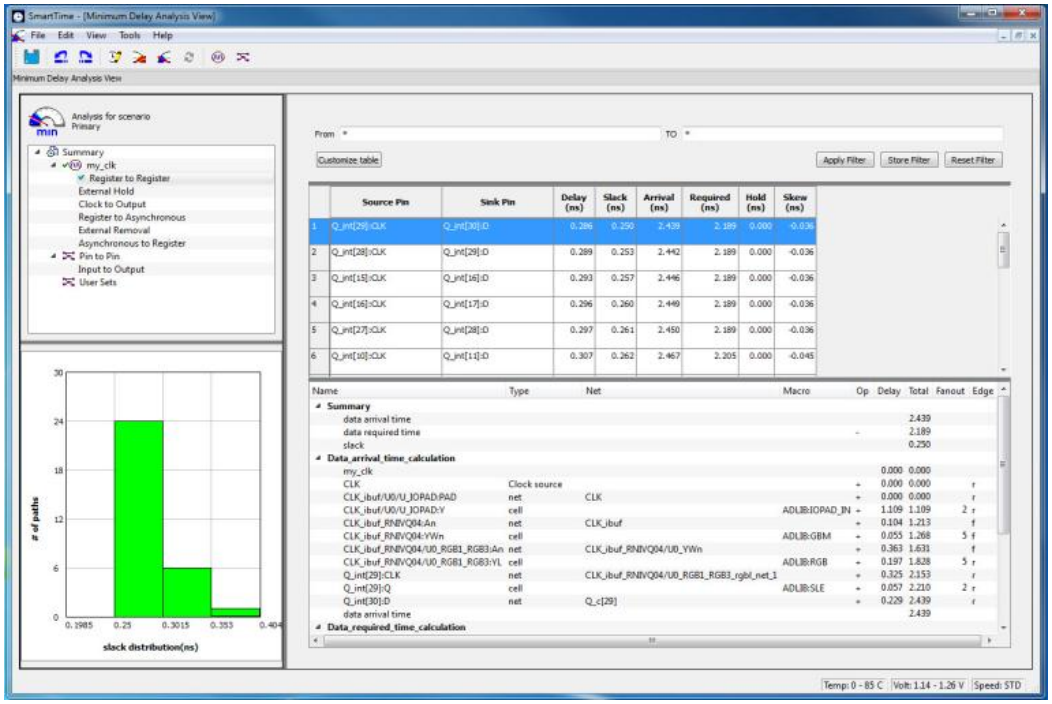


Figure 20 · SmartTime Minimum Delay Analysis

Changing Constraints and Observing Results - 32-Bit Shift Register Example

You can use SmartTime to adjust constraints and view the results in your design. To do so:

1. Open the SmartTime Constraints Editor (**Tools > Constraints Editor**).

   The Constraints Editor displays the clock constraint at 800 MHz that you entered earlier, as shown in the figure below.

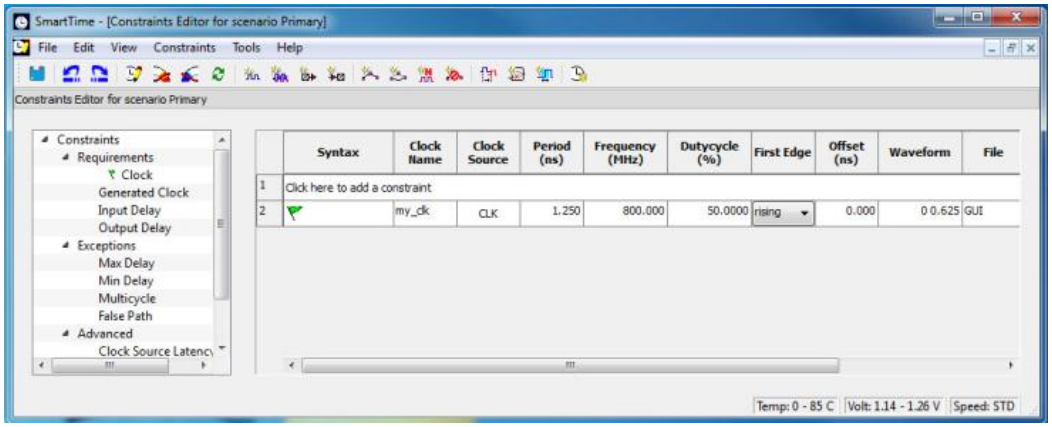


Figure 21 · Clock Constraint Set to 800 MHz

2. Select the second row. Right click on it and select Edit Clock Constraint. This will open the Edit Clock Constraint dialog box. Change the clock constraint from 800 MHz to 1000 MHz and click green check mark to continue.
3. From the **View** menu choose Recalculate All to recalculate the delays using your new clock constraint.
4. From the **Tools** menu choose **Maximum Delay Analysis View** to view the max delay analysis.
5. Expand **my_clk** in the Maximum Delay Analysis window. Click **Register to Register** to observe the timing information. Note that the slacks decrease after you increase the frequency and recalculate. You may see the slacks go negative indicating Timing Violations. Negative slacks are shown in red.

   Note: The actual timing numbers you see may be slightly different.

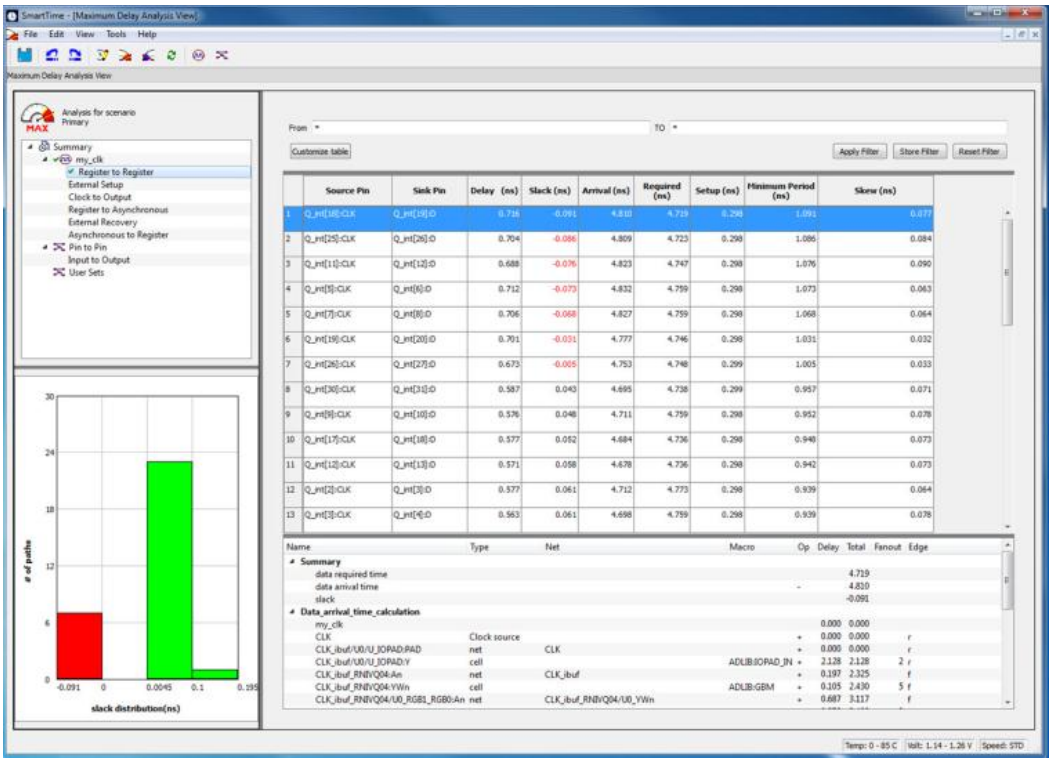


Figure 22 · Maximum Delay Analysis After Setting Clock Constraint to 1000 MHz

6. Close SmartTime. Click **No** when prompted to save changes.

Tutorial 2 - Adding an *.sdc File to Constrain Clock

This tutorial uses the 4-bit count16 example to step you through the process of entering constraints and analysing the timing performance of the design. Two options of entering clock constraints are covered: using SmartTime's Constraint Editor and importing an *.sdc file into Libero SoC.

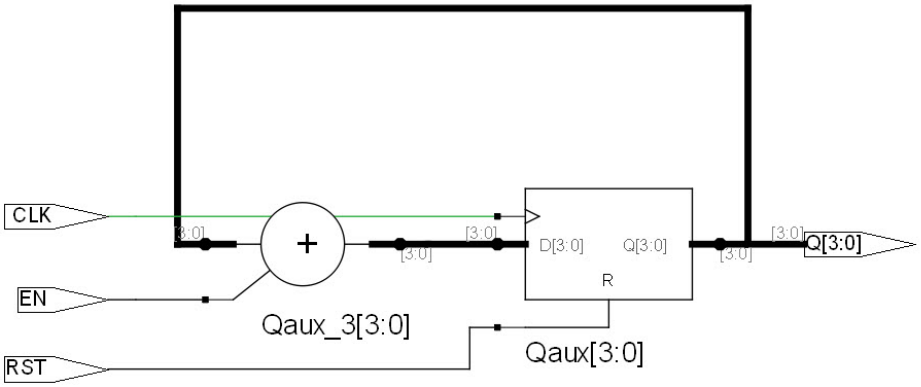


Figure 23 · Tutorial 2 – 4-bit Counter

1. From the **Project** menu, choose New Project to create a new Libero project.
2. Name the project **smarttime_tutorial** and set the project location according to your preferences. Enter the following values for the new project:

- **Family**: SmartFusion2
- **Die**: M2S050

- **Speed**: STD
- **Die Voltage**: 1.2 V
- **Package**: 484 FBGA
- **Range**: COM

7. Click Finish to create the new project.
8. Import the count16_behave.v into your project (**File > Import > HDL Source Files**) from the folder where you have downloaded the tutorial files (link to tutorial files).
9. Double-click **Compile** in the Design Flow window to run both Synthesis and Compile with default settings.

Refer to the Compile and Layout help topics for more information.

You are ready to create your clock constraints.

Creating a Clock Constraint

You may create a Clock Constraint in one of two ways:

- Option 1 - Add Clock Constraint in the Constraint Editor
- Option 2 - Import an *.sdc file that contains the Clock Constraint

Option 1 - Create a clock constraint in the Constraint Editor:

1. In the Design Flow window, expand Edit Constraints and right click on Timing Constraints > Open Interactively to start SmartTime and open the SmartTime Constraints Editor (as shown in the figure below).

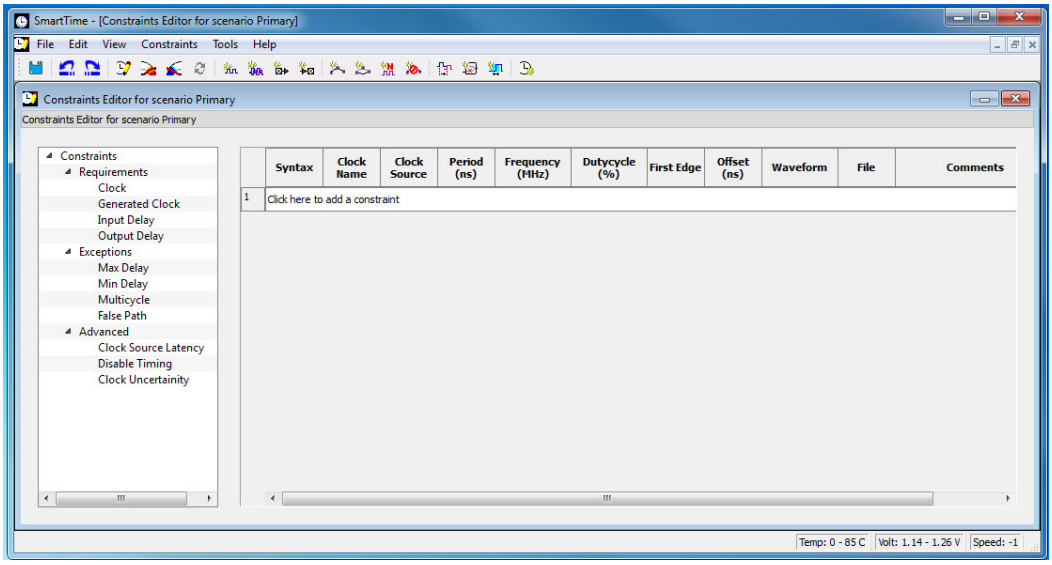


Figure 24 · SmartTime Constraints Editor

2. Add a clock constraint in one of the followng three ways:

- Click the New Clock Constraint icon  in the SmartTime toolbar.
- Double-click Clock under Requirements in the Constraints Pane.
- Right-click Clock and choose Add Clock Constraint

The Create Clock Constraint dialog box appears (as shown below).

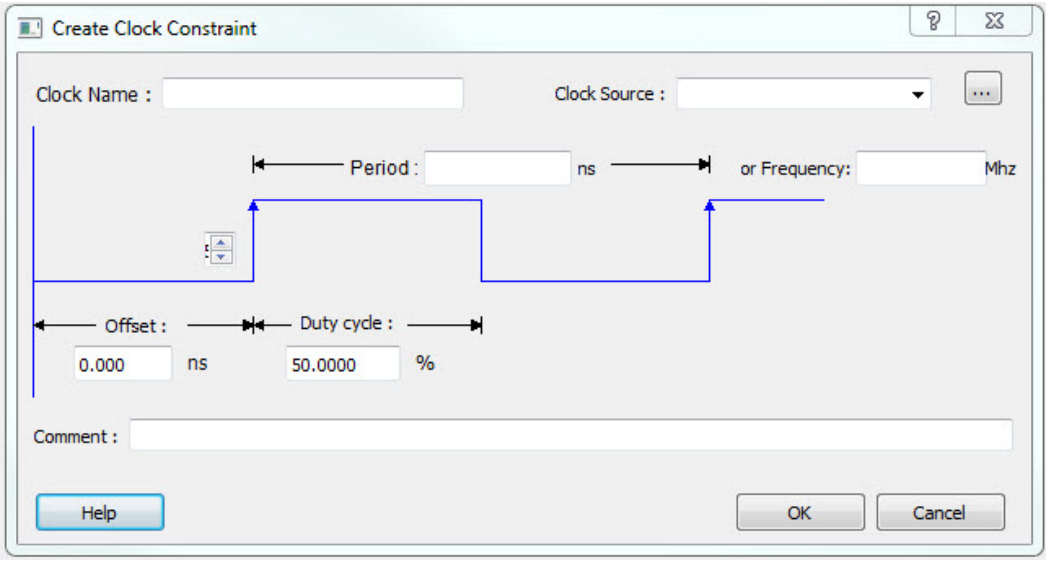
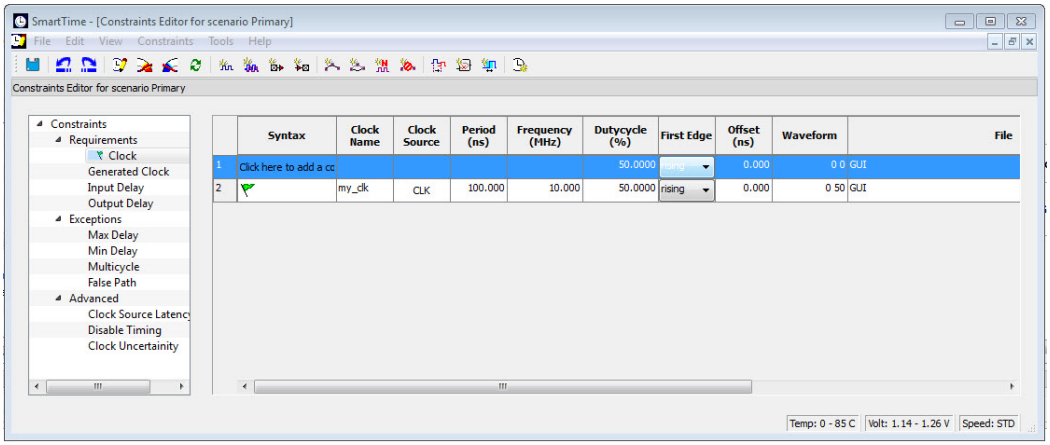


Figure 25 · Create Clock Constraint Dialog Box

3. Select the **CLK** pin from the pull-down menu in the **Clock Source** field, or click the **Browse** button to open the Select Source Pins for Clock Constraint dialog box, select the **CLK** pin and click **OK**.
4. Enter `my_clk` under **Clock Name**. The name of the first clock source is provided as default.
5. Type `100` in the **Period** field of the **Create Clock Constraint** box and accept all other default values.
6. Click **OK** to close the dialog box.

The clock constraint appears in the SmartTime Constraints Editor.



*Option 2 - Import a Timing Constraint *.sdc File*

The SDC file contains a Clock Constraint of 10.0 ns for the CLK.

1. From the **File** menu choose **Import > Timing Constraint (SDC) Files**.
2. Navigate to the folder that contains the file count16.sdc that you have downloaded. Click to select it and click **Open**.
3. A pop-up dialog appears to ask if you want to organize the constraint files for your current root (count16) for (Compile). Click **Yes** to continue.
4. In the Libero SoC Files window, check that the count16.sdc file appears in the constraint directory.

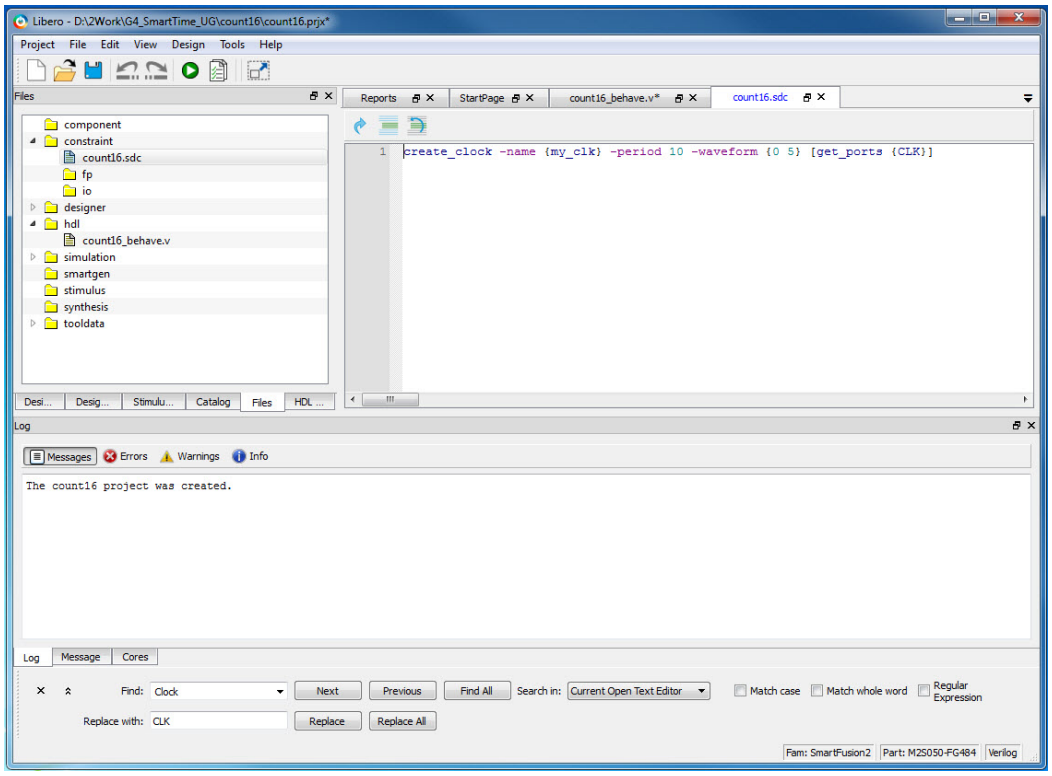


Figure 26 · count16.sdc under the Constraints folder in Files tab

Adding an Input Delay Constraint

Add an input delay constraint for Inputs EN and RST in one of the following three ways:

- Click the Add Input Delay Constraint icon  .
- Double-click **Input Dela**y under **Requirement** in the Constraint pane.
- In the Constraint pane, right-click **Input Delay** and choose **Add Input Delay Constraint**.

The Add Input Delay Constraint dialog box appears.

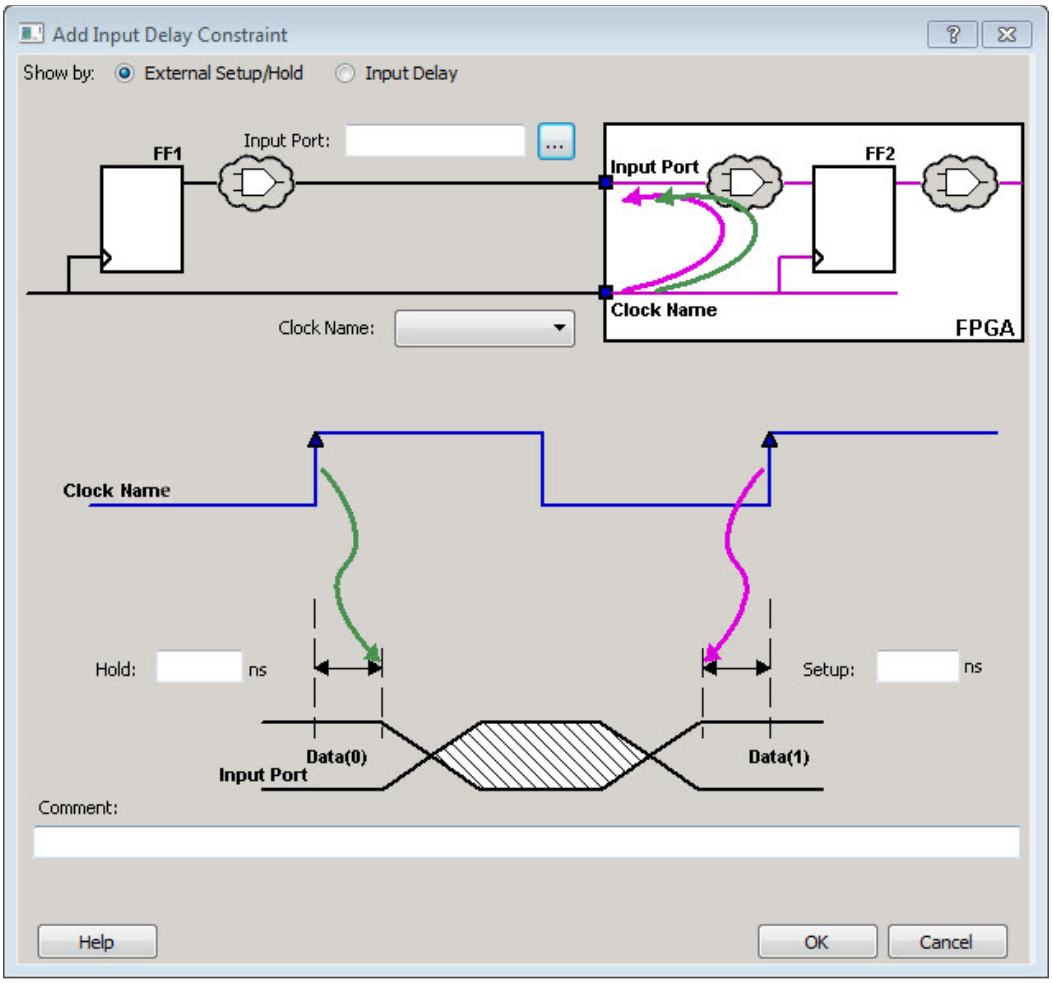


Figure 27 · Add Input Delay Constraint Dialog Box

1. In the **Show by** field, select **External Setup/Hold**.
2. Click the **Browse** button in the **Input Port** field to select the ports for the external setup constraints. The Select Ports for Input Ports dialog box appears and displays the input ports in the design.

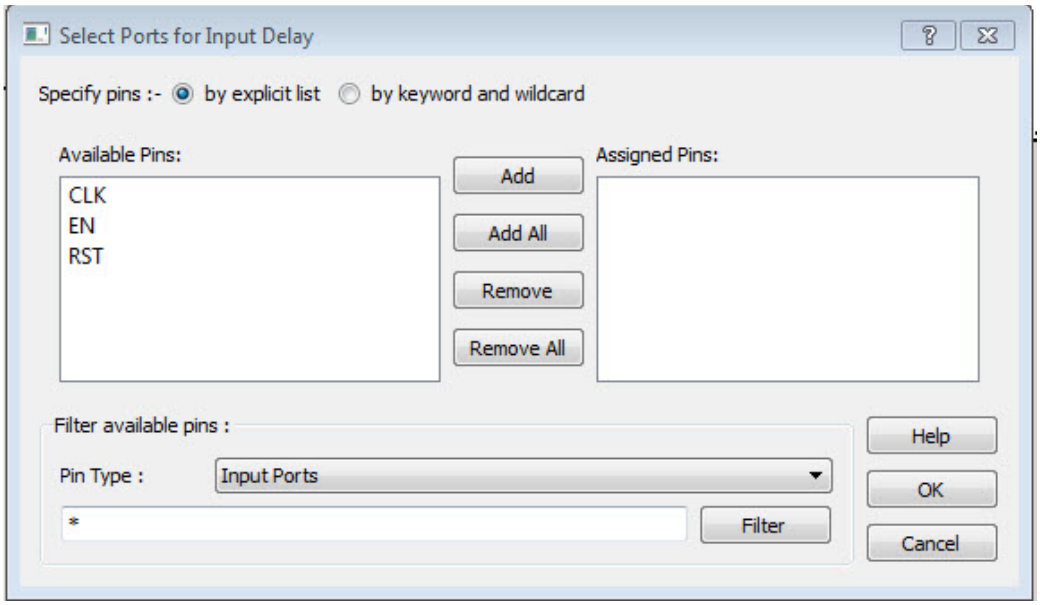


Figure 28 · Select Ports for Input Delay Dialog Box

3. Select the ports **EN** and **RST**, and then click **Add** to move the pins from the **Available Pins** list to the **Assigned Pins** list. Click **OK** to close the Select Ports for Input Delay dialog box.
4. Enter the following values in the Set Input Delay Constraint dialog box:
   - Clock Name: Select **my_clk** from the Clock Name drop-down list.
   - Hold Delay: **1 ns**
   - Setup Delay: **8 ns**
5. Add text in comment field if required for better readability.
6. Click **OK** to close the Set Input Delay Constraint dialog box.

The Input Delay constraints appear in the SmartTime Constraint Editor. Note that the Timing Constraints Editor View displays the external setup/hold requirement.

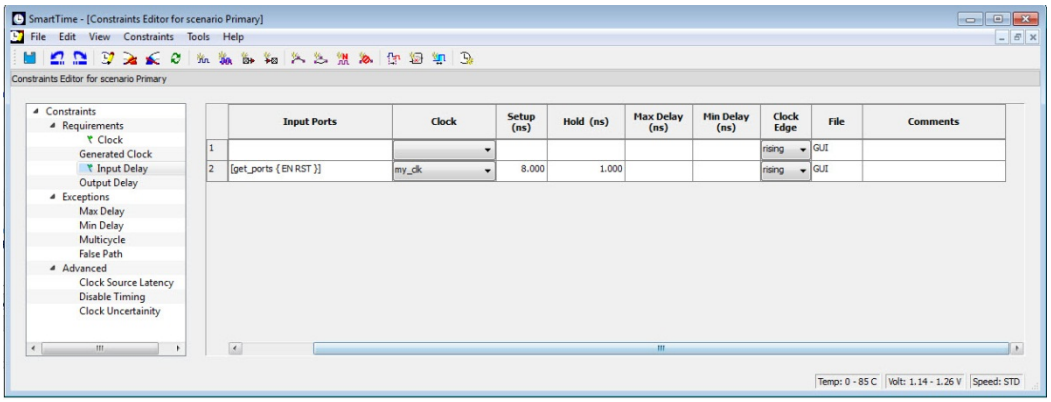


Figure 29 · SmartTime Constraints Editor with Input Delay Constraint

Continue to add an output delay constraint.

Adding an Output Delay Constraint

Add an output delay constraint in one of three ways:

- Click the Add Output Delay Constraint icon  in the SmartTime toolbar.

- Double-click Output Delay in the Requirement pane.
- Right-click Output Delay and choose Add Output Delay Constraint.

The Set Output Delay Constraint dialog box appears.

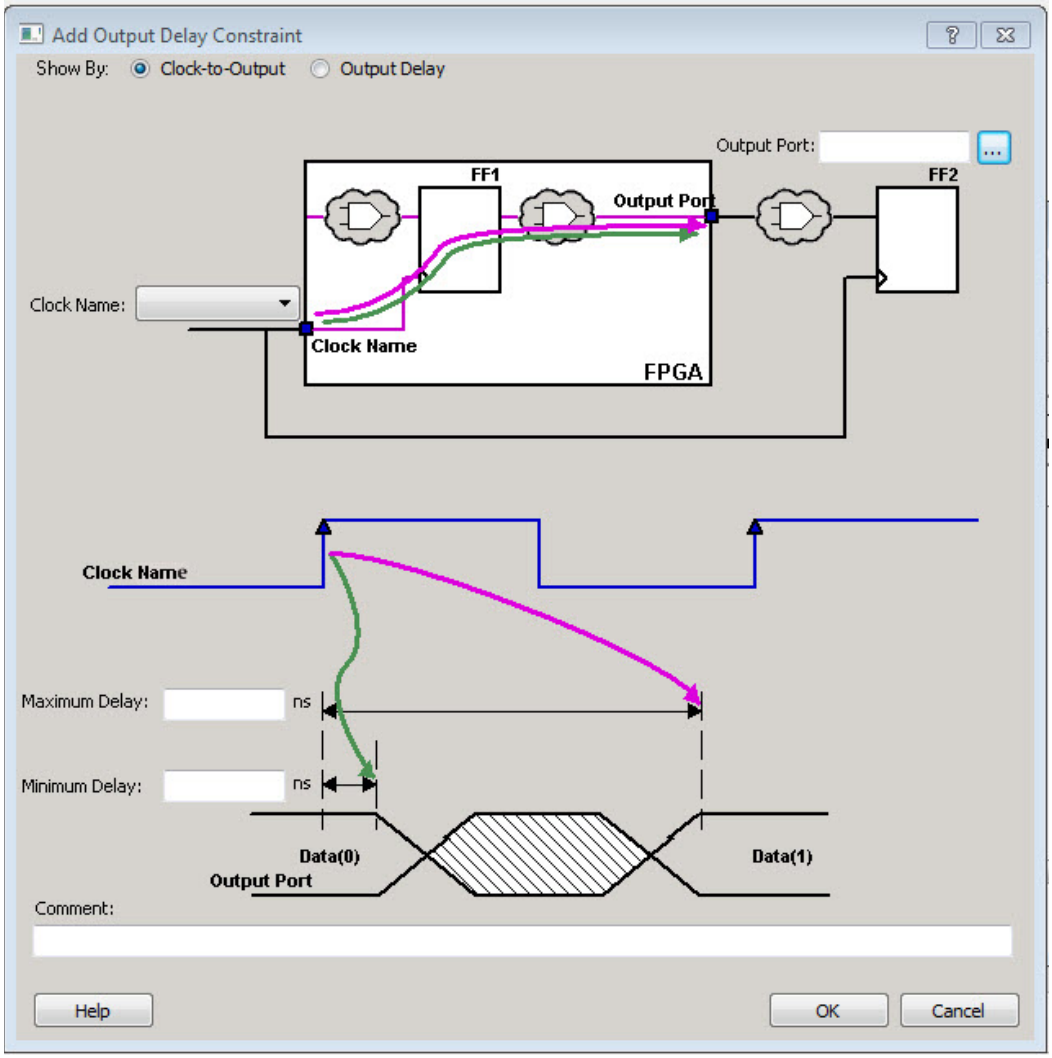


Figure 30 · Add Output Delay Constraint Dialog Box

1. In the **Show by** field, select **Clock-to-Output**.
2. Click the **Browse** button in the **Output Port** field to select the ports for the output delay constraint. The Select Ports for Output Delay dialog box appears and displays the output ports in the design.

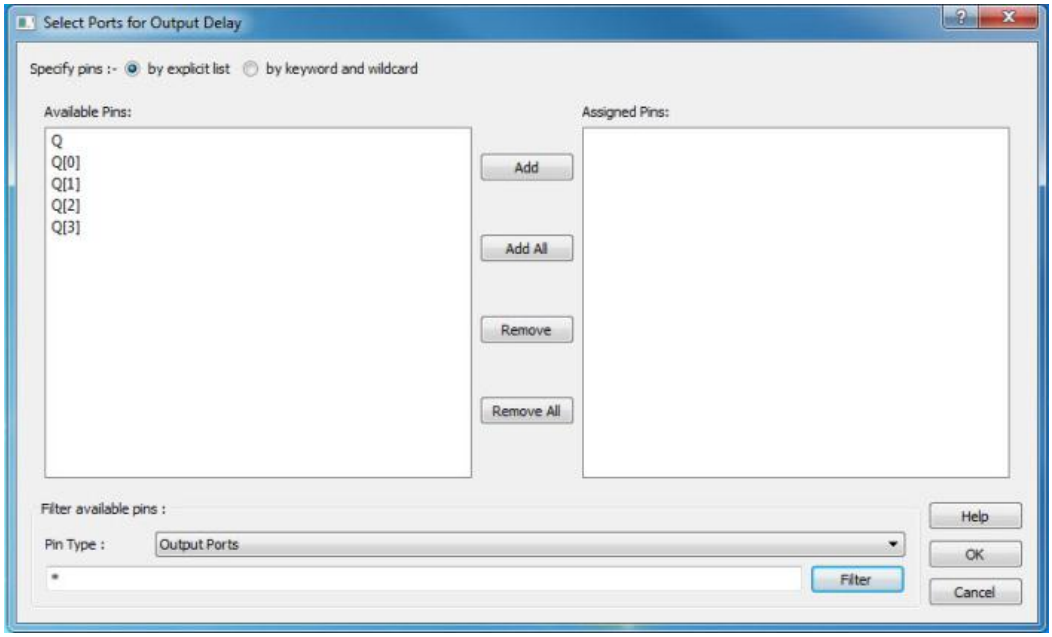


Figure 31 · Select Ports for Output Delay Dialog Box

3. Click **Add All** to select all the output ports. SmartTime moves the output pins from the **Available Pins** list to the **Assigned Pins** list.
4. Click **OK** to close the Select Ports for Output Delay dialog box.
5. Click **OK** to close the Select Ports for Output Delay dialog box.
6. Select **my_clk** from the **Clock Name** drop-down list in the Add Output Delay Constraint dialog
7. Enter **10** in the **Maximum Delay** field and **8** in the **Minimum Delay** field.
8. Add comments if required (optional)
9. Click **OK** to close the Set Output Delay Constraint dialog box. After the dialog box closes, the clk-to-out delay constraints appear in the SmartTime Constraint Editor.

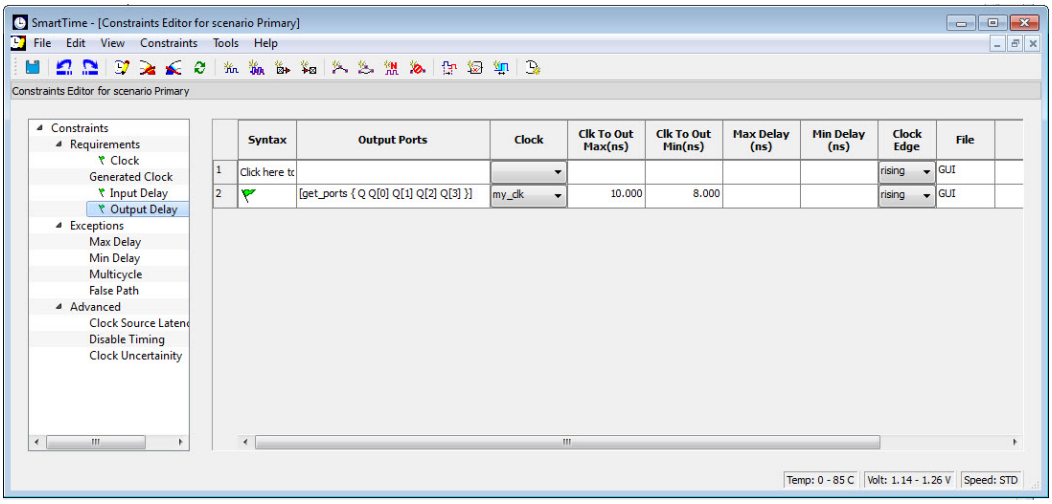


Figure 32 · SmartTime Constraints Editor with Output Delay Constraint

10. Click the Save icon  in the SmartTime toolbar to save your constraints.
11. Exit (**File -> Exit**) the SmartTime tool.

Place and Route Your Design

To run Place and Route:

1. Right-click **Place and Route** in the Design Flow window and choose **Configure Options**.
2. Click the checkbox to enable Timing-Driven layout in Layout Options and leave the other values at the default settings. Click **OK** to close the Layout Options dialog box.
3. Right-click Place and Route and choose Run.

A green check mark appears next to Place and route when it completes successfully.

You are now ready to analyze your design.

Analyzing the Maximum Operating Frequency

The Maximum Delay Analysis View indicates the maximum operating frequency for a design and displays any setup violations.

To perform the Maximum Delay Analysis:

Right-click Verify Timing and choose **Open Interactively** to open the Maximum Delay Analysis View.

Alternatively, you may click the Max Delay Analysis icon  to open the SmartTime Maximum Delay Analysis View.

A green flag next to the name of the clock indicates there are no timing violations for that clock domain (as shown below).

The SmartTime Maximum Delay Analysis View displays the maximum operating frequency for a design and any setup violations.

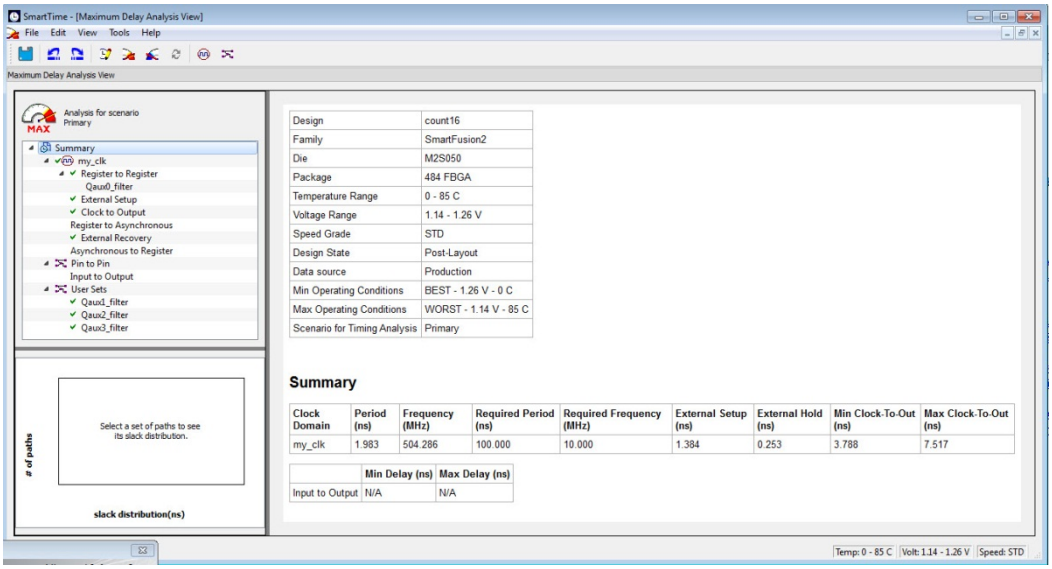


Figure 33 · SmartTime Maximum Delay Analysis View Summary

The Summary in the Maximum Delay Analysis View displays the maximum operating frequency for the design, the required frequency if any, the external setup and hold requirements, and the maximum and minimum clock-to-out times. In this example, the maximum clock frequency for CLK is 504 MHz.

You can now View Register-to-Register paths as part of the Maximum Delay Analysis.

See Also

Analyzing Your Design

Viewing Register-to-Register Paths

To view register to register paths:

1. Expand my_clk domain in the Domain Browser and display the Register to Register, External Setup, and Clock to output path sets.
2. Click **Register to Register** to display the register to register paths in the Paths List. It displays a list of register-to-register paths at the top of the Path List and detailed timing analysis for the selected path in the Path Details. Note that all the slack values are positive, indicating that no setup time violations exist.

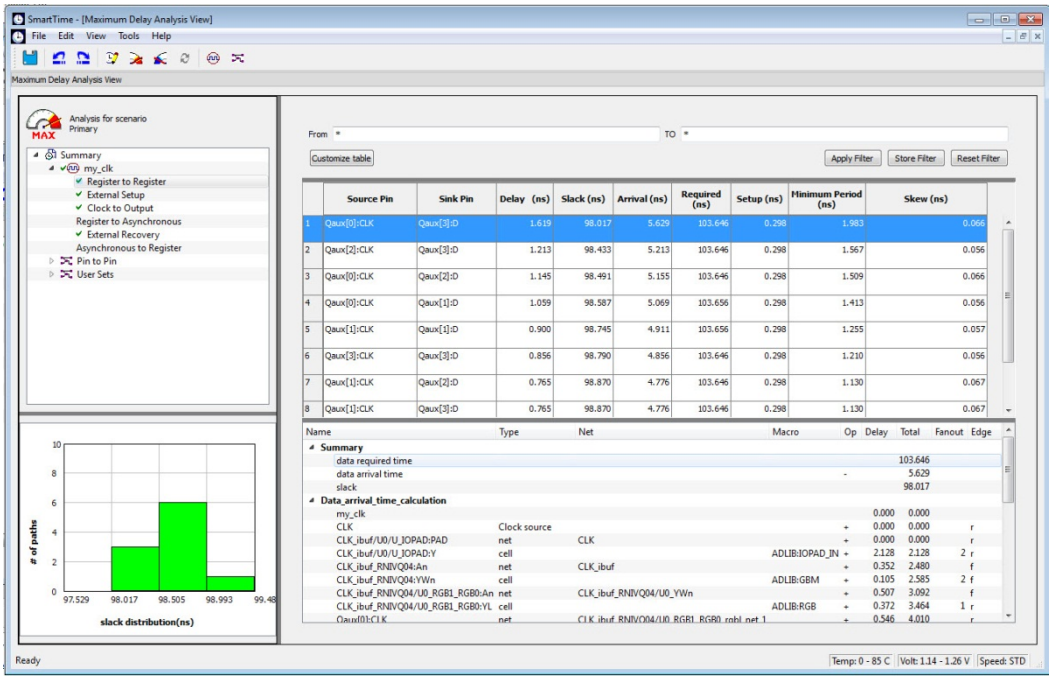


Figure 34 · SmartTime Register to Register Paths List

3. Double-click a path row to open the Expanded Path View (see figure below). The top of the view shows a calculation of the required and arrival times. A schematic of the path is shown at the bottom of the view.

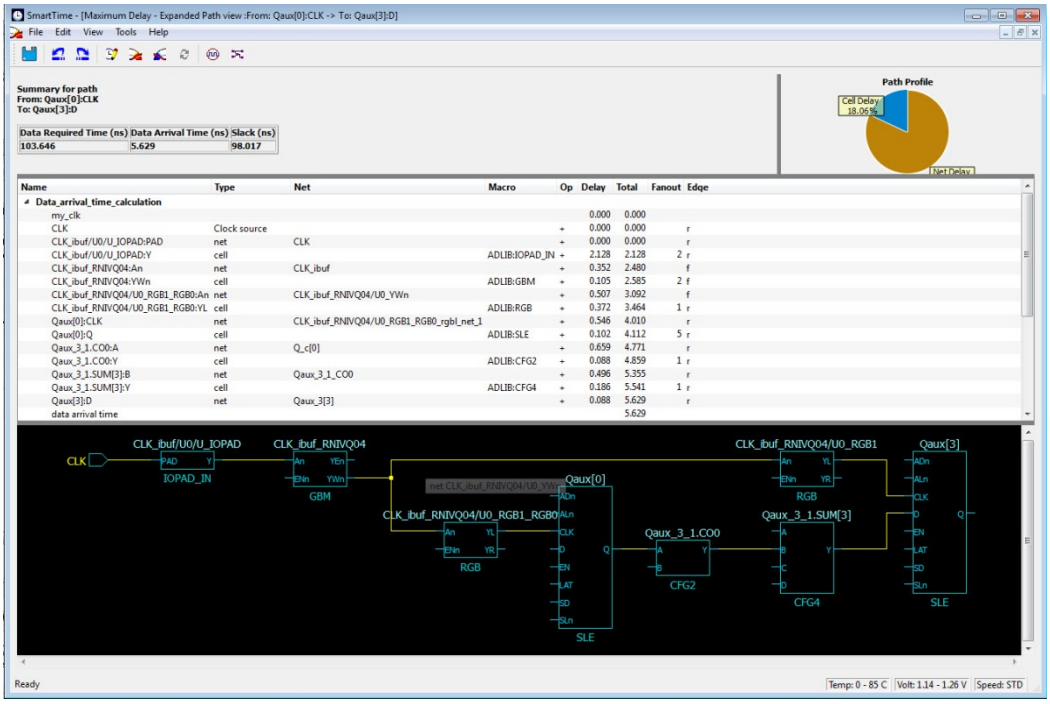


Figure 35 · SmartTime Expanded Paths View

Tip: Left-click and drag the mouse to zoom in or out in the schematic window.

4. Close the Expanded Paths View.

Viewing External Setup Paths

To view External Setup paths, click **External Setup** in the Domain Browser to display the external setup timing (as shown below). Note that the slack is positive in the tutorial example, indicating there are no timing violations.

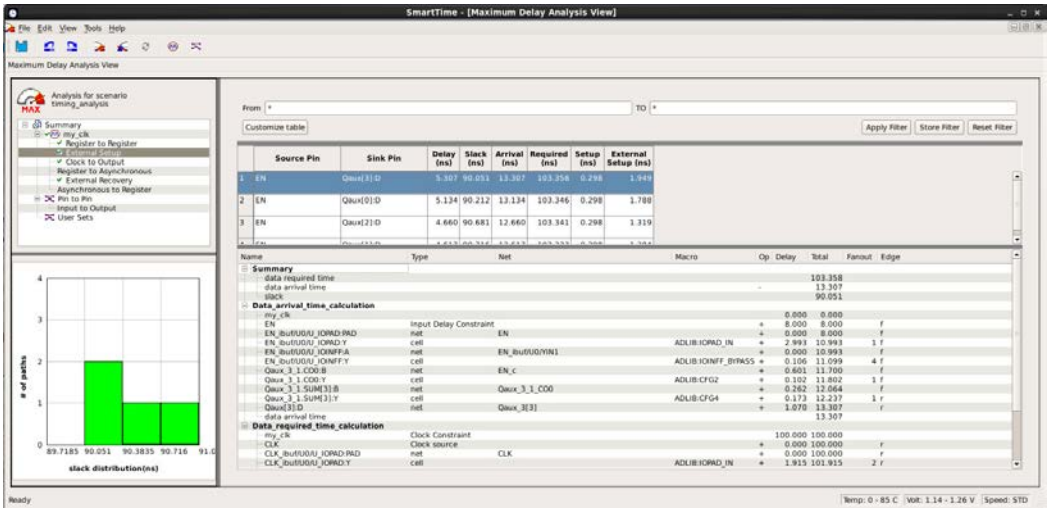


Figure 36 · SmartTime External Setup Path List

Viewing Clock-to-Output Paths

To view Clock-to-output paths, click **Clock to Output** in the Domain Browser to display the register to output timing. Again, the slack is positive in the tutorial example, indicating there are no timing violations.

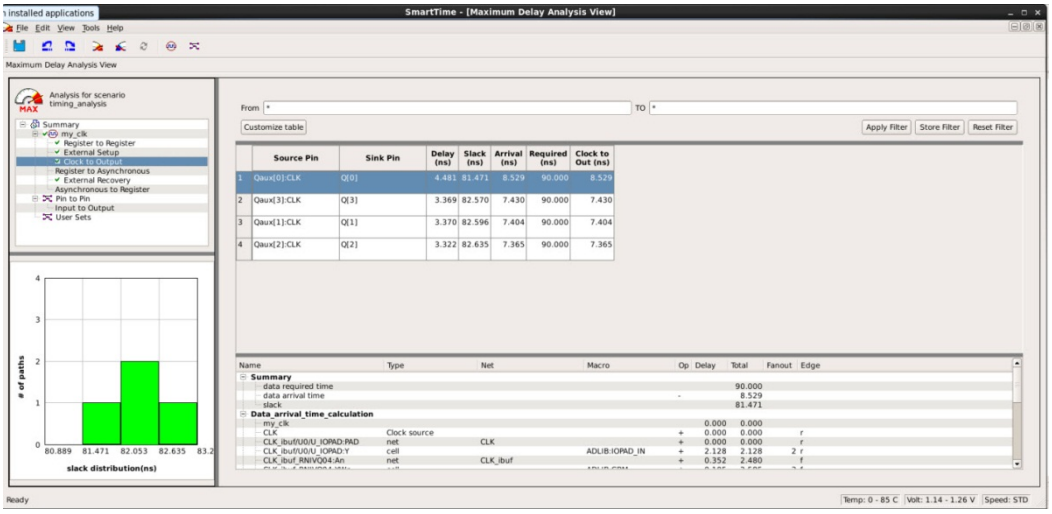


Figure 37 · SmartTime Clock to Output Paths List

Using Filters and Creating Analysis Sets

Filters can be used and saved to display analysis sets in the Maximum Delay Analysis window and the Minimum Delay Analysis window.

To create a filter:

1. When the Place and Route step is complete, click Verify Timing > Open Interactively.
2. In the Maximum Delay Analysis View, select the Register to Register path. Enter the following in the Filter fields (as shown in the figure below), then click **Apply Filter**:

   From: Qaux[0]:CLK
   To: *:D

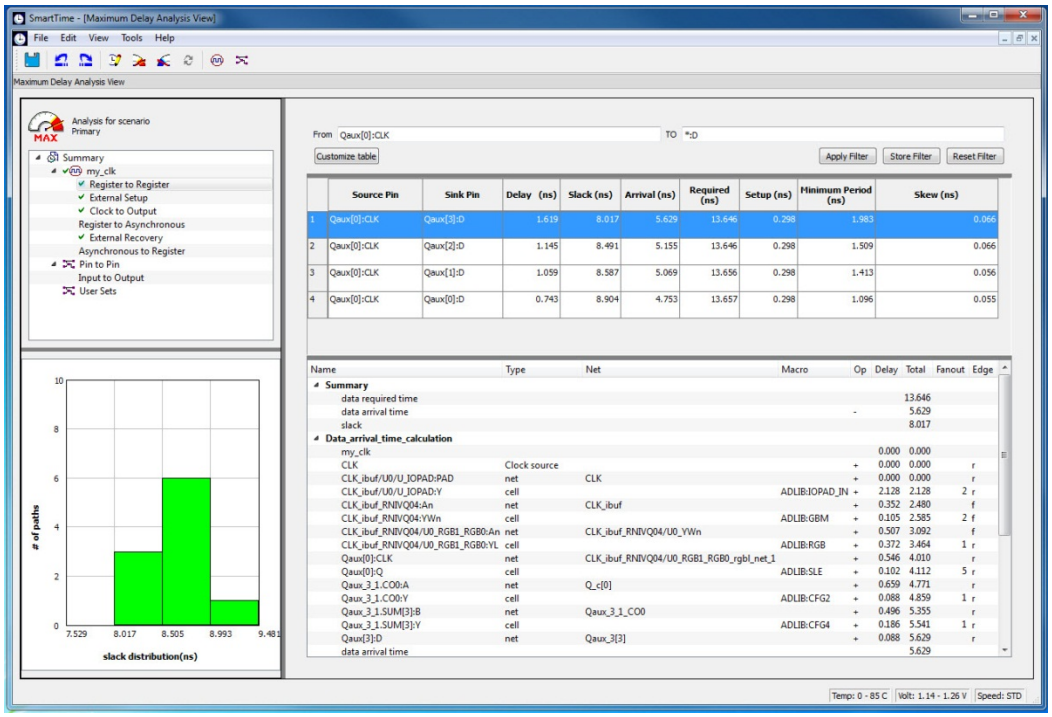


Figure 38 · Applying a Filter in the Maximum Delay Analysis View

3. Click **Store Filter** to save the filter. Enter **Qaux0_filter** in the Name field of the Create Filter Set dialog box. The set will be visible in the Maximum Delay Analysis View under Register to Register, as shown in the figure below.

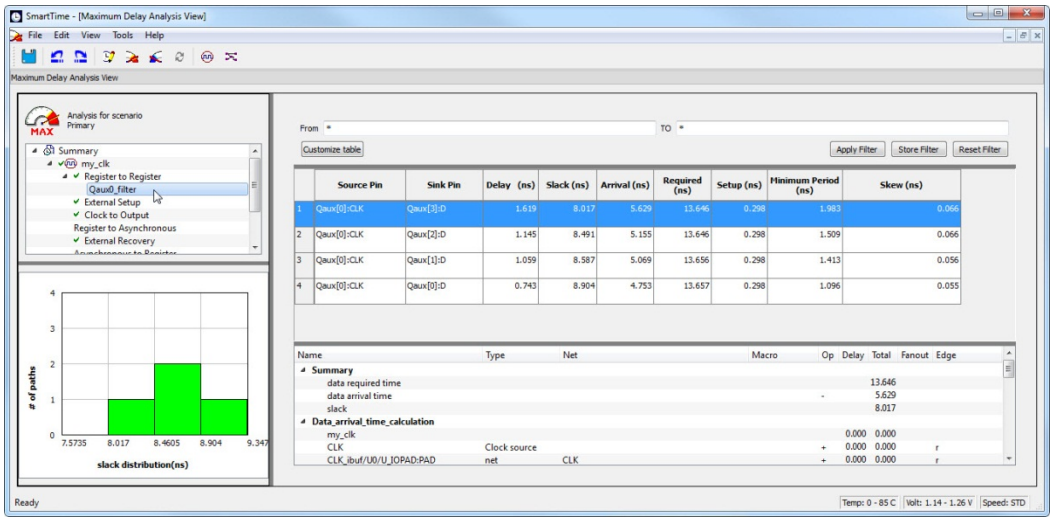


Figure 39 · Qaux0_filter Path Set

4. Click **Reset Filter**.
5. Right-click **Register to Register** and choose **Add Set** to open the Add Path Analysis Dialog box.

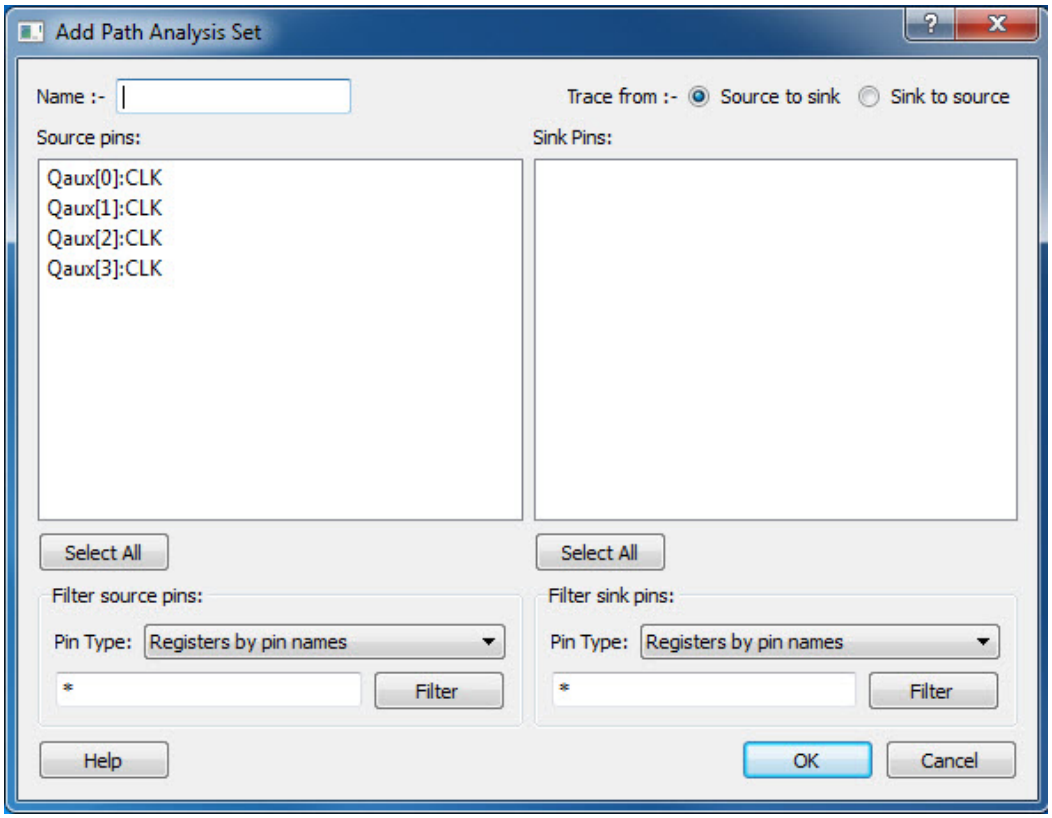
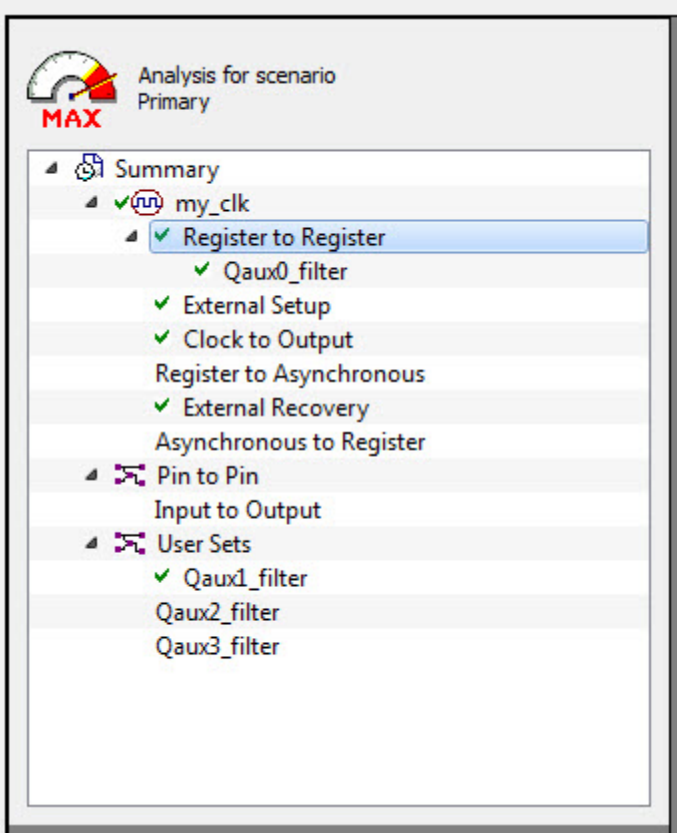


Figure 40 · Add Path Analysis Set Dialog Box

6. Use the values in the table below to add timing path sets.

From	To	Name
Qaux[1]:CLK	*:D	Qaux1_filter
Qaux[2]:CLK	*:D	Qaux2_filter
Qaux[3]:CLK	*:D	Qaux3_filter

The path sets appear under User Sets in the Maximum Delay Analysis View, as shown in the figure below.



7. Close SmartTime and Libero SoC.

Tutorial 3 - Design Using Both Clock Edges

This tutorial example analyzes SmartTime reports that include both rising and falling edges of a clock in the same design. The design (see the figure below) consists of a 16-bit serial-in parallel-out (SIPO) shift register. The shift register tmp1 is clocked on the rising edge of the clock. The output register tmp2 is clocked on the falling edge of the clock.

You will import the RTL verilog file shiftreg16.v and enter a clock constraint of 100 MHz. After routing the design, you will analyze the timing to determine the maximum operating frequency and export a timing report.

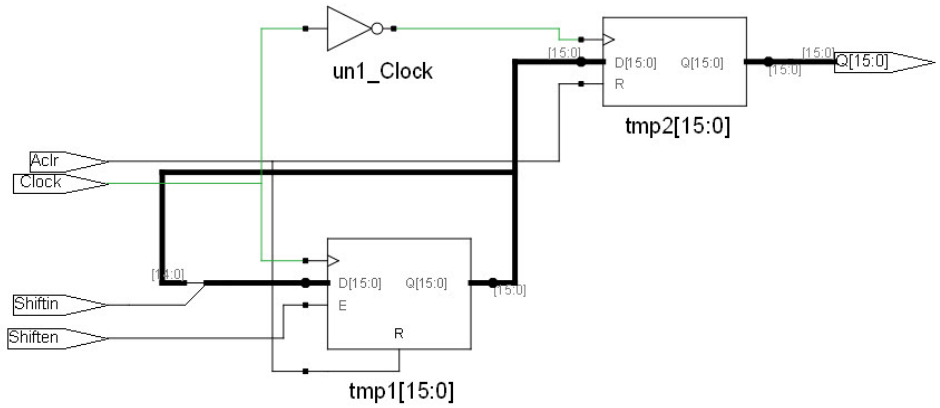


Figure 41 · Example Design that Uses Both Clock Edges

Set Up Your Example Design Project

1. Open Libero and create a new project (from the Project menu choose New Project).
2. Name the project shiftreg16 and set the project location according to your preferences. Enter the following values for your new project:

- **Family**: SmartFusion2
- **Die**: M2S090TS
- **Package**: FG 484
- **Speed**: STD
- **Die Voltage**: 1.2 V
- **Range**: COM

3. Click **Finish** to create the project.

Import the Verilog Source File - Design Uses Both Clock Edges

You must import the shiftreg16.v Verilog file into your design for this tutorial. Download the design files from the Microsemi website (URL to location of tutorial files).

To import the Verilog source file:

1. From the **File** menu, choose **Import > HDL Source Files**.
2. Browse to the location of the shiftreg16.v file you have downloaded and select it. Click **Open** to import the file.
3. Verify that the file appears in your project, as shown in the figure below.

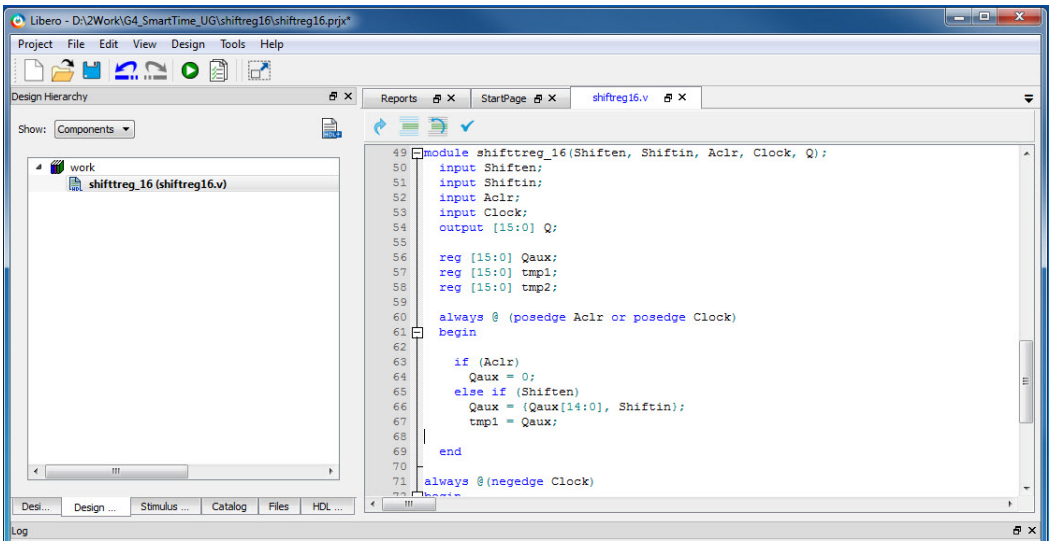


Figure 42 · shiftereg.v in the Design Hierarchy

Add a Clock Constraint - Design Uses Both Clock Edges

To add a clock constraint to your example design:

1. In the Design Flow window, expand **Edit Constraint** and choose **Timing Constraints > Open Interactively**.

   This triggers the synthesis and compile steps to run.
   The SmartTime Constraints Editor opens when the compile step is completed.
2. In the Constraints pane, right-click **Requirements > Clock** and choose **Add clock constraint**.
3. Enter a constraint with Clock Source (Clock), Clock Name (my_clk) of 100 MHz at a 50% duty cycle.

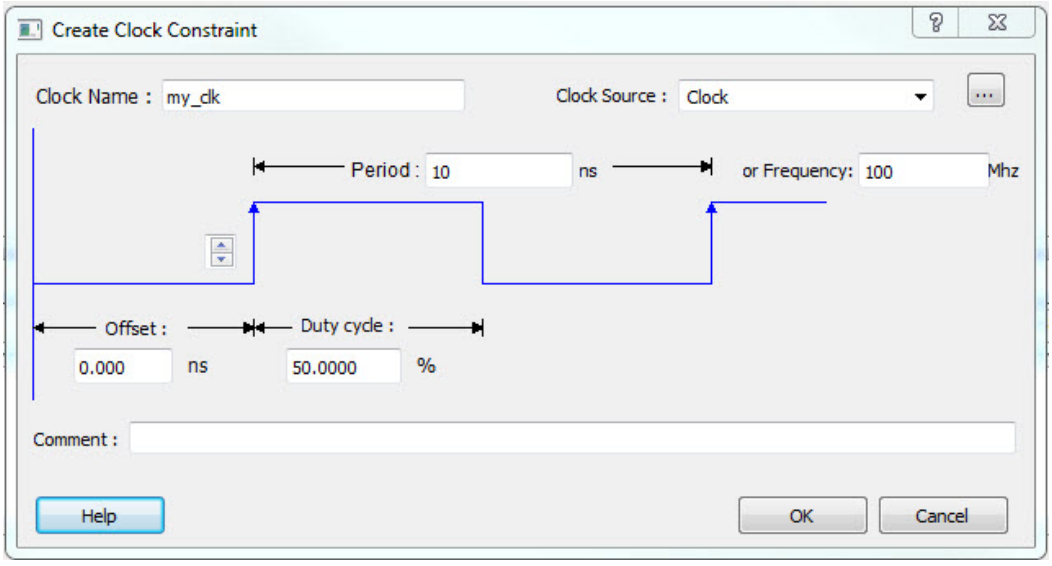
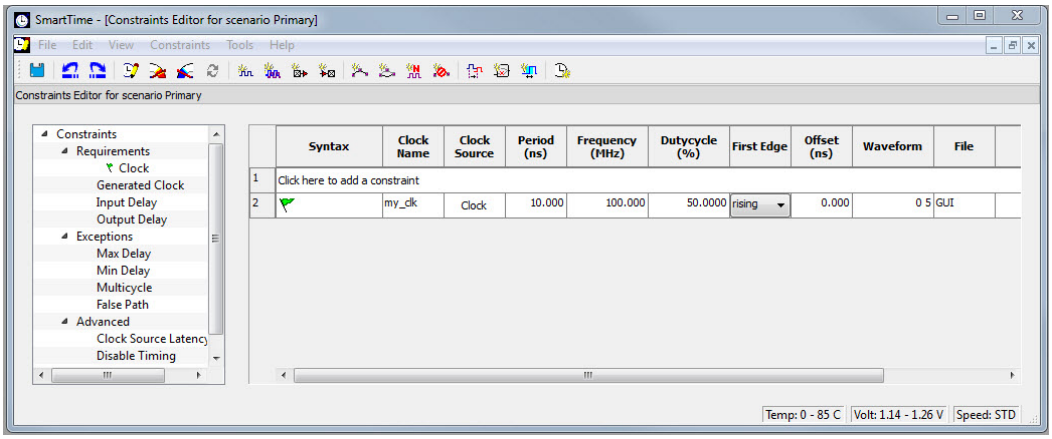


Figure 43 · Add 100 MHz Clock Constraint

The new constraint appears in the Constraints Editor, as shown in the figure below.



100 MHz Clock Constraint in the Constraint Editor

4. Save the constraints (**File > Save**).
5. Exit SmartTime Constraints Editor (**File > Exit**).

Run Place and Route for a Design that Uses Both Clock Edges

To run Place and Route on the design 'shiftreg_16':

1. In the Design Flow window, click **Place and Route** and choose **Configure Options**.
2. Click the checkbox to enable Timing-Driven layout in Layout Options and leave the other values at the default settings. Click **OK** to exit the Layout Options Dialog box.
3. Right-click **Place and Route** and choose **Run**.

A green check mark appears next to Place and Route to indicate successful completion of Place and Route.

Maximum Delay Analysis - Design Using Both Clock Edges

The SmartTime Maximum Delay Analysis window displays the design's maximum operating frequency and lists any setup violations.

To perform Maximum Delay Analysis:

1. In the Design Flow window, click **Verify Timing > Open Interactively** to open SmartTime. The Maximum Delay Analysis View window appears (as shown in the figure below). A green check next to the clock name indicates there are no timing violations for that clock domain. The Summary page displays a summary of the clock domain timing performance.

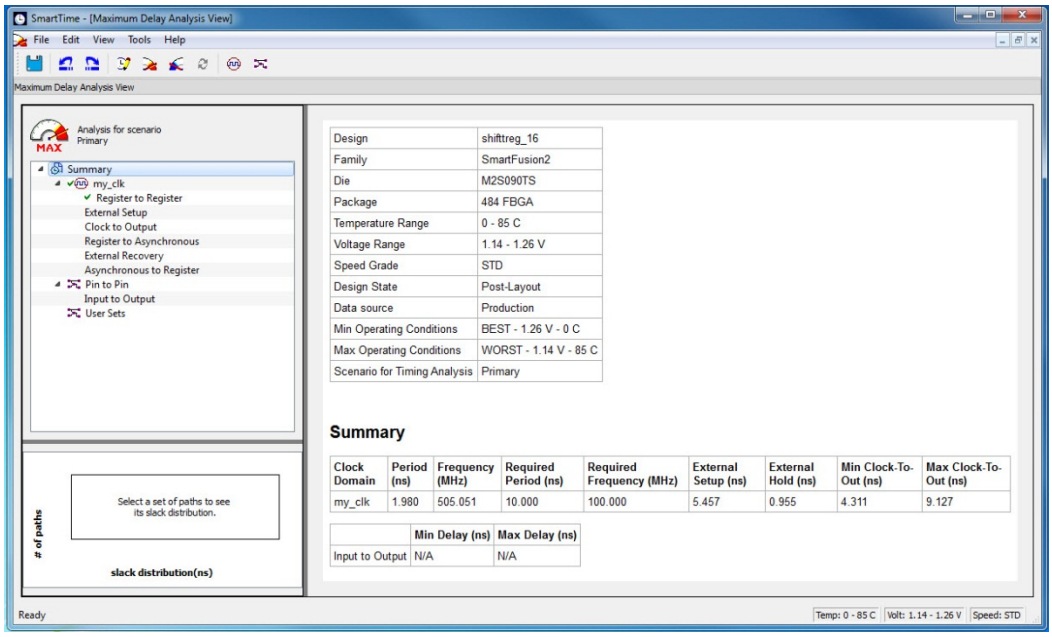


Figure 44 · Maximum Delay Analysis for Design Shifter

The Summary in the Maximum Delay Analysis View window indicates the maximum operating frequency for this design is 505.05 MHz.

2. Expand **my_clk** to display the Register to Register, External Setup and Clock to Output path sets.
3. Select **Register to Register** to display the register-to-register paths. The window displays a list of register-to-register paths and detailed timing analysis for the selected path (as shown in the figure below). Note that all the slack values are positive, indicating that there are no setup time violations.
4. Click to select row 1 and study the timing analysis (resize the Maximum Delay Analysis View window as required). The path is from register tmp1 to register tmp2. Note that SmartTime uses 5 ns in the data required calculation (circled in red in the figure below). This is because the source flip flop uses the rising edge of the clock and the destination flip-flop uses the falling edge of the clock.

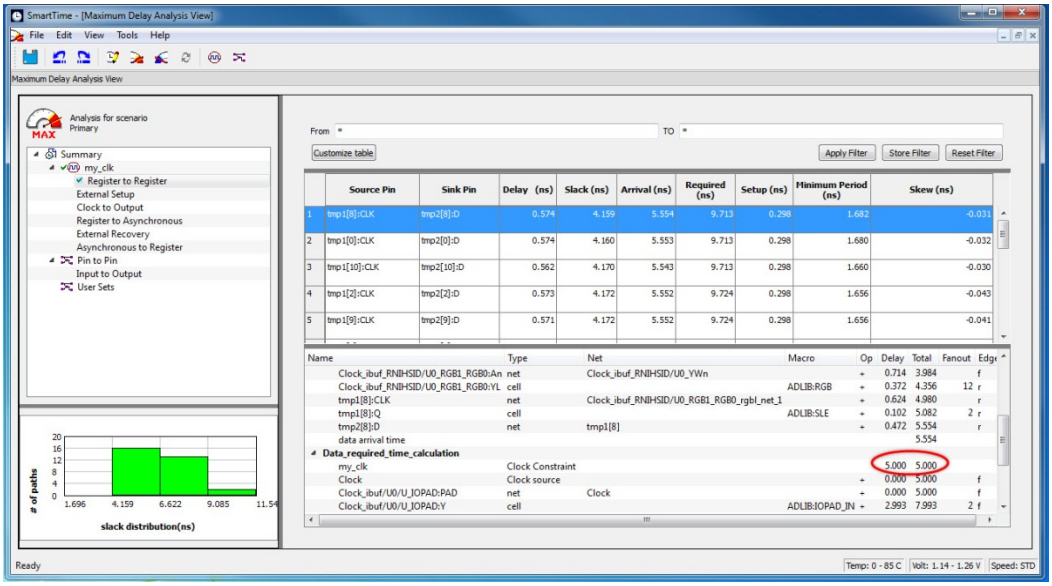


Figure 45 · Slack Calculation in Maximum Delay Analysis View

Generate a Timing Report - Design Uses Both Clock Edges

Timing reports can be generated from SmartTime. Timing reports enable you to quickly determine if there are any timing problems. The timing report lists the following information:

- Design information including device, speed grade and operating conditions.
- Design performance summary (maximum frequency, external setup and hold, minimum and maximum clock-to-out)
- Clock domain details
- Inter clock domain details
- Pin to pin timing

The timing report can be printed and saved.

To generate a Timing Report:

1. From the **Maximum Delay Analysis View** menu, choose **Tools > Reports > Timer** to open the Timing Report Options dialog box, as shown in the figure below.

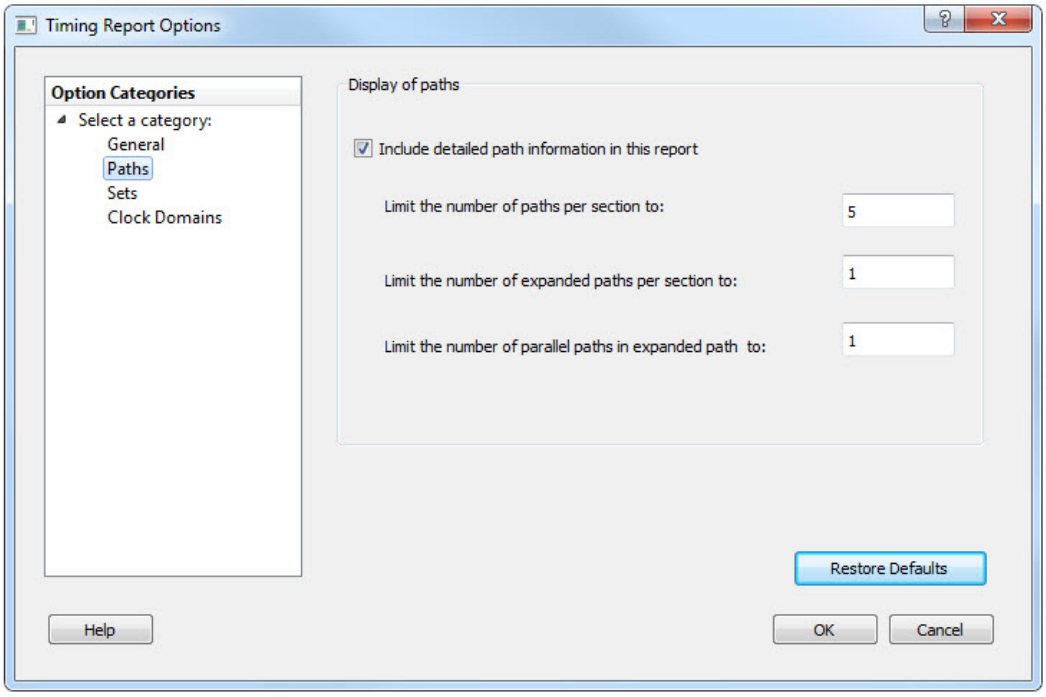


Figure 46 · Timing Report Options Dialog Box

2. Click the **Paths** category. Limit the number of reported paths to **5** (default), and click **OK**. The timing report opens in a new window, as shown in the figure below.

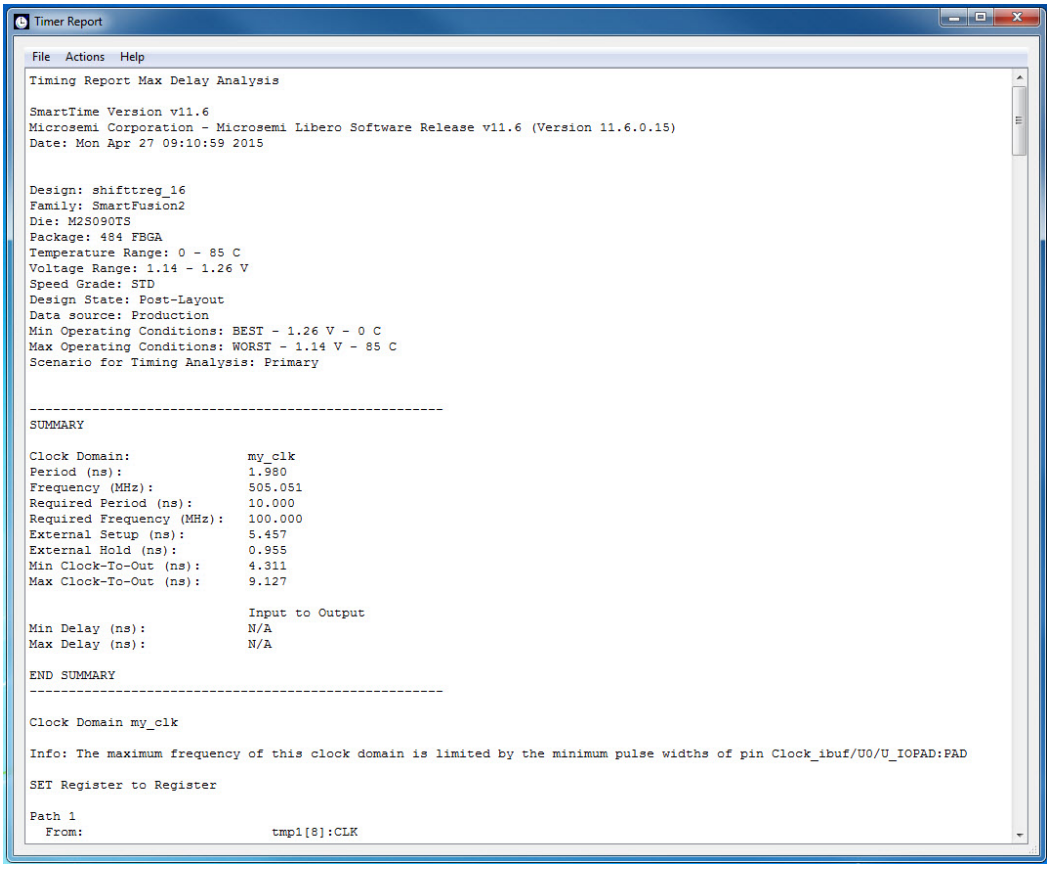


Figure 47 · Timing Report for shifter

The timing report contains the following sections:

- Header
- Summary
- Clock domain details for my_clk and expanded path information
- External setup information
- Clock to output delay information

6. Save the timing report (**File > Save As**) as **shifter_timing.rpt** and close the report window.
7. Close SmartTime and Libero.

Tutorial 4 - False Path Constraints

This section describes how to enter false path constraints in SmartTime. You will import an RTL source file from the design shown below. After routing the design, you will analyze the timing, set false path constraints, and observe the maximum operating frequency in the SmartTime Timing Analysis window.

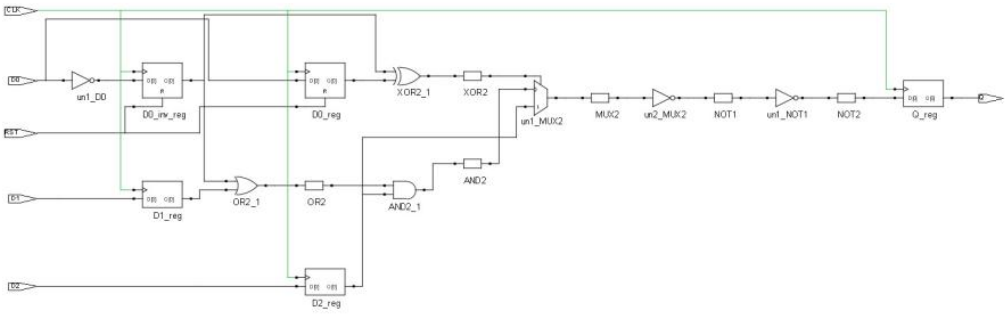


Figure 48 · Example Design with False Paths

Set Up Your False Path Example Design Project

1. Open Libero and create a new project (from the **Project** menu, choose **New Project**).
2. Name the project **false_path** and set the project location according to your preferences. Click **Next**. Enter the following values for your Device Selection settings:
   - **Family**: SmartFusion2
   - **Die**: M2S050
   - **Package**: 484 FBGA
   - **Speed**: STD
   - **Die Voltage**: 1.2 V
   - **Range**: COM
3. Click **Finish** to create the new project.
4. At the pop-up window, click **Use Enhanced Constraint Flow** in the New Project Information dialog box.

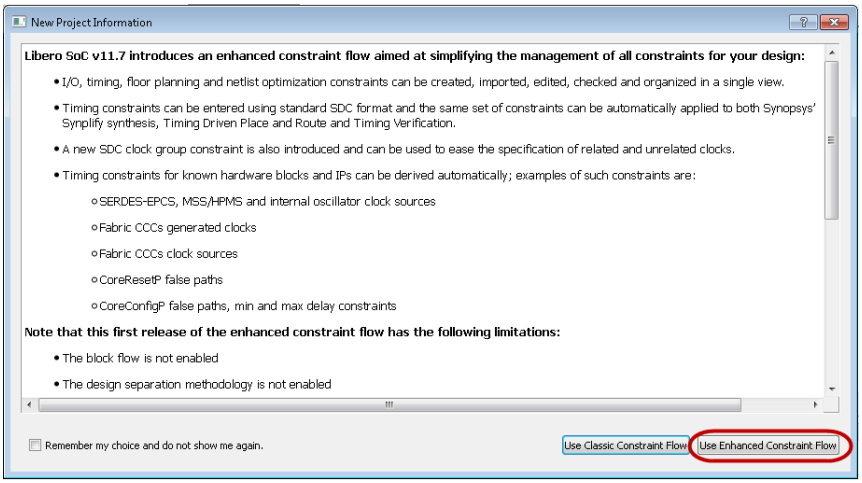


Figure 49 · New Project Information Dialog Box

Import the false_path Verilog File and Add Constraints

You must import the false_path.v Verilog source file into your design for this tutorial. Download the design files from the Microsemi website (URL link).

To import the Verilog Source File:

1. From the **File** menu, choose **Import > HDL Source Files**.

2. Browse to the location of the false_path.v where you have downloaded from the Microsemi website and select it. Click **Open** to import the file.
3. Verify that the file appears in your project, as shown in the figure below.

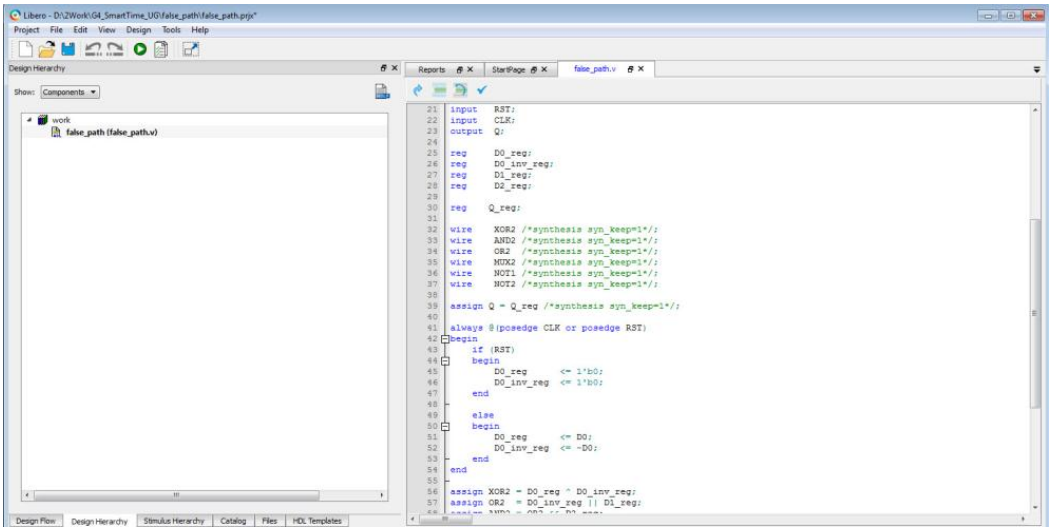


Figure 50 · false_path Design in Design Hierarchy

4. In the Design Flow window, double-click **Synthesize** to run synthesis. A green check mark appears when the Synthesis step completes successfully.
5. In the Design Flow window, double-click **Compile** to Compile with default settings. A green check mark appears next to Compile to indicate that it has completed successfully.
6. Expand **Edit Constraints**. Right-click **Timing Constraints** and choose **Open Interactively**.
7. In the SmartTime Constraints Editor, right click on **Requirements > Clock** and choose **Add clock constraint** under the Constraints pane on the left.
8. Enter a constraint for the clock source (CLK) with clock name (my_clk) of 100 MHz (50% duty cycle), as shown in the figure below.

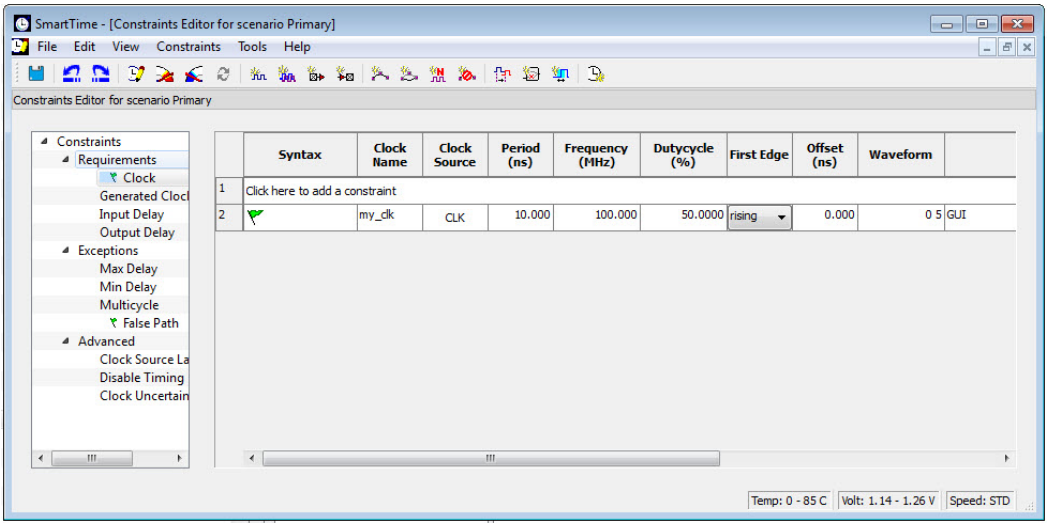


Figure 51 · Clock Constraint of 100 MHz in false_path design

9. Save your changes (**File > Save**) and close SmartTime (**File > Close**).

Place and Route Your FALSE_PATH Design

To run Place and Route on false_path design:

1. In Libero SoC, right-click **Place and Route** and choose **Configure Options**.

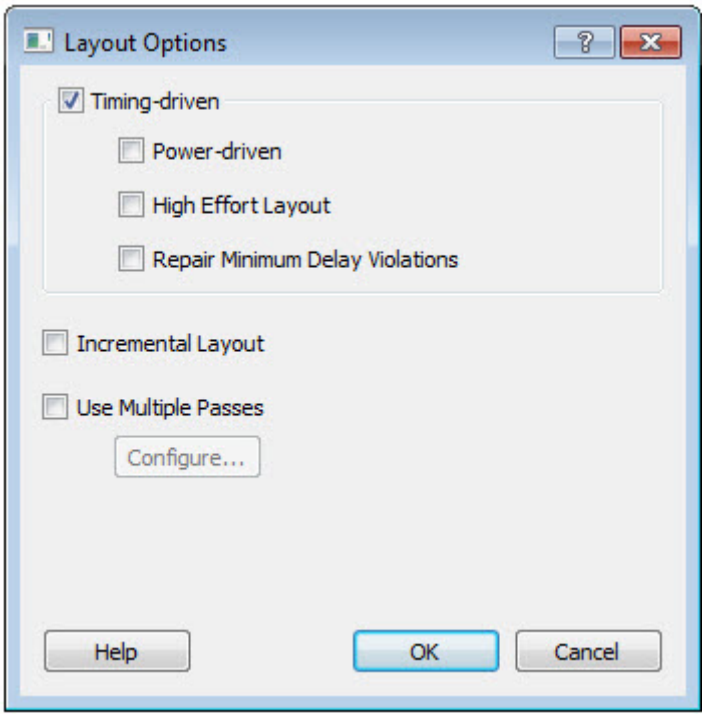


Figure 52 · Layout Options Dialog Box

2. Click the checkbox to enable Timing-Driven layout in Layout Options and leave the other values at the default settings. Click **OK** to close the Layout Options dialog box.
3. Right-click **Place and Route** and choose **Run**.

A green check mark appears next to Place and Route in the Design Flow window when Place and Route completes successfully.

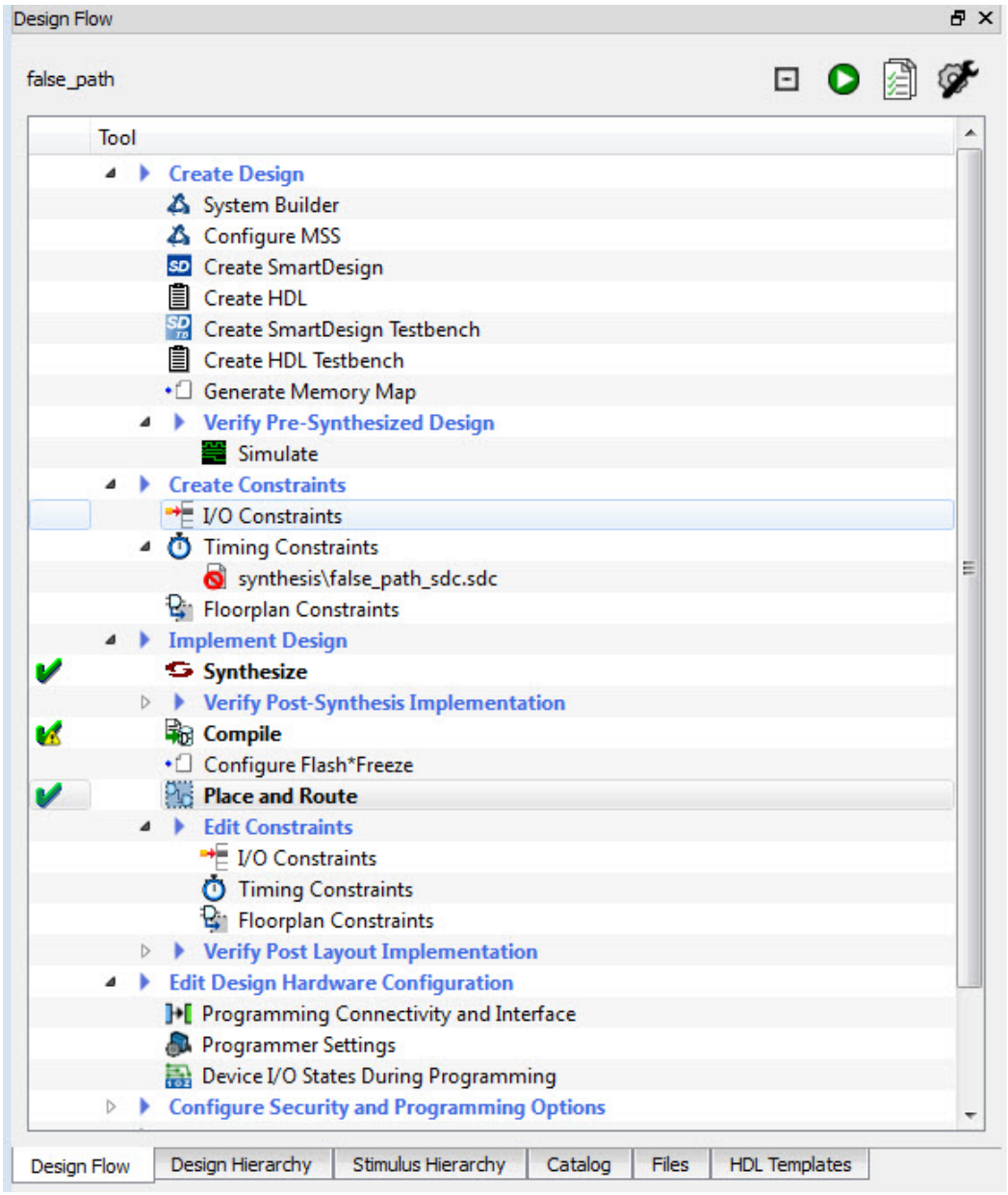


Figure 53 · Synthesize, Compile and Place and Route Successful Completion

Timing Analysis - Maximum Clock Frequency

The SmartTime Maximum Delay Analysis View displays the design maximum operating frequency and lists any setup violations.

To perform Maximum Delay Analysis:

1. Expand **Verify Post Layout Implementation**. Right-click **Verify Timing** and choose **Open Interactively** to open SmartTime. The Maximum Delay Analysis View appears (as shown in the figure below). The Maximum Delay Analysis View displays a summary of design performance and indicates that the design will operate at a maximum frequency of 442.48 MHz.

Note: You may see a slightly different maximum frequency with a different Libero software version.

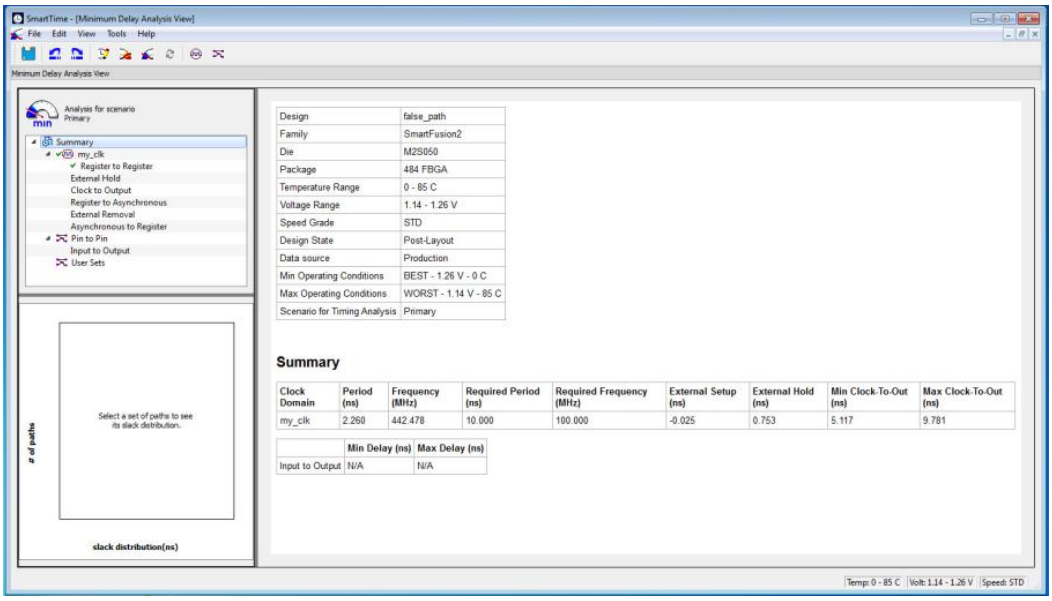


Figure 54 · Maximum Delay Analysis Summary

2. Expand **my_clk** to expand the display and show the Register to Register path sets.
3. Select **Register to Register** to display the register-to-register paths. Notice that the slack values are positive.
4. Click to select the row in the path list with the path is from the CLK pin of flip-flop D0_inv_reg to the D input of flip flop Q_reg. Note that the path goes through the S input of multiplexer un1_MUX2.

Figure 48 shows that the S input of un1_MUX2 will always be logic 1; consequently, all the paths through the 0 input of un1_MUX2 and the S input of un1_MUX2 are false paths. You must set a false path on these paths to determine the true maximum operating frequency.

5. To set the path from D0_inv_reg:CLK to Q_reg :D as false, select the row containing this path in the Register to Register path set, right-click and choose **Add False Path Constraint** (as shown in the figure below). The Set False Path Constraint dialog box appears.

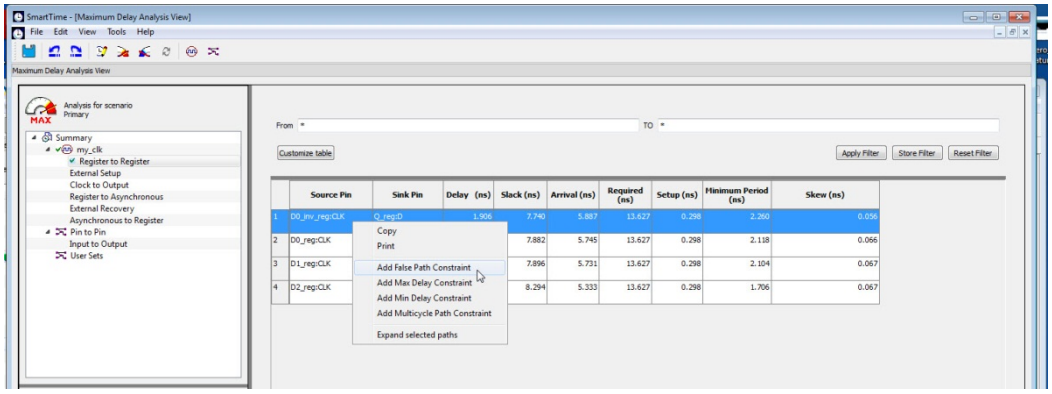


Figure 55 · Right-Click > Add False Path Constraint

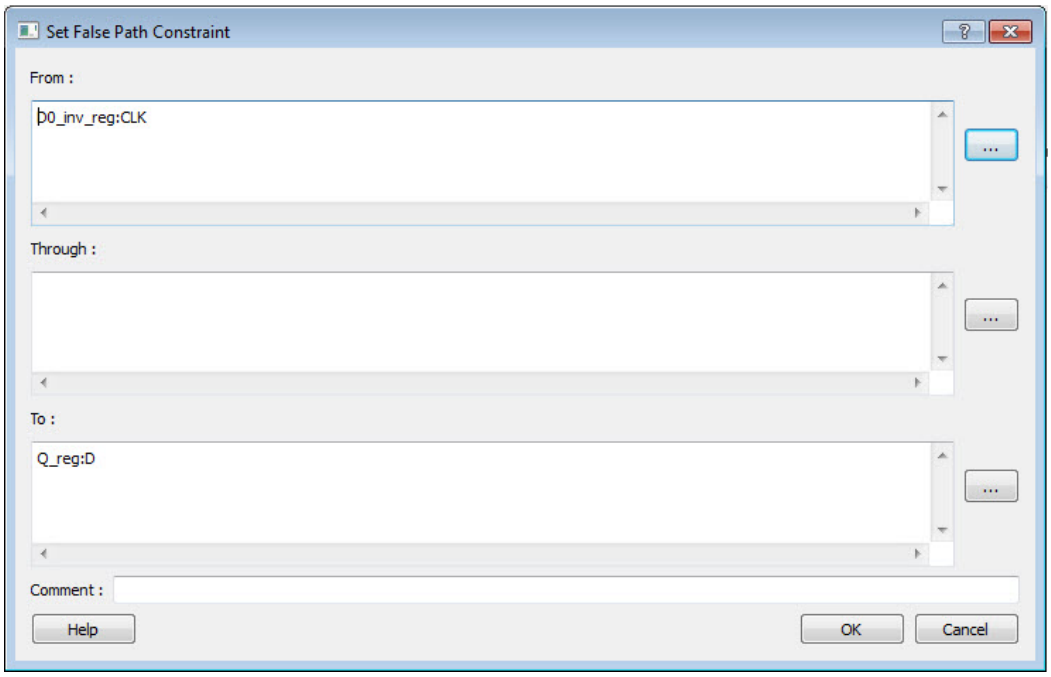


Figure 56 · Set False Path Constraint Dialog Box

6. **Click** OK to close the Set False Path Constraint dialog box.
7. From the **View** menu, choose **Recalculate All** to recalculate the delays.
8. There are a total of three register-to-register false paths in this design (see table below). Repeat steps 5 through 8 to set the false path constraint on the other two false paths (#2 and #3) using the values shown in the table.

	From	To
False Path #1	D0_inv_reg:CLK	Q_reg:D
False Path #2	D0_reg:CLK	Q_reg:D
False Path #3	D1_reg:CLK	Q_reg:D

9. Open the Constraint Editor (**Tools > Constraints Editor**) and click **Exceptions > False Path** (in the left pane under Constraints). The three False Path constraints are listed as shown in the figure below.

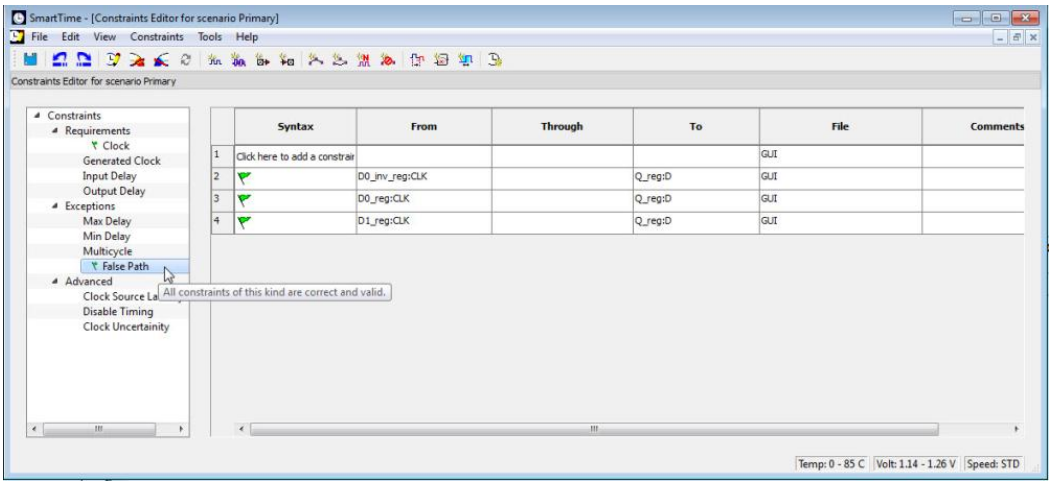


Figure 57 · False Path Constraints in the SmartTime Constraint Editor

10. View the summary in the Maximum Delay Analysis View (**Tools > Max Delay Analysis**). Note that SmartTime now reports the maximum operating frequency as 586.17 MHz, as shown in the figure below.

Note: The maximum operating frequency may vary slightly with a different version of the Libero software.

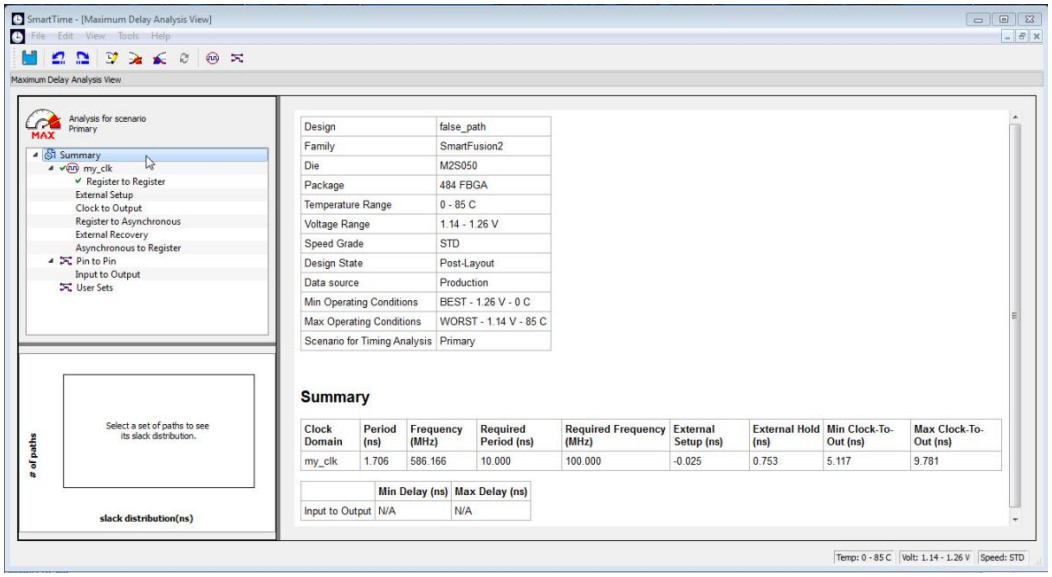


Figure 58 · Maximum Delay Analysis View - Summary

11. Select the Register to Register set for my_clk. Observe that only one path is visible, from D2_reg: CLK to Q_reg:D. This is the only path that propagates a signal (as shown in the figure below).

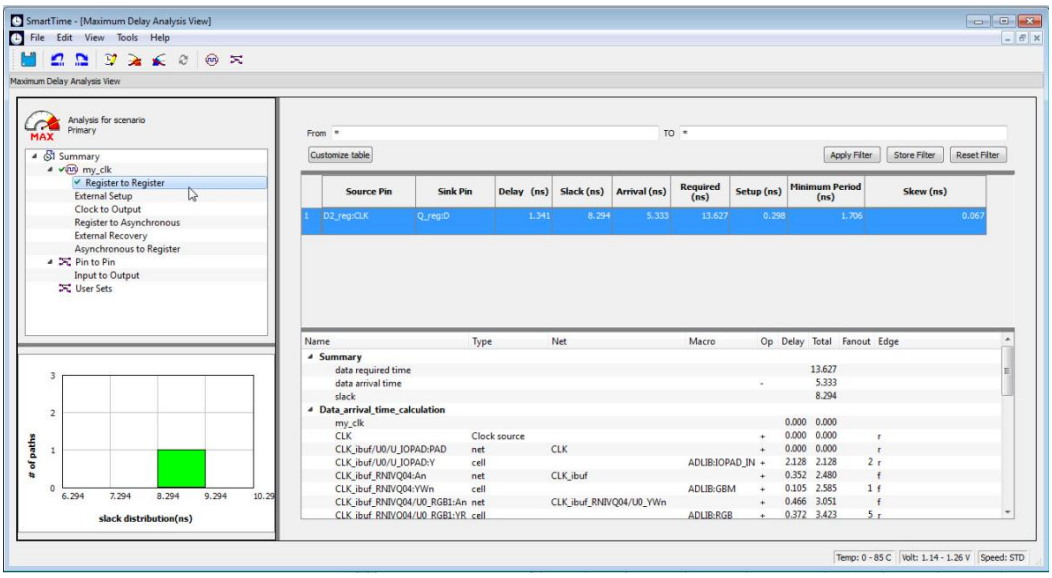


Figure 59 · Maximum Delay Analysis View - Register to Register

12. Close SmartTime.
13. Close Libero SoC.

Tutorial 5 - Cross Clock Domain Analysis

SmartTime performs inter-clock domain timing checks for designs that contain functional paths that cross two clock domains (the register launching the data and the register capturing the data are clocked by two different clock sources). Accurate specification of both clocks is required to allow a valid inter-clock domain timing check.

SmartTime analyzes each inter-clock domain by determining a common period equal to the least common multiple of the two clock periods.

For setup check, the tightest launch-capture time period is considered to ensure that the data arrives before the capture edge (as shown in the figure below). The hold check verifies that a setup relationship is not overwritten by a following data launch.

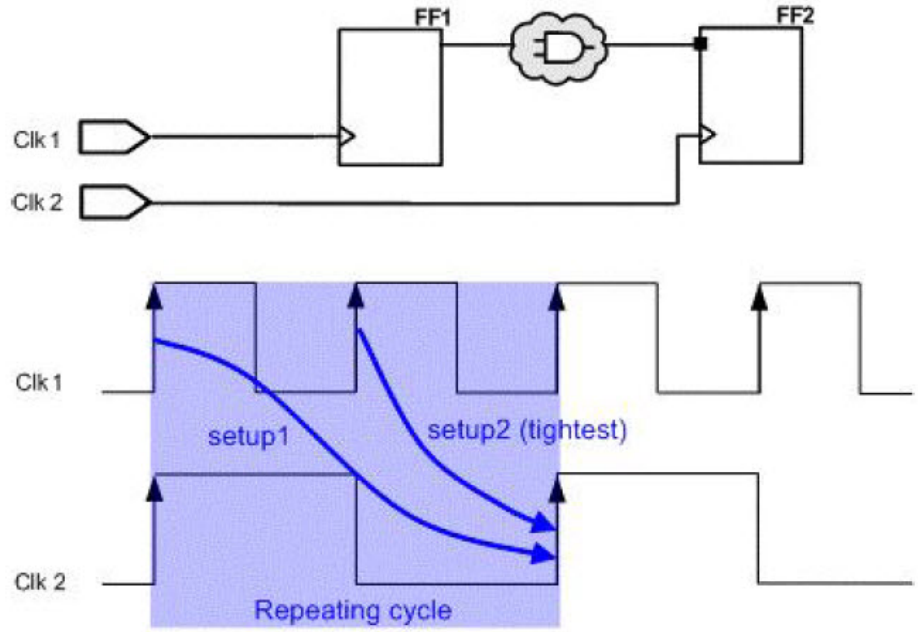


Figure 60 · Tightest Launch - Capture Relation for Setup Check in SmartTime

In this tutorial you will:

1. Create a new Libero project.
2. Import a Verilog Source for the design shown in the figure below.
3. Enter timing constraints for the two clock domains.
4. Use SmartTime to analyze the inter-clock domain timing.

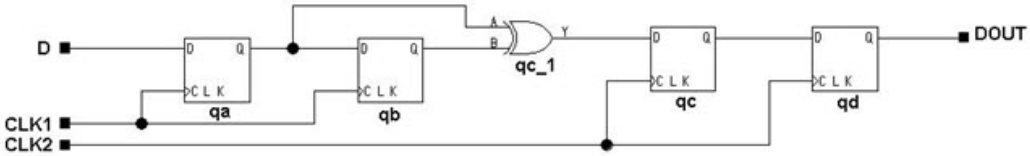


Figure 61 · Inter-Clock Domain Example Design Diagram

Set Up Your Cross Clock Domain Analysis Example Design Project

1. Open Libero and create a new project (from the **Project** menu, choose **New Project**).
2. Name the project **multi_clocks** and set the project location according to your preferences. Enter the following values for your new project:

- **Family**: SmartFusion2
- **Die**: M2S050
- **Package**: 484 FBGA
- **Speed**: STD
- **Core Voltage**: 1.2 V
- **Range**: COM

Leave all other fields at their default values.

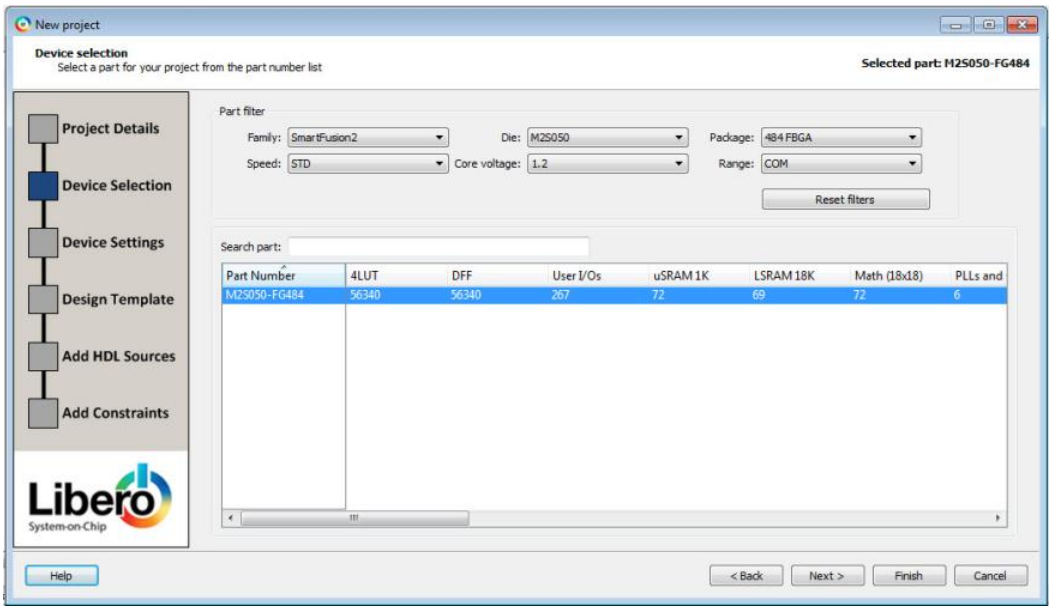


Figure 62 · multi_clocks Project Settings

3. Click **Finish** to create the new project.

Import Verilog Source File and Run Compile for Cross Clock Domain Analysis Example

You must import the multi_clk.v file into your design for this tutorial. Download the design files from the Microsemi website (Url link to tutorial file) .

To import and constrain the Verilog source file:

1. From the **File** menu choose **Import > HDL Source Files**.
2. Choose **HDL Source Files** from the file type drop-down list in the Import Files dialog box.
3. Browse to the location of the **multi_clk.v** file you have downloaded and select it. Click **Open** to import the file.
4. Verify that the file appears in the Design Hierarchy window of your project, as shown in the figure below.

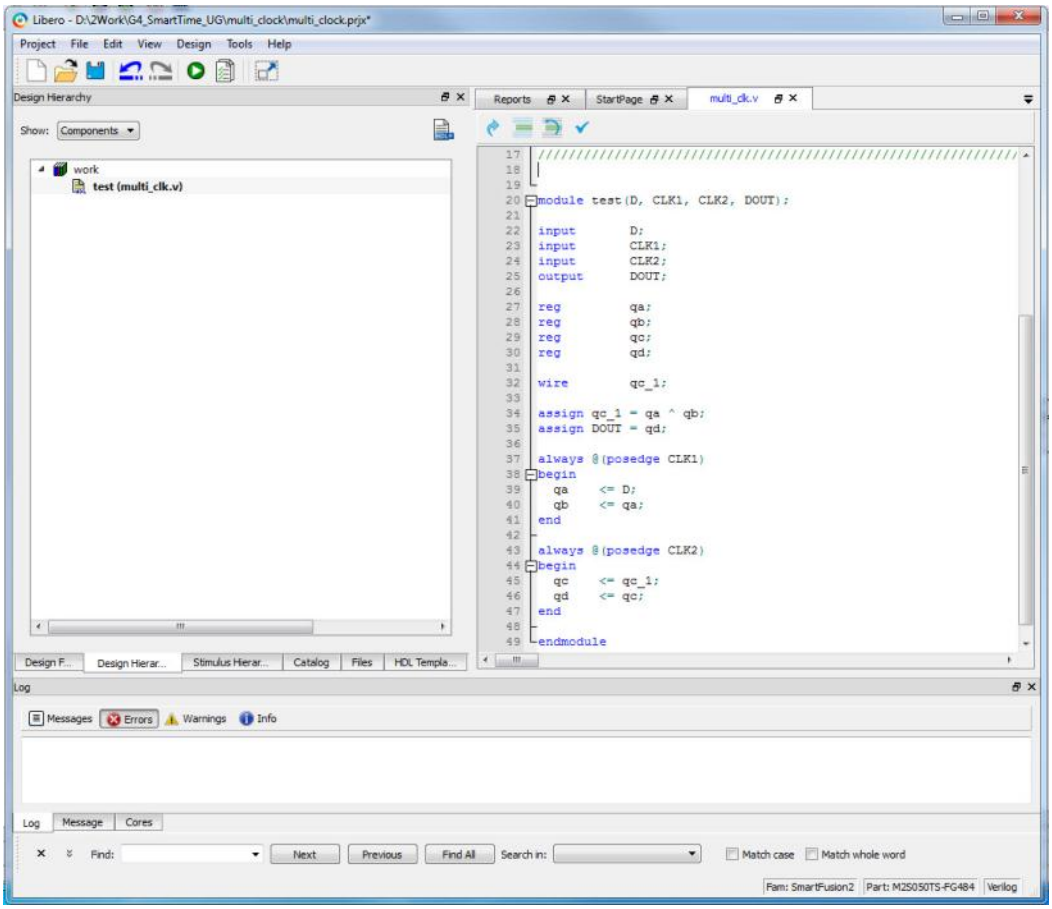


Figure 63 · multi_clocks Design in the Design Hierarchy Window and HDL Editor

5. Double-click **Compile** in the Design Flow window to run Synthesis first and then Compile with default settings. A green check mark appears next to Synthesize and Compile when they have run successfully, as shown in the figure below.

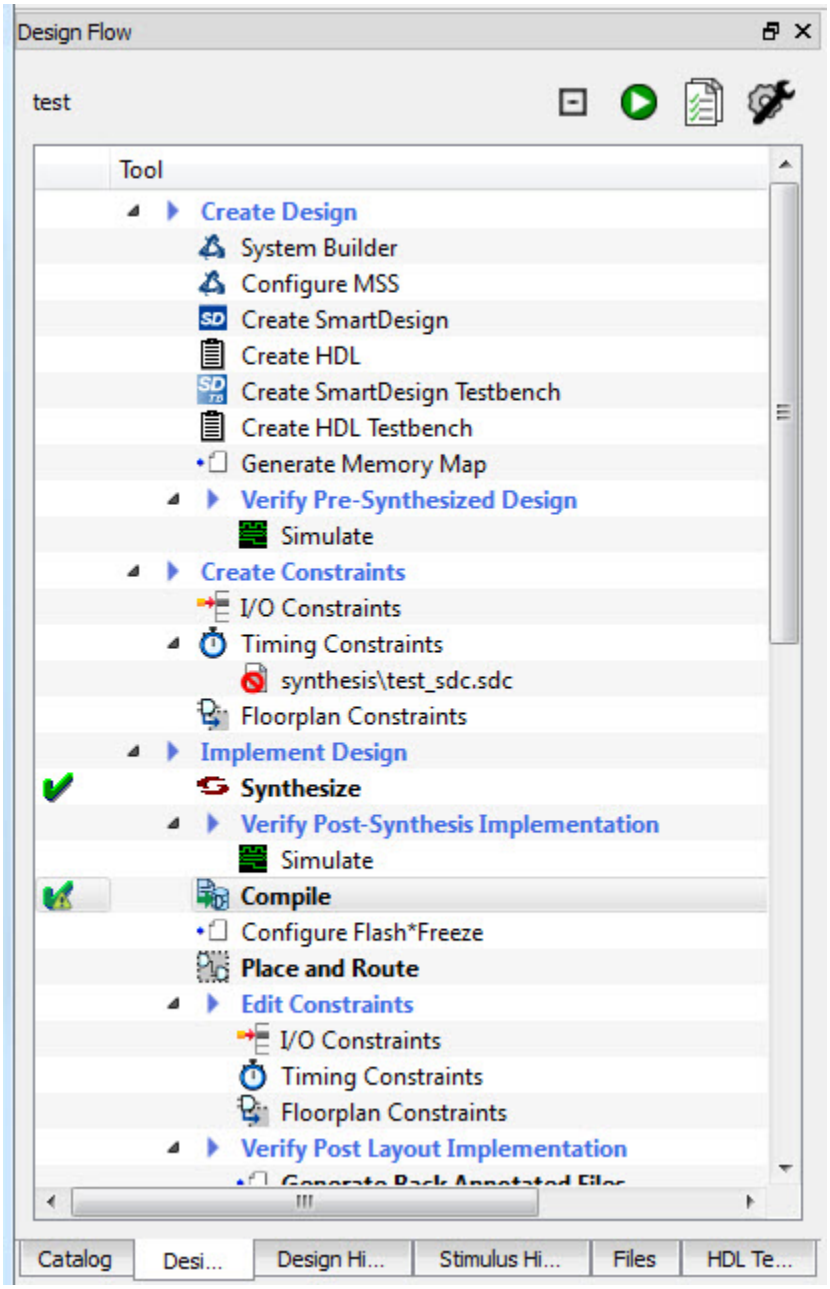


Figure 64 · Design Flow Window – Synthesize and Compile Successful

Enter Timing Constraints for the Cross Clock Domain Analysis Example

To add a clock constraint to your example design:

1. In the Design Flow window, expand **Edit Constraints**. Right-click on Timing Constraints and choose **Open Interactively** to open the SmartTime Constraints Editor.
2. In the Constraint Browser, expand **Requirements** and double-click **Clock** to enter the following clock constraints:

- Clock Source: **CLK1**, Clock Name: my_clk1, Frequency:250 MHz
- Clock Source: **CLK2**, Clock Name: my_clk2, Frequency:100 MHz

3. Verify that your new constraints are listed in the Constraints Editor, as shown in the figure below.

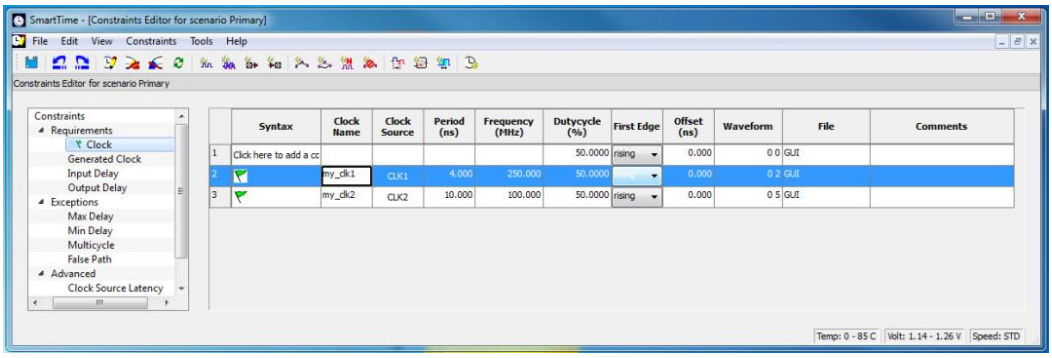


Figure 65 · Clock Constraints in the SmartTime Constraints Editor

4. Click **Save** to save your constraints.
5. Open the Max Delay Analysis View (**Tools > Max Delay Analysis**).
6. Choose **Tools > Options**. Ensure that the checkbox to Include inter-Clock domain analysis in calculations for timing analysis is enabled (default) as shown in the figure below.

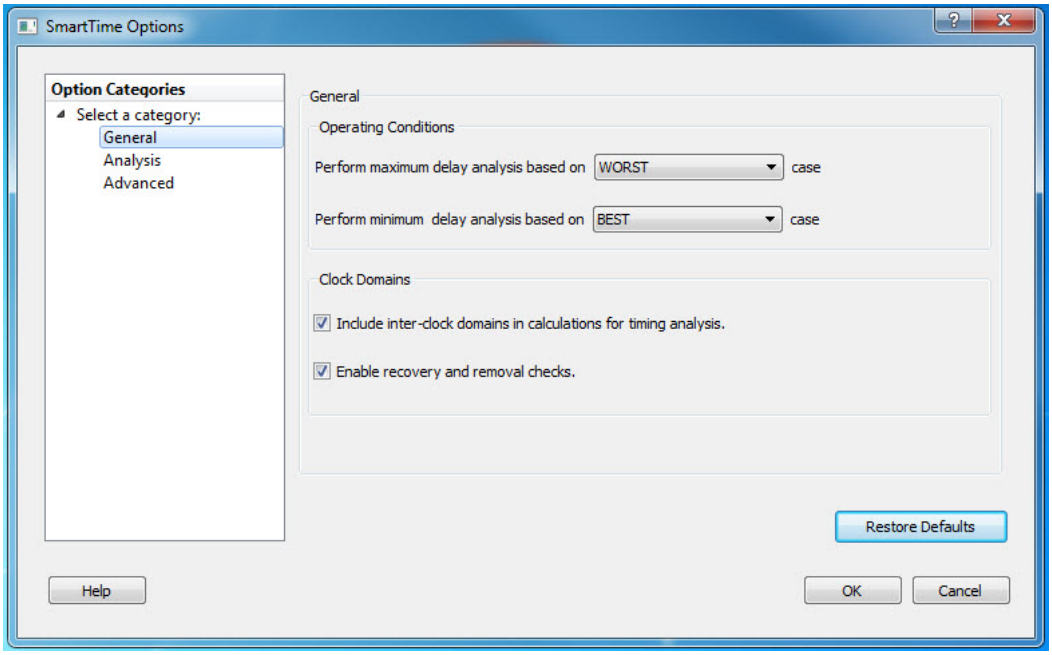


Figure 66 · Inter-Clock Domain Analysis Enabled in the SmartTime Options Dialog Box

7. Exit SmartTime (**File > Exit**).

Place and Route Your Cross Clock Domain Analysis Example

To run Place and Route on multi_clocks:

1. Right-click **Place and Route** in the Design Flow window and choose **Configure Options**.
2. Click the checkbox to enable Timing-Driven layout in Layout Options and leave the other values at the default settings, as shown in the figure below. Click **OK** to close the Layout Options dialog box.
3. Right-click **Place and Route** and choose **Run**.

A green check mark appears next to Place and route when it completes successfully.

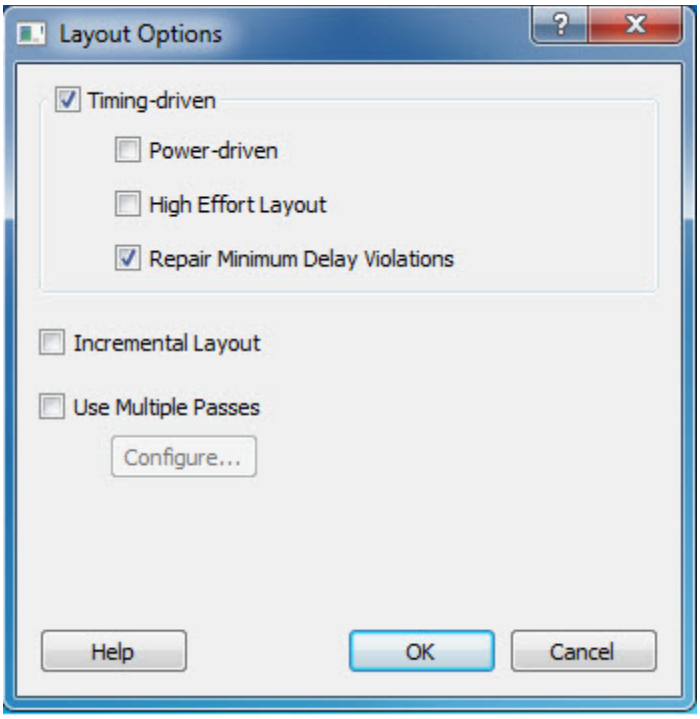


Figure 67 · Layout Options - multi_clocks

Analyze Inter-Clock Domain Timing

Inter-clock domain timing enables you to analyze timing for designs that contain functional paths that cross two clock domains.

To analyze inter-clock domain timing:

1. Right-click **Verify Timing** and choose **Open Interactively** to open the Maximum Delay Analysis View.
2. Expand the **my_clk2** path in the Maximum Delay Analysis View. Click to select the **my_clk1 to my_clk2** path and observe the inter-clock domain path timing (as shown in the figure below).

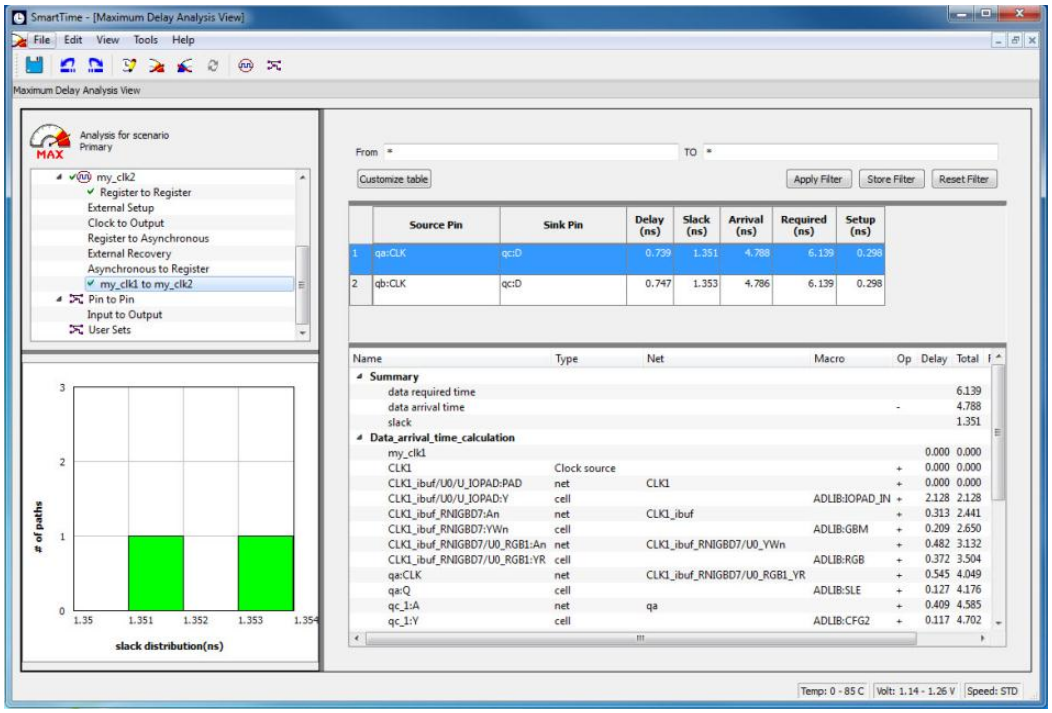


Figure 68 · Maximum Delay Inter-Clock Domain Timing Analysis - multi_clocks Example Design

The Paths list shows the detailed timing analysis. The longest reported path is from qa:clk to qc:d, as shown in the figure below. This path has a slack of 1.351 ns. The clock edges used in the calculation are shown in the timing diagram below.

Note: The actual slack value may vary with different die size and different Libero SoC versions.

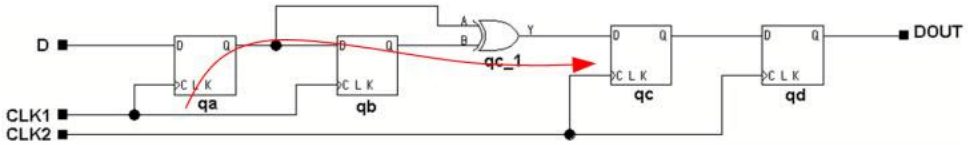


Figure 69 · Longest Reported Inter-Clock Domain Path - multi_clocks Example Design

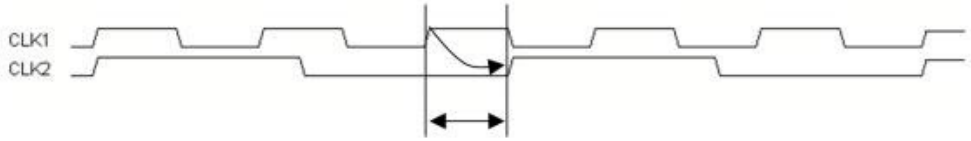


Figure 70 · Clock Edges Used in Inter-Clock Domain Max Delay Calculation - multi_clocks Example Design

3. Click **Tools** and choose **Minimum Delay Analysis**.
4. Expand the CLK2 paths in the Minimum Delay Analysis View. Click to select **my_clk1 to_my_clk2** path and observe the inter-clock domain path timing, as shown in the figure below.

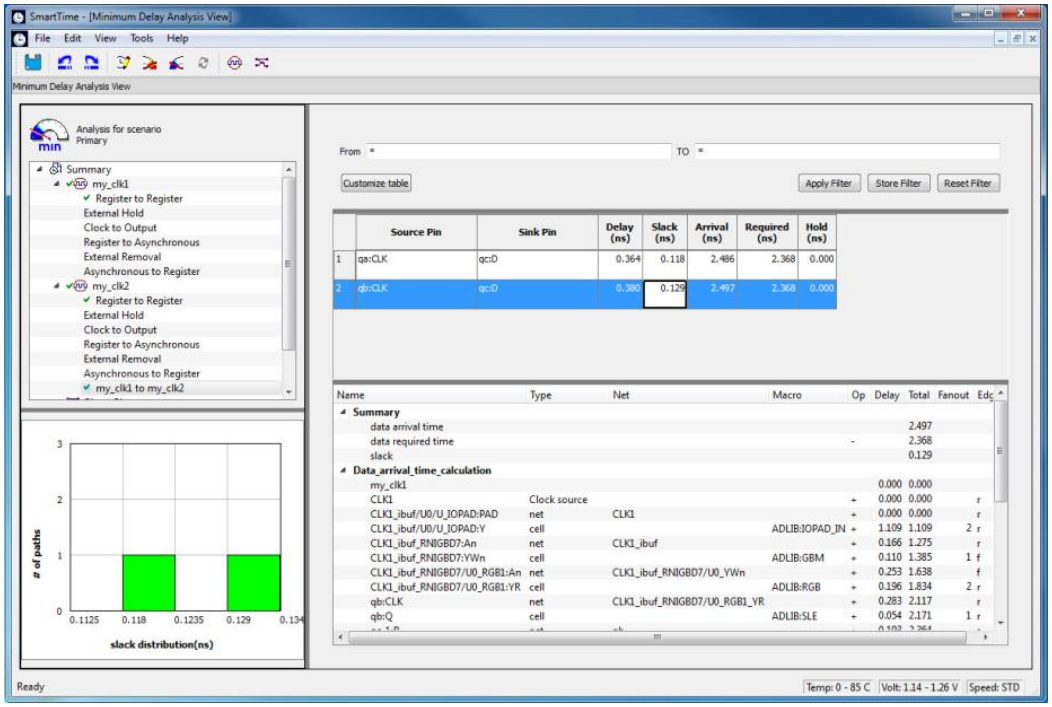
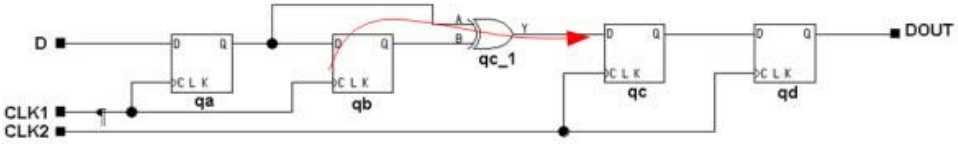


Figure 71 · Minimum Delay Inter-Clock Domain Timing Analysis - multi_clocks Example Design

The Paths list shows the detailed timing analysis. The shortest reported path is from qb:clk to qc:d, as shown in the figure below. This path has a positive slack of 0.129 ns. The clock edges used in the calculation are shown.

Note: The actual slack value may vary with different die sizes and different versions of Libero SoC.



Shortest Reported Inter-Clock Domain Path - multi_clocks Example Design

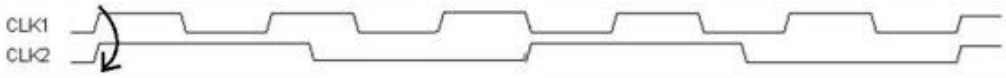


Figure 72 · Clock Edges Used in Inter-Clock Domain Min Delay Calculation - multi_clocks Example Design

5. Exit SmartTime (**File > Exit**).
6. Exit Libero (**Project > Exit**).

Editable Constraints Grid

The Constraints Editor allows you to add, edit and delete constraints directly from the Constraints Editor View.

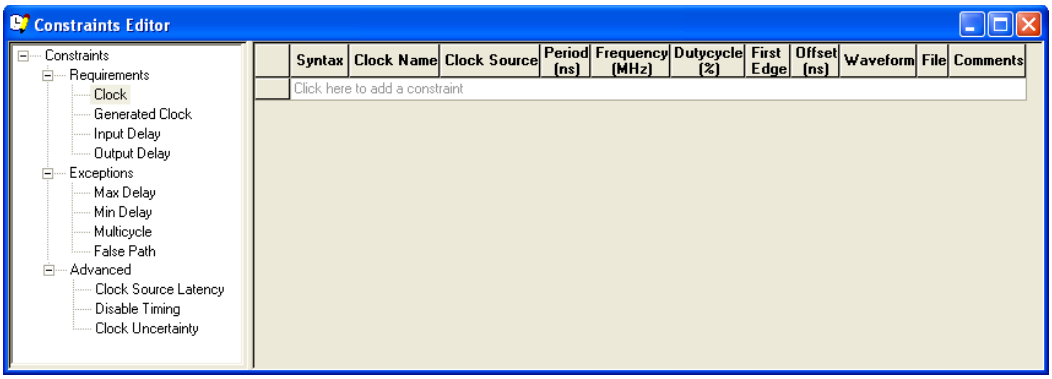


Figure 73 · Constraints Editor View

To add a new constraint:

1. Select a constraint type from the constraint browser.
2. Enter the constraint values in the first row and click **OK** . Click the Save icon.
3. The new constraint is added to the Constraint List. The green syntax flag indicates that the syntax check on the constraint was successful.

To edit a constraint:

1. Select a constraint type from the constraint browser.
2. Select the constraint, edit the values and click **Save**. The green syntax flag indicates that the syntax check for the constraint was successful.

To delete a constraint:

1. Select a constraint type from the constraint browser.
2. Select the constraint you want to delete and from the right-click menu, select **Delete Constraint**.

SmartTime Constraints Editor

Components of the SmartTime Constraints Editor

SmartTime Constraints Editor is a tool in the Libero software that enables you to create, view, and edit timing constraints of the selected scenario for use with SmartTime timing analysis and timing-driven optimization tools. This editor includes powerful visual dialogs that guide you toward capturing your timing requirements and timing exceptions quickly and correctly. In addition, it is closely connected to the SmartTime Timing Analysis View, which enables you to analyze the impact of constraint changes.

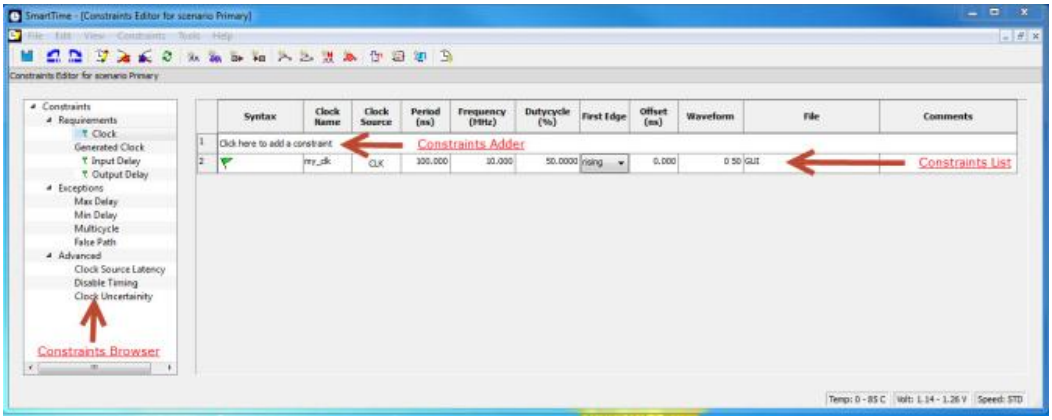


Figure 74 · SmartTime Constraints Editor View

Constraint Hierarchy Browser

The SmartTime Constraints Editor window is divided into a Constraint Browser and a Constraint List. The Constraint Browser categorizes constraints based on requirements, exceptions and advanced categories, while the Constraint List provides details about each constraint and enables the user to add, edit and delete constraints.

You can perform the following tasks in the SmartTime Constraints View:

- Select a constraint type from the Constraint Browser and create or edit the constraint.
- Add a new constraint and check the syntax.
  To add a constraint, double-click on the constraint type. To edit a constraint , select the constraint from the constraint list, right-click and choose **Edit Constraint**.
- Select a row and right-click to display the shortcut menu, which you can use to edit, delete, or copy the selected constraint to a spreadsheet.
- Select the entire spreadsheet and copy it to another spreadsheet.

See Also

Editable Grid and Quick Adder

SmartTime scenarios

Constraint Wizard

The SmartTime Constraint Wizard enables you to quickly and easily create clock and timing I/O constraints for your design.

To open the Constraint Wizard (shown below) from the SmartTime **Tools** menu, click the **Constraint Wizard** icon  . This window can be resized.

Constraint Wizard

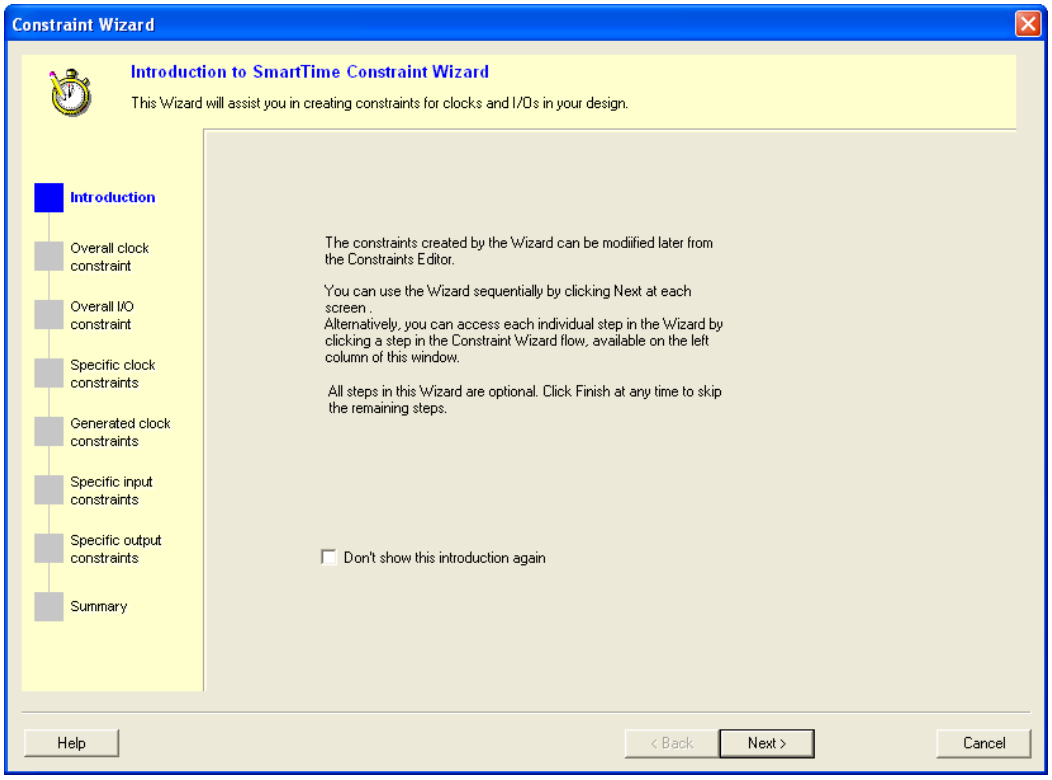


Figure 75 · Constraint Wizard

This window provides information about the Constraint Wizard and how to use it. Check the **Don't show this introduction again** box to skip this window next time you use this wizard.

Press **Next** to continue to the next step in the wizard.

Note: All steps in this Wizard are optional; click Finish to exit the wizard.

Overall Clock Constraint

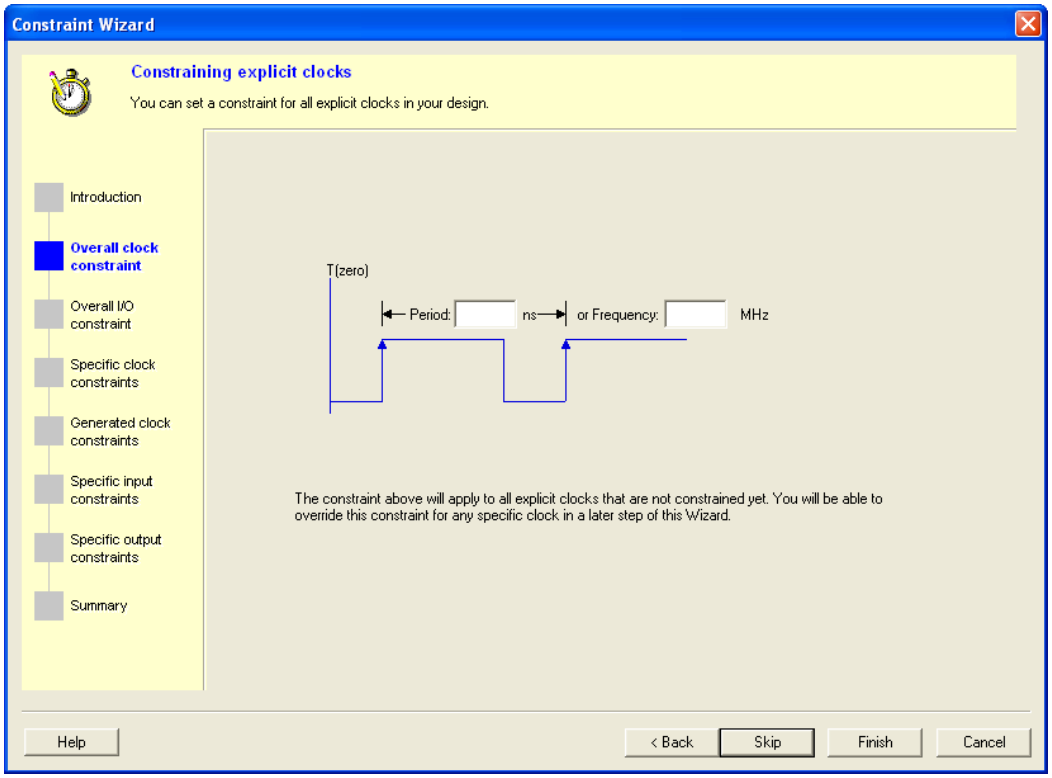


Figure 76 · Constraint Wizard – Overall Clock Requirements

In this window you can set a default required period or frequency for all explicit clocks in your design. Clocks that already have a constraint will not be affected.

To set a constraint for all explicit clocks, enter the **Period** or the **Frequency**, and click **Next** to go to the next step or **Finish** to exit the wizard.

Overall I/O Constraint

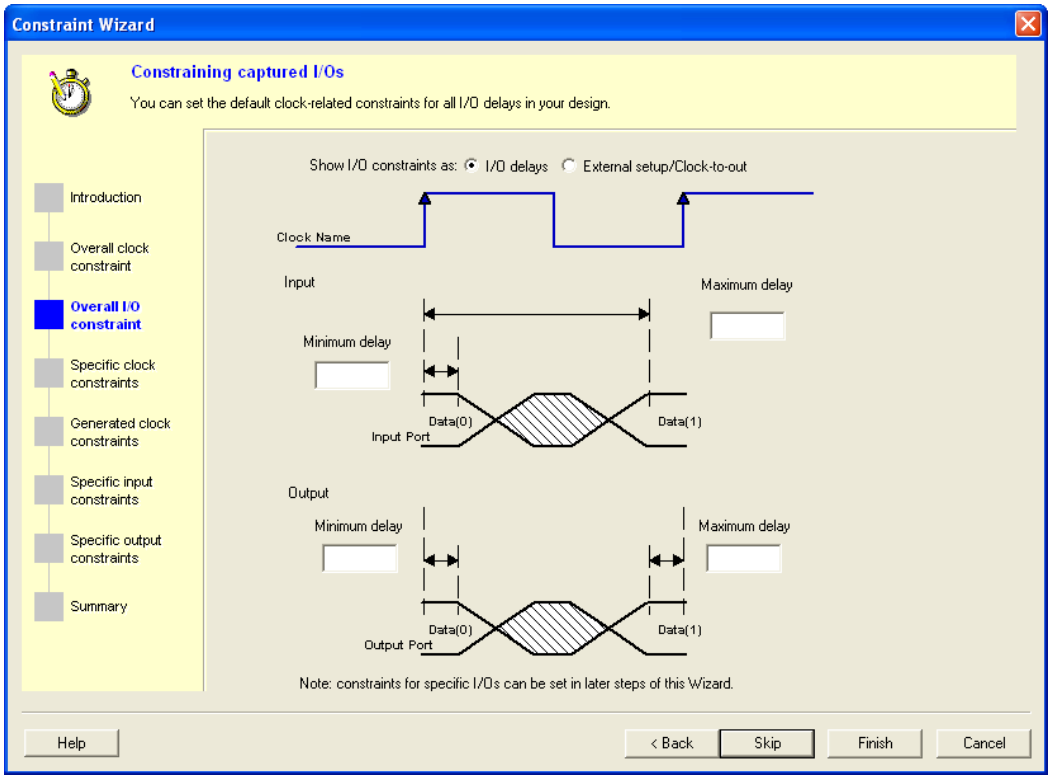


Figure 77 · Constraint Wizard – Overall I/O Constraint

In this window you can set a default constraint for all I/O's in the design. Constraints will be applied with respect to clocks related to the I/O's. This constraint will not override existing I/O constraints.

Show I/O constraints enables you to display I/O constraints as I/O delays (minimum and maximum delays for input and output) or external setup/clock-to-out.

To set a constraint for all I/Os:

1. Enter the **Maximum** and/or **Minimum** delays for the **Input** and/or **Output**.
2. Click **Next** to go to the next step or **Finish** to exit the wizard.

Specific Clock Constraints

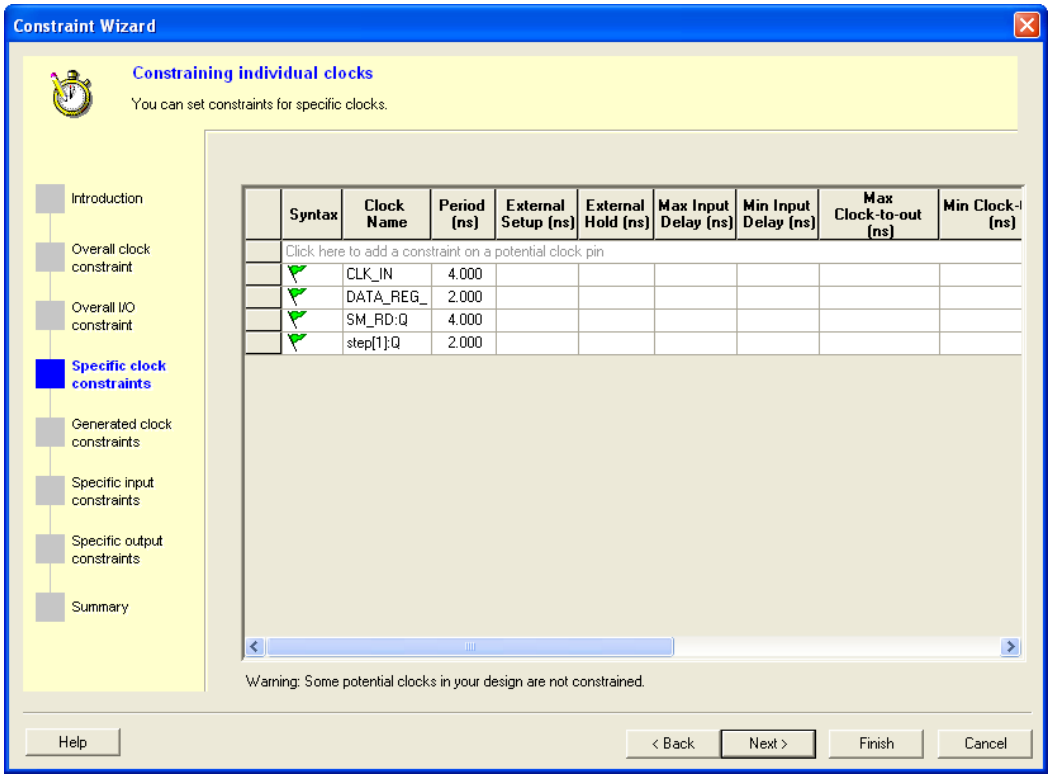


Figure 78 · Constraint Wizard – Specific Clock Constraints

In this window you can set a period and I/O timing constraints for a specific clock domain. All I/Os within the domain will be affected by the I/O timing constraints. You can modify the constraints from the grid.

To add a constraint for a potential clock:

1. Click the first row in the grid, enter the constraint information, and click the green check mark.
2. Click **Next** to go to the next step or **Finish** to exit the wizard.

Note: This option is available only when there is a potential clock in your design.

Generated Clock Constraints

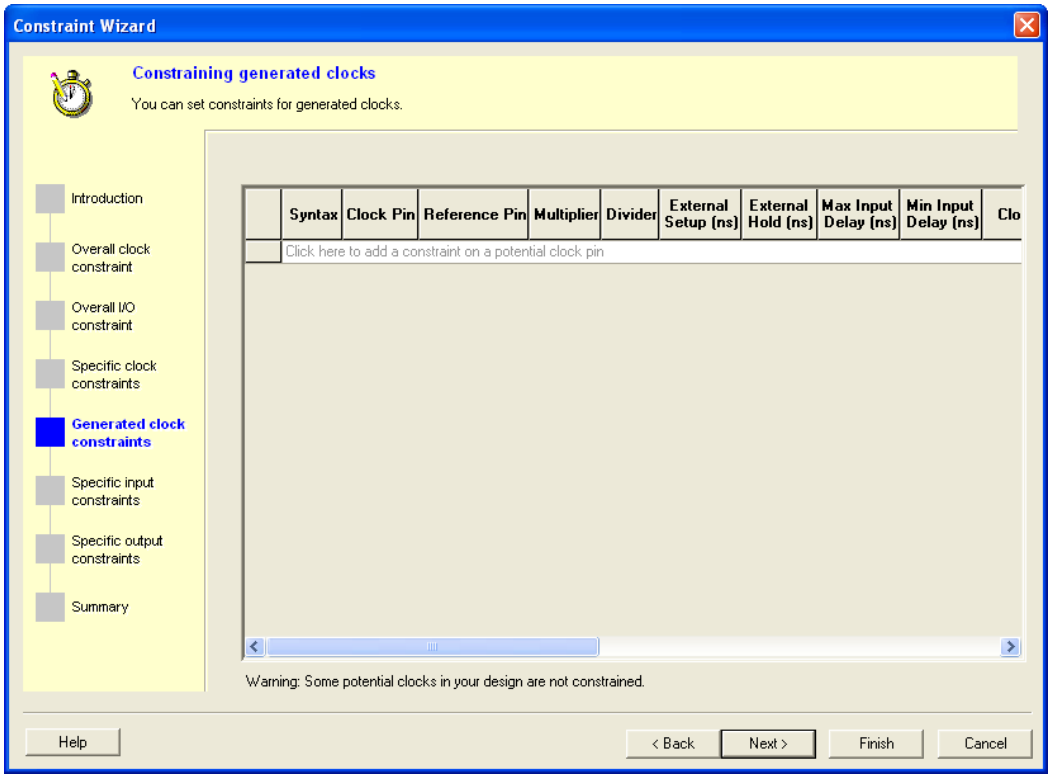


Figure 79 · Constraint Wizard – Generated Clock Constraints

In this window you can set a period and I/O timing constraints for a specific generated clock domain. You can modify the constraints from the grid.

To add a constraint for a generated clock:

1. Click the first row in the grid, enter the constraint information, and click the green check mark.
2. Click **Next** to go to the next step or **Finish** to exit the wizard.

Note: This option is available only when there is a generated clock in your design.

Specific Input Constraints

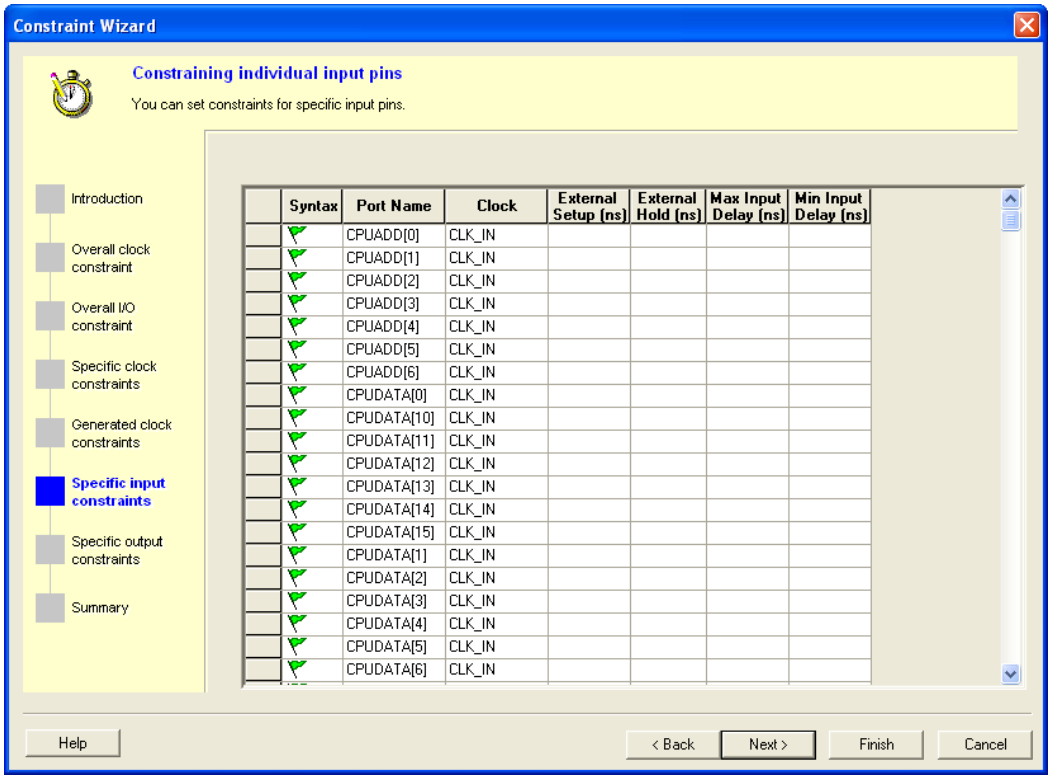


Figure 80 · Constraint Wizard – Specific Input Constraints

In this window you can set constraints for specific input pins. You can modify the constraints from the grid.

To set a constraint for an input pin:

1. Set the maximum and/or minimum input delay for selected pin in the grid.
2. Click **Next** to go to the next step or **Finish** to exit the wizard.

Note: This option is available only when there is an input pin in your design.

Double-click the **Port Name** or **Clock** header in the table to change the sorting order by input port name or clock name.

Specific Output Constraints

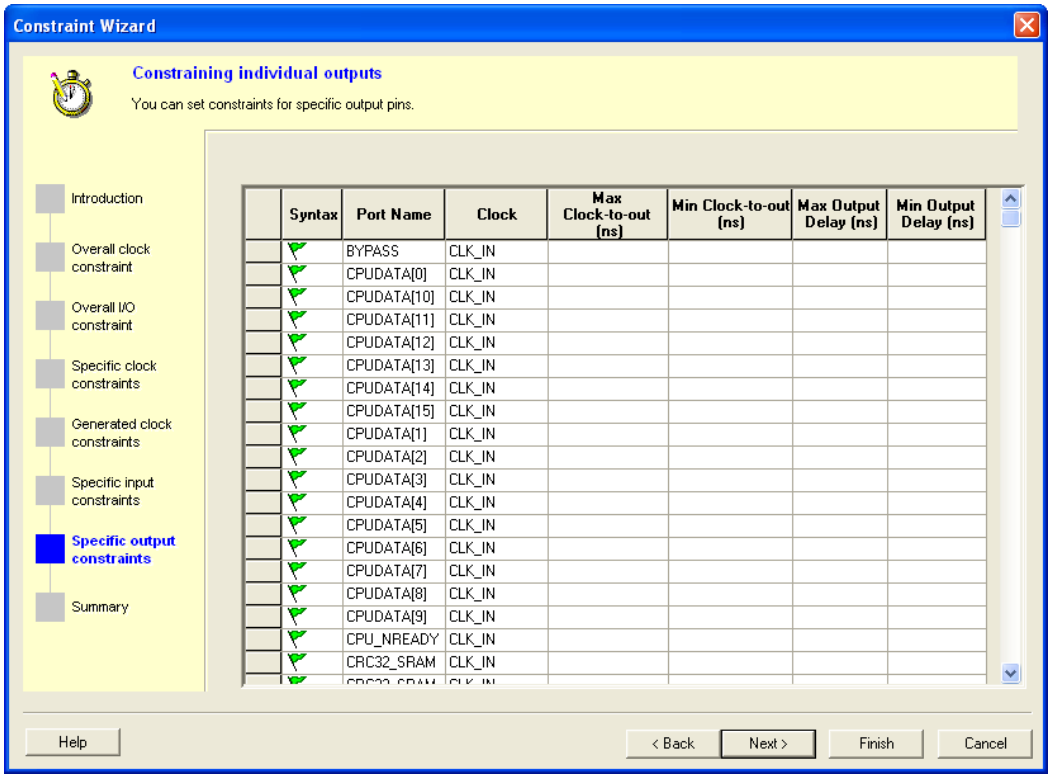


Figure 81 · Constraint Wizard – Specific Output Constraints

In this window you can set constraints for specific output pins. You can modify the constraints from the grid.

To set a constraint for an output pin:

1. Set the maximum and/or minimum output delay for selected pin in the grid.
2. Click **Next** to go to the next step or **Finish** to exit the wizard.

Note: This option is available only when there is an output pin in your design.

Double-click the **Port Name** or **Clock** header in the table to change the sorting order by output port name or clock name.

Summary

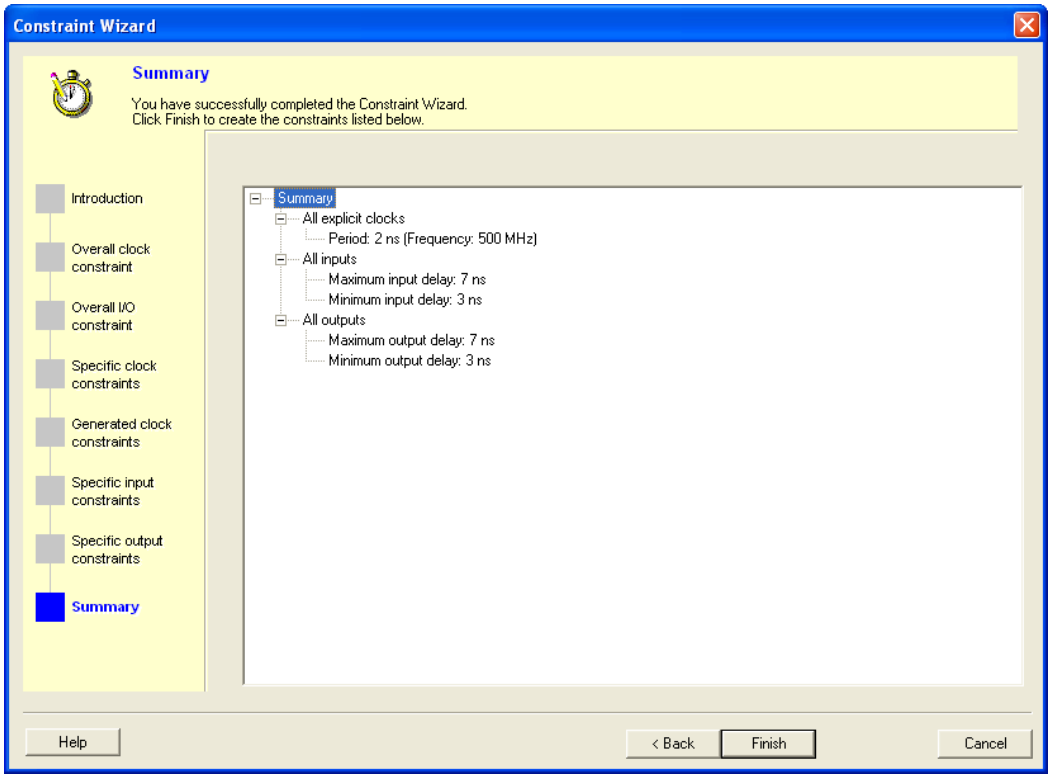


Figure 82 · Constraint Wizard – Summary

This window summarizes the requirements specified in the wizard and information about all clock and I/O constraints in the design.

Click **Finish** to create the constraints.

See Also

Editable Constraints Grid

Using Clock Types

Clock constraints enable you to specify your clock sources and clock requirements, such as the frequency and duty cycle. SmartTime detects possible clocks by tracing back the design from the clock pins of all sequential components until it finds an input port, the output of another sequential element, or the output of a PLL. SmartTime classifies clock sources into three types:

- Explicit Clocks
- Potential Clocks
- Clock Networks

Grouping clocks into these three types helps you manage clock domains efficiently when you add a new clock domain for analysis or when you create a new clock constraint using the Select Source Pins for Clock Constraint dialog box (as shown below).

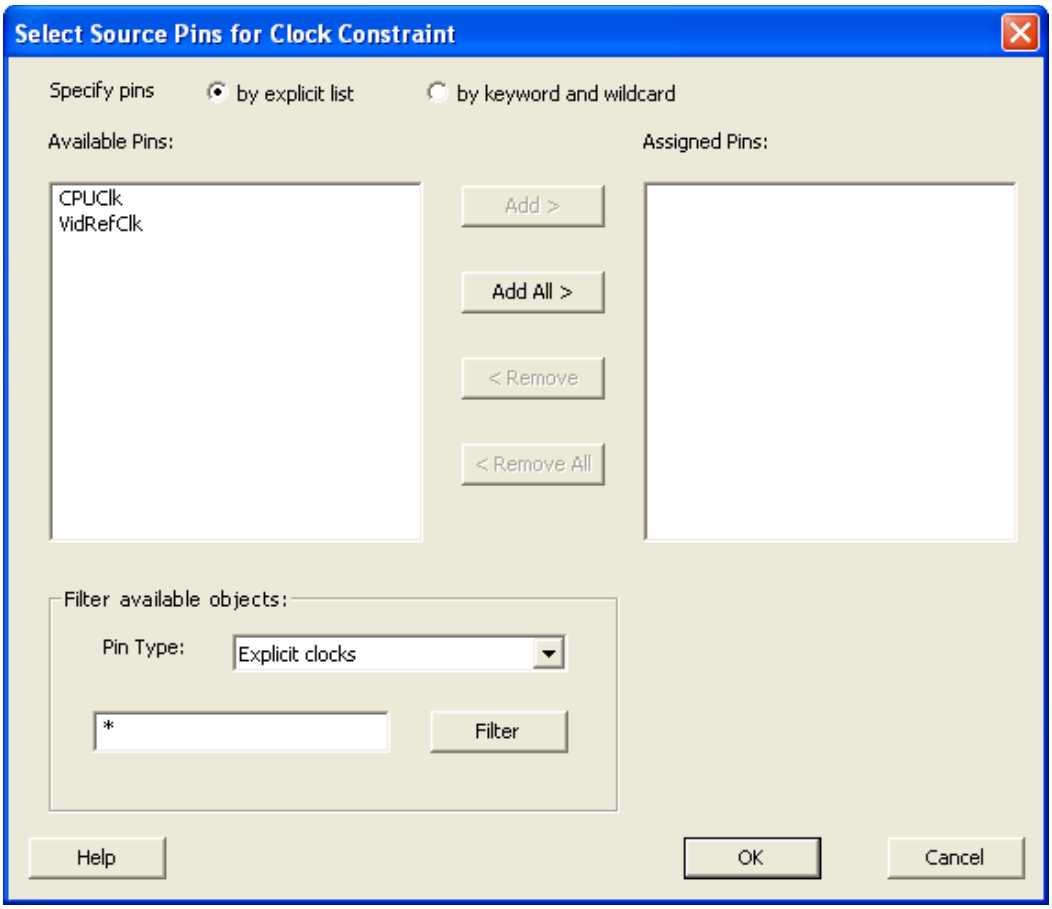


Figure 83 · Select Source Pins for Clock Constraint Dialog Box

See Also

Select Source Pins for Clock Constraint Dialog Box

Understanding Explicit Clocks

Understanding Potential Clocks

Understanding Clock Networks

Understanding Explicit Clocks

Explicit clocks are pins or ports connected to the clock pin of one or more sequential component, and where each clock is one of the following:

- The output of a PLL
- An input port that does not get gated between the source and the clock pins it drives
- The output pin of a sequential element that does not get gated between the source and the clock pins it drives
- Any pin or port on which a clock constraint was specified

By default, SmartTime displays domains with explicit clocks in the Timing Analysis View. You can browse these domains in the Domain Browser of the Timing Analysis View.

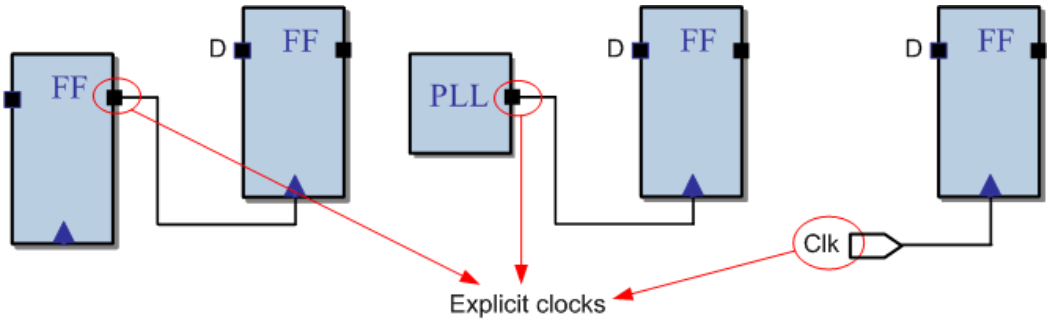


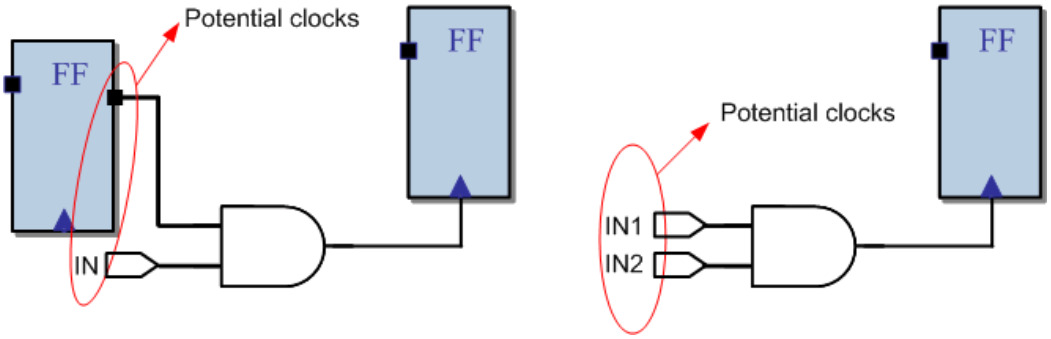
Figure 84 · Explicit Clocks

See Also

Choose the Clock Source Dialog Box
Using Clock Types
Understanding Potential Clocks
Understanding Clock Networks

Understanding Potential Clocks

Potential clocks are the clock sources that could be either enabled sources or clock sources. This type of clock is generally associated with the use of gated clocks. When associated with gated clocks, SmartTime cannot differentiate between the enabled sources and clock sources. Both sources appear in the potential clocks list and not the explicit clocks list.



See Also

Choose the Clock Source Dialog Box
Using Clock Types
Understanding Explicit Clocks
Understanding Clock Networks

Understanding Clock Networks

Clock networks are internal clock network pins used as a clock source. With this network type, you can set the clock constraint on any pin in the clock network. You may want to do this to eliminate clock network pessimism by short-cutting a reconvergent combinational logic on the clock network (as shown below). Clock network pessimism triggers an overestimation of the clock skew, making the timing analysis inaccurate.

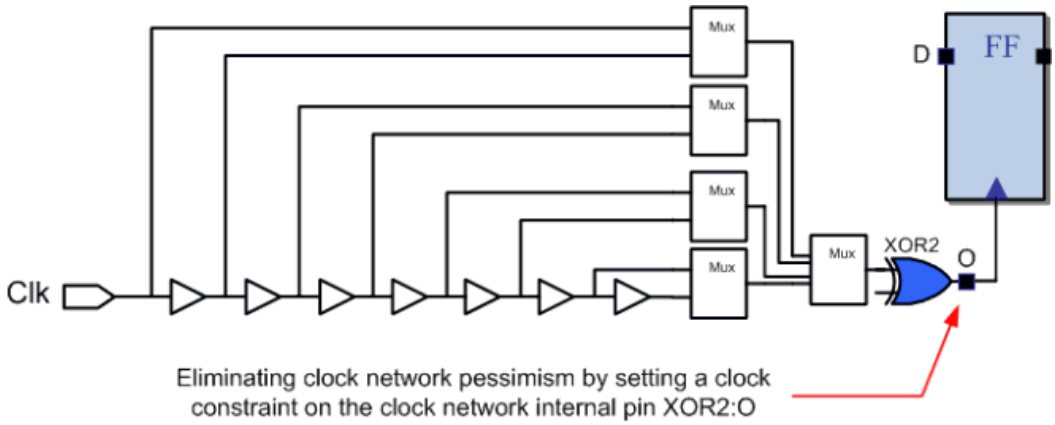


Figure 85 · Source Being Set Within a Clock Network

See Also

[Choose the Clock Source Dialog Box]

[Using Clock Types]

[Understanding Potential Clocks]

[Understanding Clock Networks]

Specifying Clock Constraints

Specifying clock constraints is the most effective way to constrain and verify the timing behavior of a sequential design. Use clock constraints to meet your performance goals.

To specify a clock constraint:

1. Add the constraint in the [editable constraints grid] or open the [Create Clock Constraint] dialog box using one of the following methods:
   - Click the  icon in the Constraints Editor.
   - Right-click the **Clock** in the Constraint Browser and choose **Add Clock Constraint**.
   - Double-click **Clock** in the Constraint Browser.

   The Create Clock Constraint dialog box appears (as shown below).

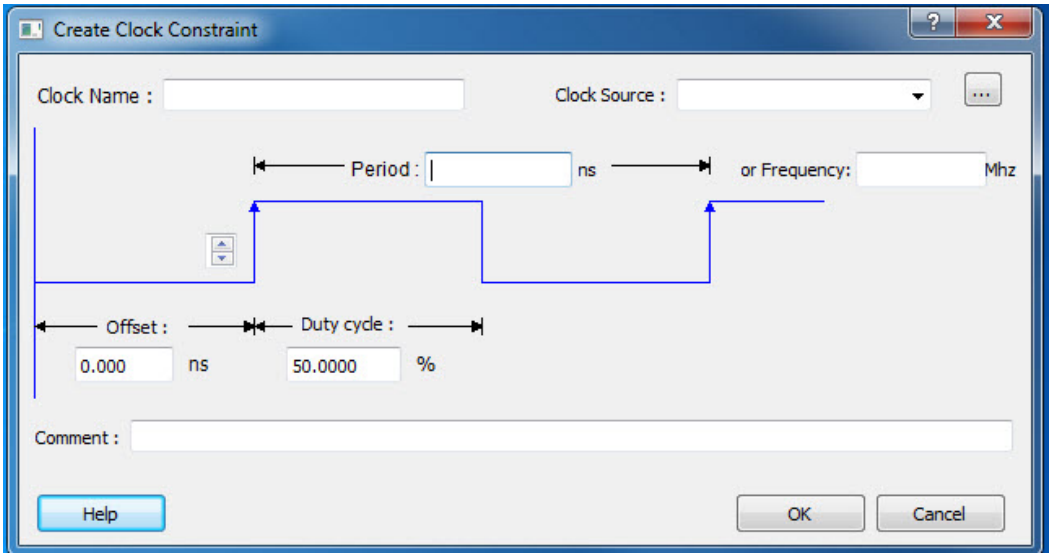


Figure 86 · Create Clock Constraint Dialog Box

2. Select the pin to use as the clock source. You can click the **Browse** button to display the Select Source Pins for Clock Constraint Dialog Box (as shown below).

   Note: Do not select a source pin when you specify a virtual clock. Virtual clocks can be used to define a clock outside the FPGA that it is used to synchronize I/Os.

   Use the Choose the Clock Source Pin dialog box to display a list of source pins from which you can choose. By default, it displays the explicit clock sources of the design. To choose other pins in the design as clock source pins, select **Filter available objects - Pin Type** as **Explicit clocks**, **Potential clocks**, **All Ports**, **All Pins**, **All Nets**, **Pins on clock network**, or **Nets in clock network**. To display a subset of the displayed clock source pins, you can create and apply a filter.
   Multiple source pins can be specified for the same clock when a single clock is entering the FPGA using multiple inputs with different delays.
   Click **OK** to save these dialog box settings.
3. Specify the **Period** in nanoseconds (ns) or **Frequency** in megahertz (MHz).
4. Modify the **Clock Name**. The name of the first clock source is provided as default.
5. Modify the **Duty cycle**, if needed.
6. Modify the **Offset** of the clock, if needed.
7. Modify the first edge direction of the clock, if needed.
8. Click **OK**. The new constraint appears in the Constraints List.

   Note: When you choose File > Save, SmartTime saves the newly created constraint in the database.

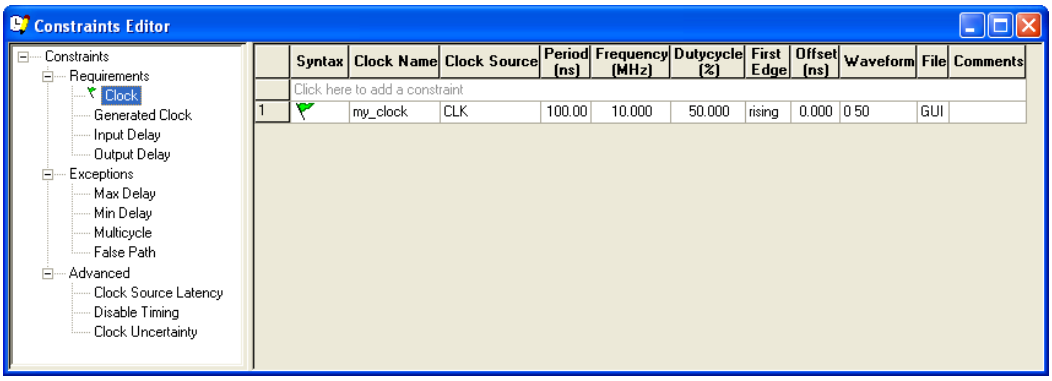


Figure 87 · SmartTime Timing Constraint View

See Also

Clock definition

Create a Clock

Create Clock Constraint Dialog Box

Specifying Generated Clock Constraints

Specifying a generated clock constraint enables you to define an internally generated clock for your design and verify its timing behavior. Use generated clock constraints and clock constraints to meet your performance goals.

To specify a generated clock constraint:

1. Open the Create Generated Clock Constraint dialog box using one of the following methods:

   - Click the  icon.
   - Right-click the **GeneratedClock** in the Constraint Browser and choose **Add Generated Clock**.
   - Double-click the Generated Clock Constraints grid. The Create Generated Clock Constraint dialog box appears (as shown below).

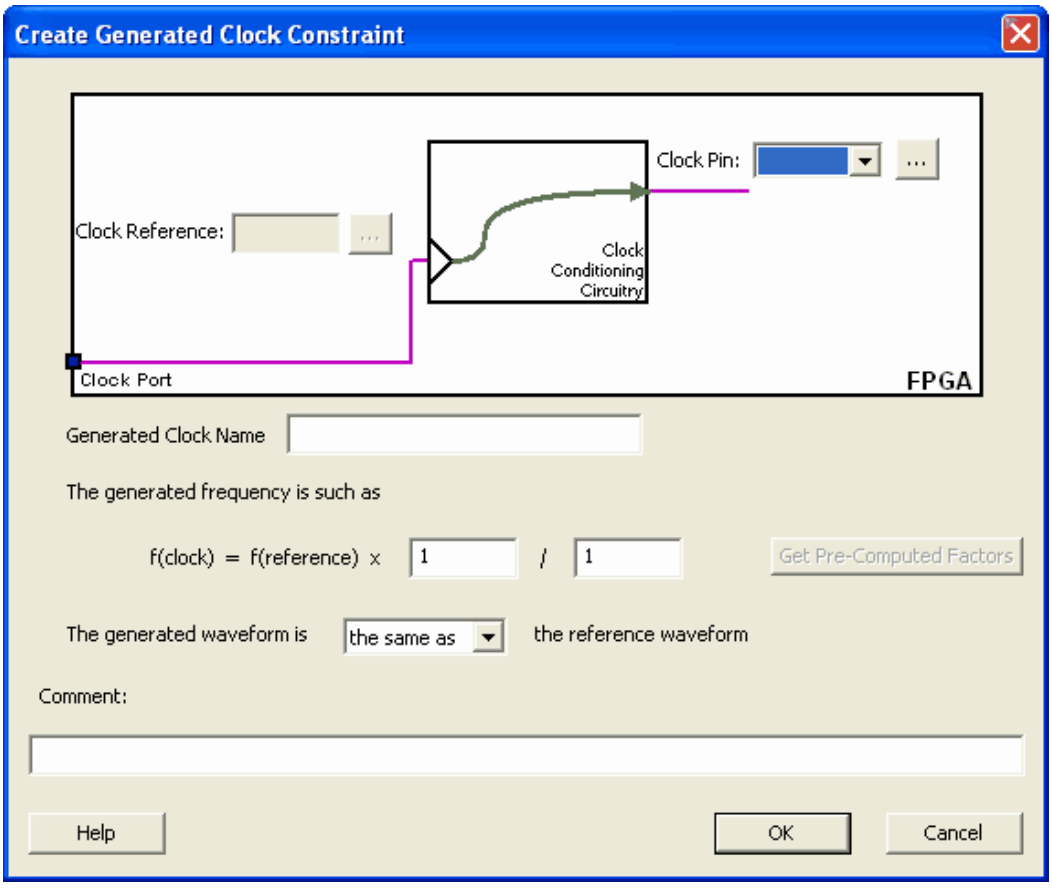


Figure 88 · Create Generated Clock Constraint

2. Select a **Clock Pin** to use as the generated clock source. To display a list of available generated clock source pins, click the **Browse** button. The Select Generated Clock Source dialog box appears (as shown below).

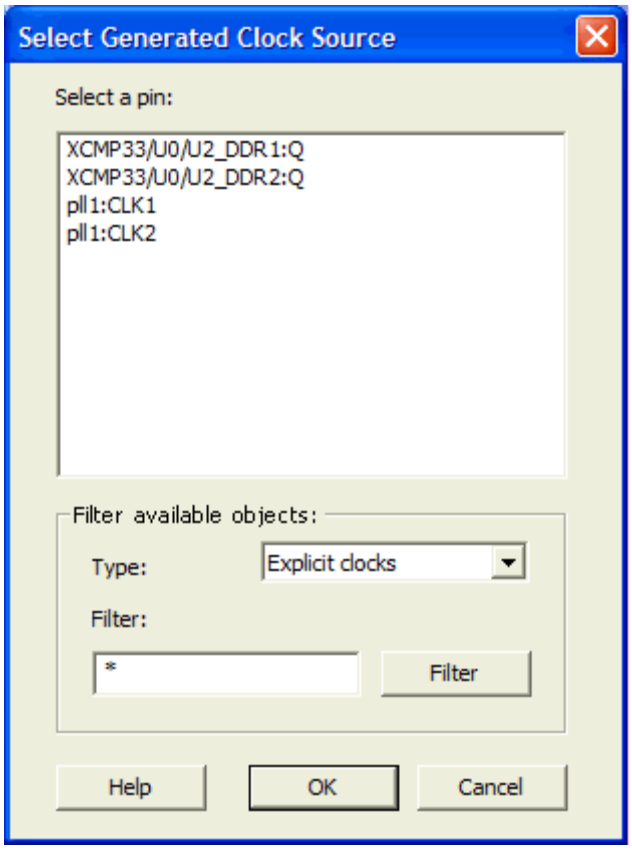


Figure 89 · Select Generated Clock Source Dialog Box

3. Modify the **Clock Name** if necessary.
4. Click **OK** to save these dialog box settings.
5. Specify a **Clock Reference**. To display a list of available clock reference pins, click the **Browse** button. The Select Generated Clock Reference dialog box appears.
5. Click **OK** to save this dialog box settings.
6. Specify the values to calculate the generated frequency: a multiplication factor and/or a division factor (both positive integers).
7. Specify the first edge of the generated waveform either same as or inverted with respect to the reference waveform.
8. Click **OK**. The new constraint appears in the Constraints List.

Tip: From the **File** menu, choose **Save**to save the newly created constraint in the database.

See Also

Design Constraint Guide: Clock

Design Constraint Guide: Create a Clock

Create Clock Constraint Dialog Box

Using Automatically Generated Clock Constraints

If your design uses a static PLL, SmartTime automatically generates the required frequency at the output of the PLL, provided you have supplied the input frequency. When you start SmartTime, a generated clock constraint appears in the Constraints List with the multiplication and division factor extracted from the PLL configuration. The **File** column specifies this constraint as **auto-generated** (as shown below).

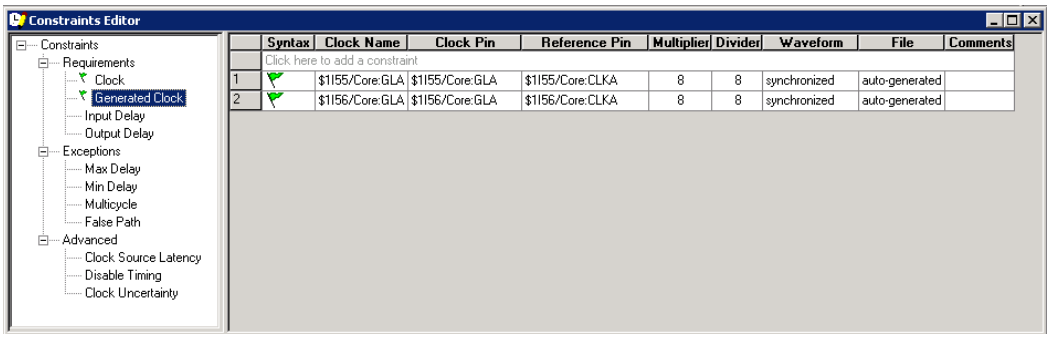


Figure 90 · Constraints Editor

Note: SmartTime does not automatically create a Generated Clock constraint if you have already set a constraint on the PLL output.

If you delete the automatically generated clock constraint, SmartTime does not regenerate it the next time you open the design. However, you can easily create it again by using the following steps:

1. Open the Create Generated Clock Constraint dialog box (as shown below).

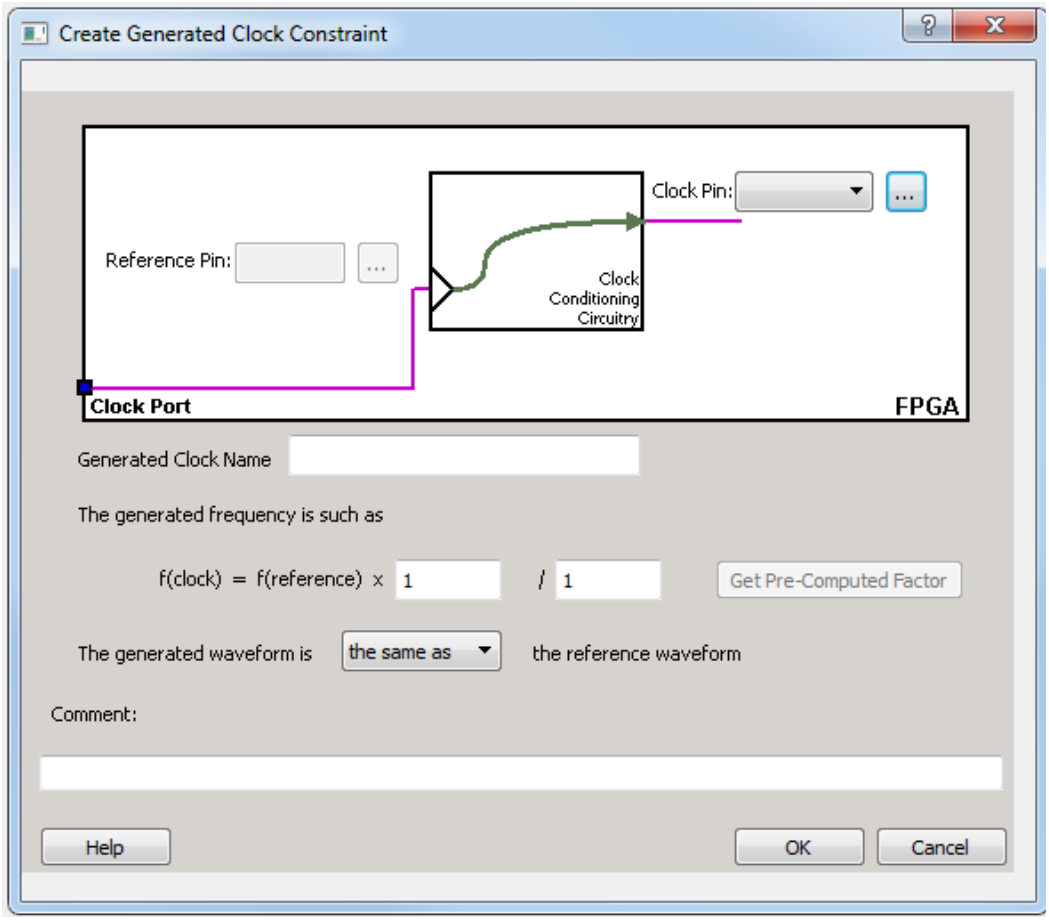


Figure 91 · Create Generated Clock Constraint

2. Select the PLL output as the **Clock Pin** source for the generated clock.
3. Select the PLL input clock as the **Clock Reference** for the generated clock.
4. Click **Get Pre-Computed Factors**. SmartTime retrieves the factor from the static PLL configuration.
5. Click **OK**.

Specifying an Input Delay Constraint

Use the input delay constraint to define the arrival time of an input relative to a clock.

To specify an input timing delay constraint:

1. Add the constraint in the editable constraints grid or open the **Set Input Delay Constraint** dialog box using one of the following methods:
   - From the SmartTime **Actions** menu, choose **Constraints > Input Delay**.
   - Click the  icon.
   - Right-click the **Input Delay** in the Constraint Browser.
   - Double-click any field in the Input Delay Constraints grid.

   The **Set Input Delay Constraint** dialog box appears (as shown below).

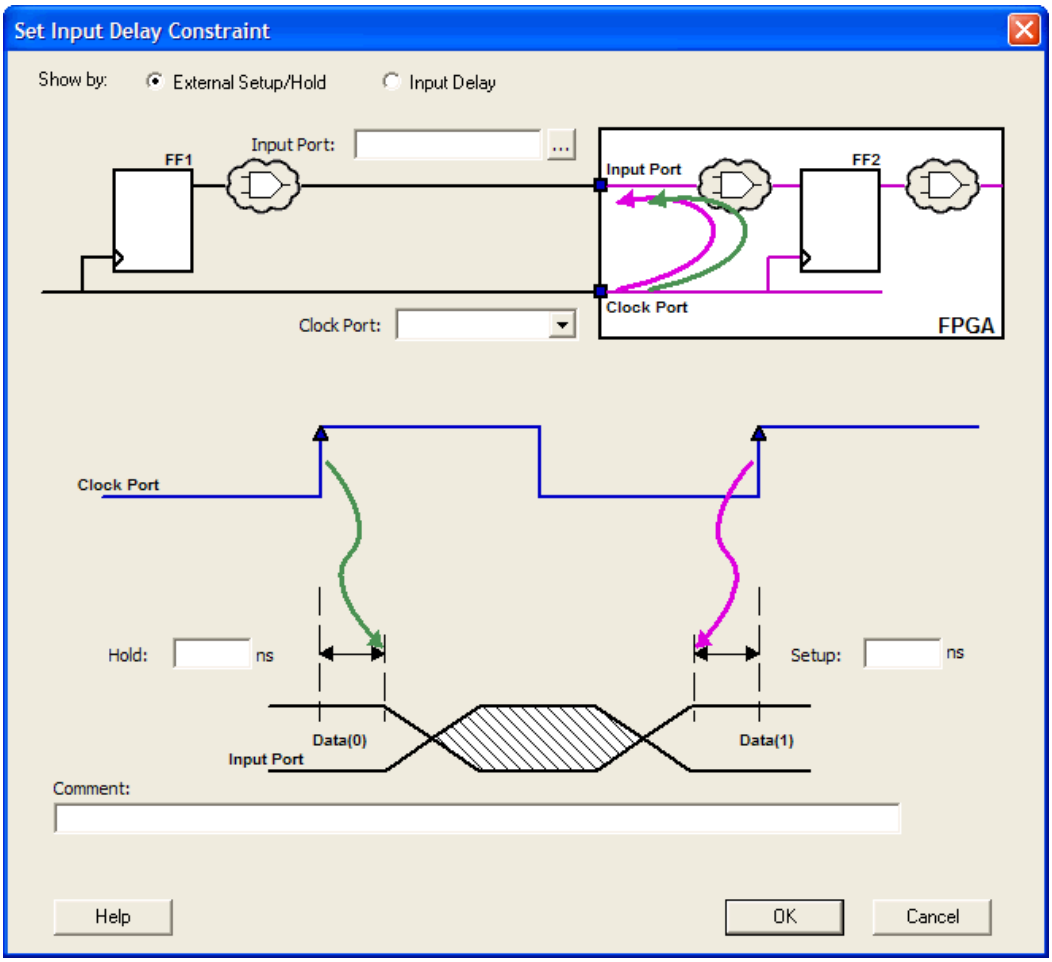


Figure 92 · Set Input Delay Dialog Box

2. Select either **External Setup/Hold** or **Input Delay**.

- **External Setup/Hold** enables you to enter an input delay constraint by specifying the timing budget inside the FPGA using the external setup and hold time. This is the default selection.

  Note: The external hold information is currently used for analysis only and not by the optimization tools. For the basic timing analysis flow of a simple design, select External Setup/Hold.

- **Input Delay** enables you to enter an input delay constraint by specifying the timing budget outside the FPGA. You can enter the Maximum Delay, the Minimum Delay, or both.

  Note: The Minimum Delay is currently used for analysis only and not by the optimization tools.

  When you change values in one view, SmartTime automatically updates the other view.

3. Specify the **Input Port** or click the **Browse** button to display the **Select Ports for Input Delay** dialog box.

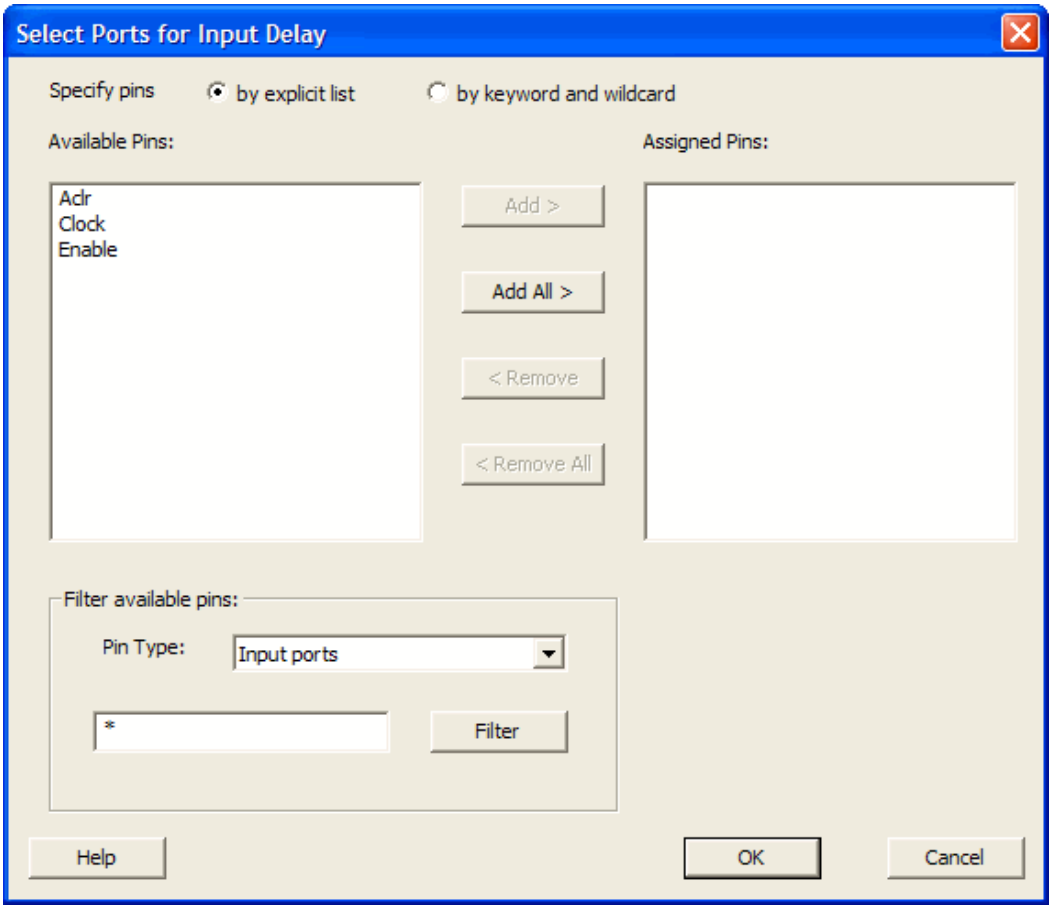


Figure 93 · Select Ports for Input Delay Dialog Box

3. Select the name of the input pin(s) from the **Available Pins** list. Choose the **Pin Type** from the drop-down list. You can use the filter to narrow the pin list. You can select multiple ports in this window.
4. Click **Add** or **Add All** to move the input pin(s) from the **Available Pins** list to the **Assigned Pins** list.
5. Click **OK**.

The Set Input Delay Constraint dialog box displays the updated Input Port information.

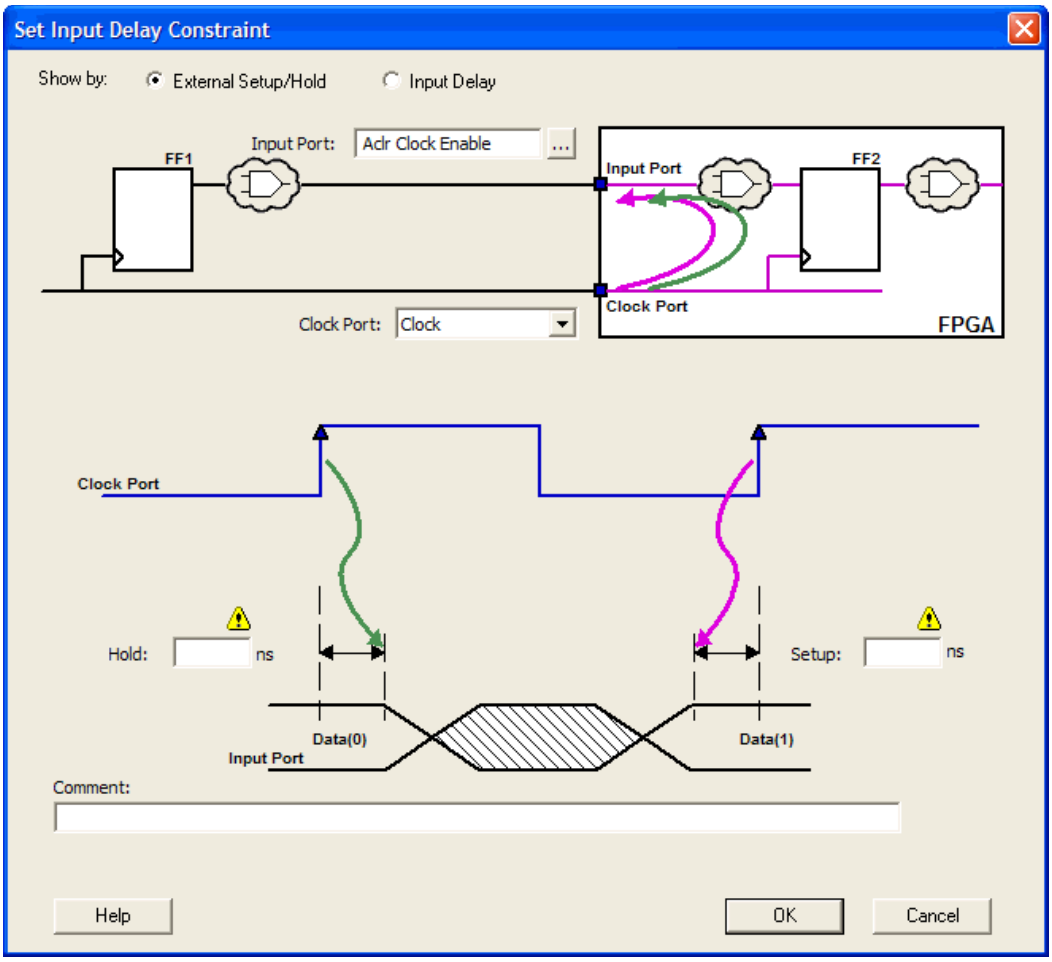


Figure 94 · Updated Set Input Delay Constraint Dialog Box

4. Select a clock from the **Clock Port** drop-down list.
5. If you selected **Show by: External Setup/Hold**, specifythe External Setup.
   If you selected **Show by: Input Delay**, specify the Maximum Delay value.
6. If you selected **Show by External Setup/Hold**, specify the External Hold. If you selected **Show by: Input Delay**, specify the Minimum Delay value.
7. Click **OK**.

SmartTime adds this constraint to the Constraints List in the SmartTime Constraints Editor.

See Also

Set Input Delay Constraint dialog box

Select Source or Destination Pins for Constraint dialog box

Specifying an Output Delay Constraint

Use the output delay constraints to define the output delay of an output relative to a clock.

To specify an output delay constraint:

1. Add the constraint in the editable constraints grid or open the **Set Output Delay Constraint** dialog box using one of the following methods:
   - From the SmartTime **Actions** menu, choose **Constraints > Output Delay**.
   - Click the  icon.

- Right-click the **Output Delay** in the Constraint Browser.
- Double-click any field in the Output Delay Constraints grid.

The **Set Output Delay Constraint** dialog box appears.

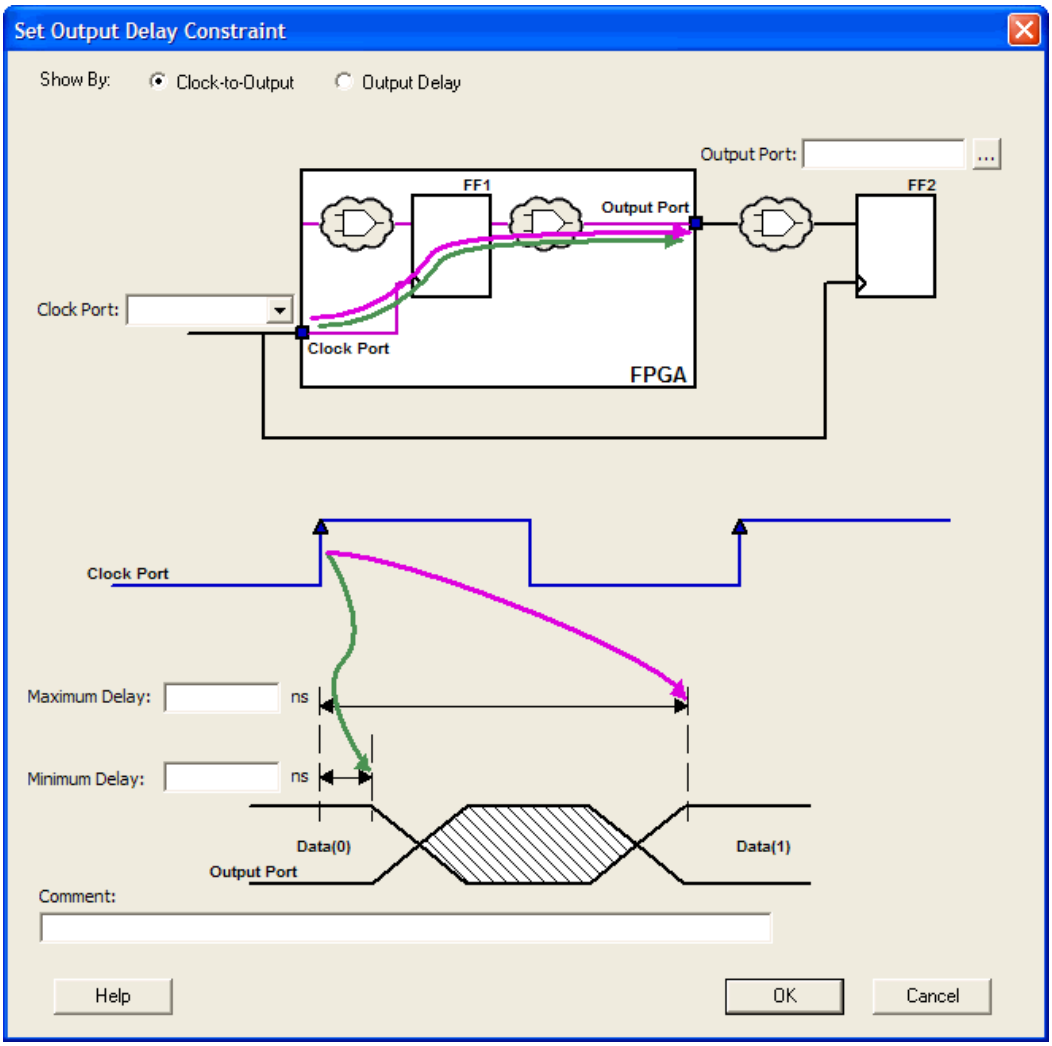


Figure 95 · Set Output Delay Constraint Dialog Box

Specify either **Clock-to-Output** or **Output Delay**.

- **Clock-to-Output** enables you to enter an output delay constraint by specifying the timing budget inside the FPGA. This is the default selection.

  Note: The Minimum Delay value is currently used for analysis only and not by the optimization tool.

- **Output Delay** enables you to enter an output delay constraint by specifying the timing budget outside the FPGA. You can enter either the Maximum Delay, the Minimum Delay, or both.

  Note: The Minimum Delay is currently used for analysis only and not by the optimization tools.

When you change values in one view, SmartTime automatically updates the values in the other view.

3. Enter the name of the **Output Port** or click the Browse button to display the **Select Ports for Output Delay** dialog box.

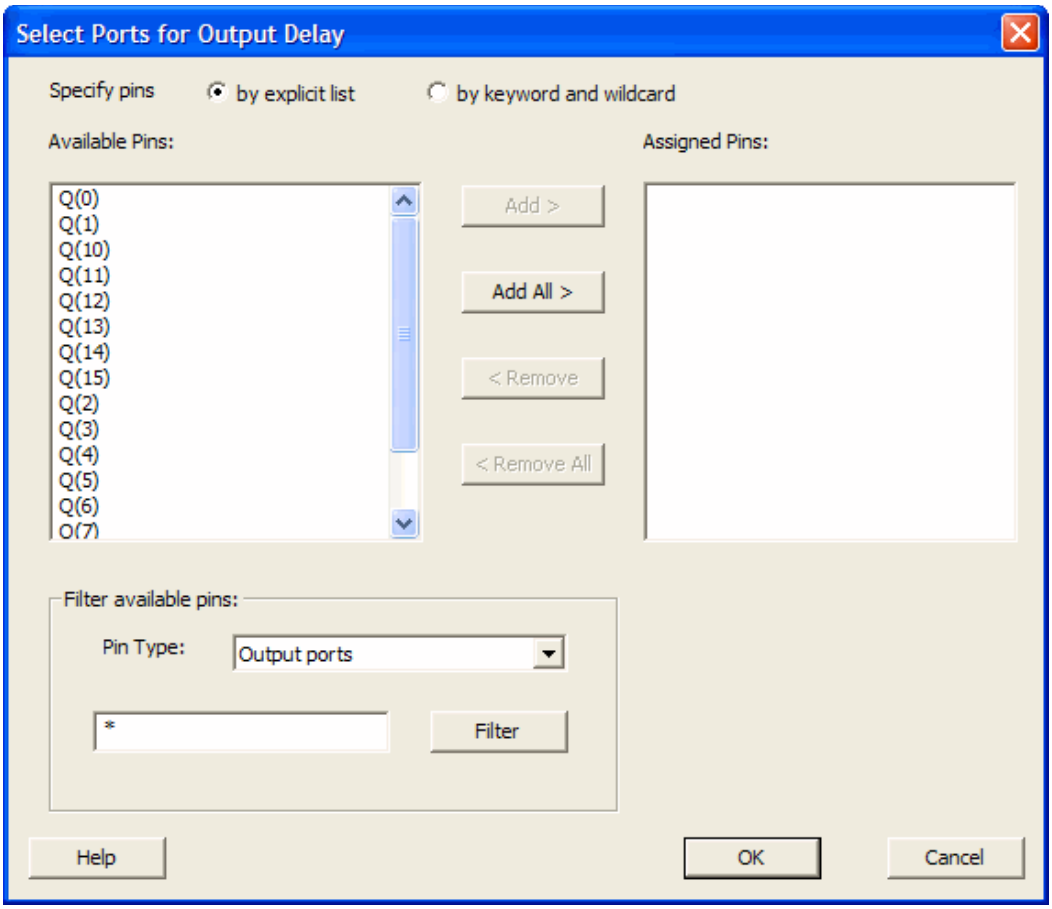


Figure 96 · Select Ports for Output Delay Dialog Box

4. Select the output pin(s) from the **Available Pin** list. Choose the **Pin Type** from the drop-down list. You can use the filter to narrow the pin list. You can select multiple ports in this dialog box.
5. Click **Add** or **Add All** to move the output pin(s) from the **Available Pins** list to the **Assigned Pins** list.
6. Click **OK**. The **Set Output Delay Constraint** dialog box displays the updated representation of the Output Port graphic.
7. Select a clock port from the **Clock Port** drop-down list.
8. Enter the **Maximum Delay** value.
9. Enter the **Minimum Delay** value.
10. Click **OK**. SmartTime adds this constraint to the Constraints List in the Constraints Editor.

See Also

Set Output Delay Constraint dialog box

Select Source or Destination Pins for Constraint dialog box

SmartTime Timing Analyzer

Components of the SmartTime Timing Analyzer

Use the SmartTime Timing Analyzer to visualize and identify timing issues in your design for the selected scenario. In this view, you can evaluate how far you are from meeting your timing requirements, create custom sets to track, set timing exceptions to obtain timing closure, and cross-probe paths with other tools.

The timing analysis view includes:

- Domain Browser: Enables you to perform your timing analysis on a per domain basis.
- Path List: Displays paths in a specific set in a given domain sorted by slack.
- Path Details: Displays detailed timing analysis of a selected path in the paths list.
- Analysis View Filter: Enables you to filter the content of the paths list.
- Path Slack Histogram: When a set is selected in the Domain Browser, the Path Slack Histogram displays a distribution of the path slacks for that set. Selecting one or multiple bars in the Path Slack Histogram filters the paths displayed in the Path List.
  You can copy, change the resolution and the number of bars of the chart from the right-click menu.

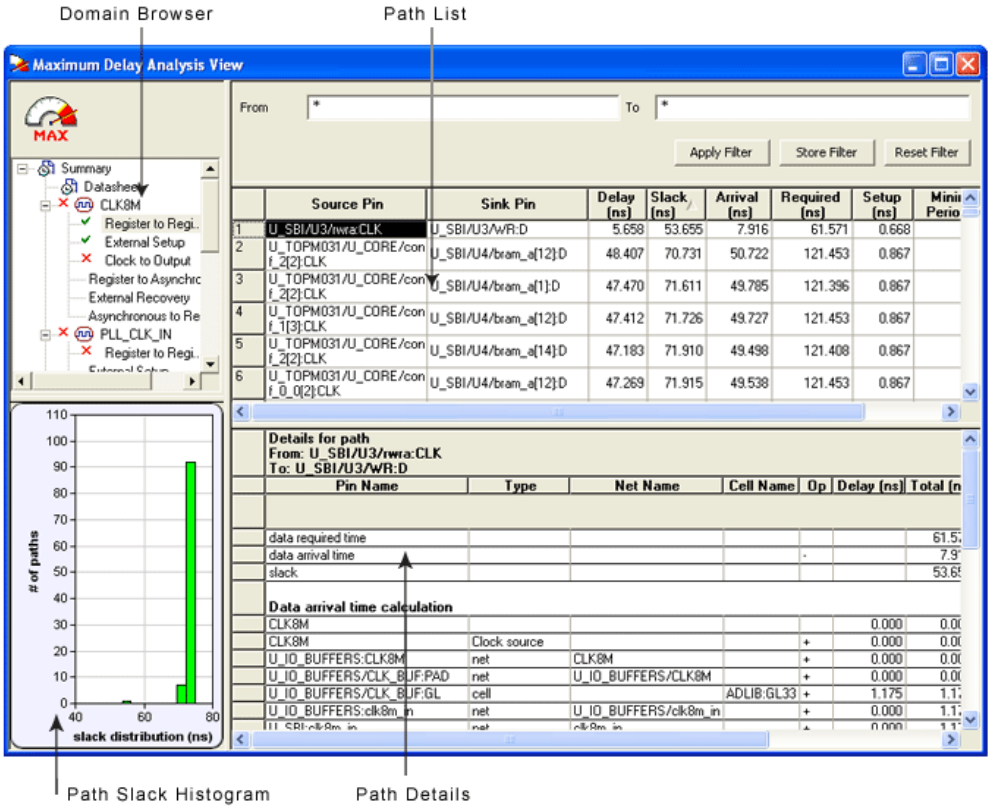


Figure 97 · SmartTime Timing Analyzer Components

See Also

SmartTime Constraint Scenario

Analyzing Your Design

The timing engine uses the following priorities when analyzing paths and calculating slack:

1. False path
2. Max/Min delay
3. Multi-cycle path
4. Clock

When multiple constraints of different priorities are set on the same timing path, the constraint with the higher priority overrides the constraint with a lower priority. The False Path constraint has the highest priority on a path and overrides all other constraints on the same path.

If multiple constraints of the same priority apply to a path, the timing engine uses the tightest constraint. You can perform two types of timing analysis: Maximum Delay Analysis and Minimum Delay Analysis.

To perform the basic timing analysis:

1. Open the Timing Analysis View using one of the following methods:
   - In the Design Flow window, click **Verify Timing > Open Interactively** to display the SmartTime Maximum Delay Analysis View.
   - From the SmartTime **Tools** menu, choose **Max Delay Analysis** or **Min Delay Analysis.**
   - Click the  icon for Maximum Delay Analysis or the  icon for Minimum Delay Analysis from the SmartTime window.

   Note: When you open SmartTime from Libero (Verify Timing > Open Interactively) , the Maximum Delay Analysis window is displayed by default.

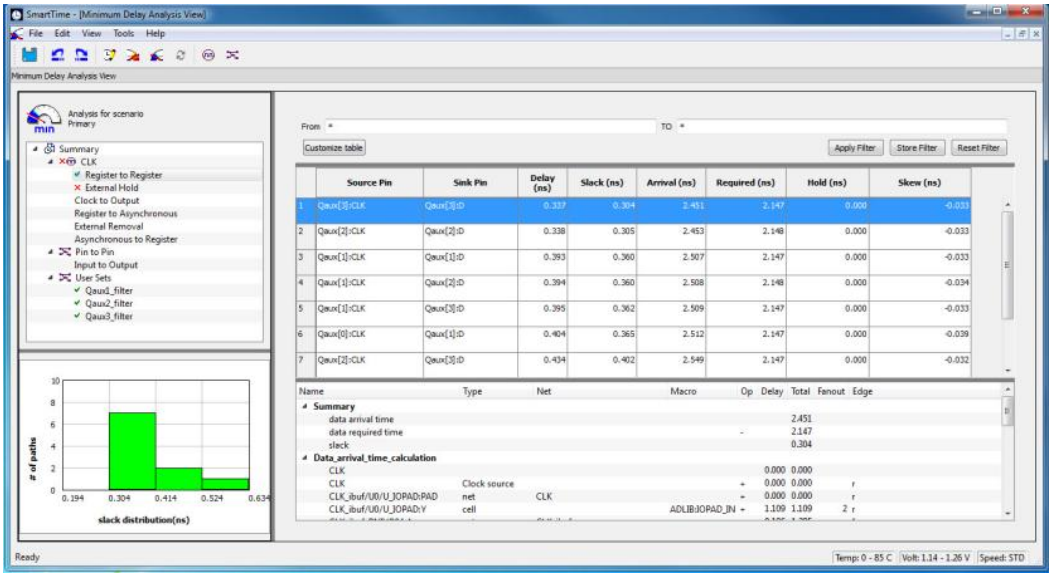


Figure 98 · Minimum Delay Analysis View

2. In the Domain Browser, select the clock domain. Clock domains with a ✔ indicate that the timing requirements in these domains were met. Clock domains with an x indicate that there are violations within these domains. The Paths List displays the timing paths sorted by slack. The path with the lowest slack (biggest violation) is at the top of the list.
3. Select the path to view. The Path Details below the Paths List displays detailed information on how the slack was computed by detailing the arrival time and required time calculation. When a path is violated, the slack is negative and is displayed in red color.
4. Double-click the path to display a separate view that includes the path details and schematic view.

   Note: In cases where the minimum pulse width of one element on the critical path limits the maximum frequency for the clock, SmartTime displays an icon for the clock name in the Summary List. Click the icon to display the name of the pin that limits the clock frequency.

5. Repeat the above steps as required.

Performing a Bottleneck Analysis

To perform a bottleneck analysis

1. From SmartTime's Max/Min Delay Analysis View, select **Tools > Bottleneck Analysis**. The **Timing Bottleneck Analysis Options** dialog box appears.
2. Select the options you wish to display for bottleneck information and click **OK**.

   The **Bottleneck Analysis View** appears in a separate window (see image below).

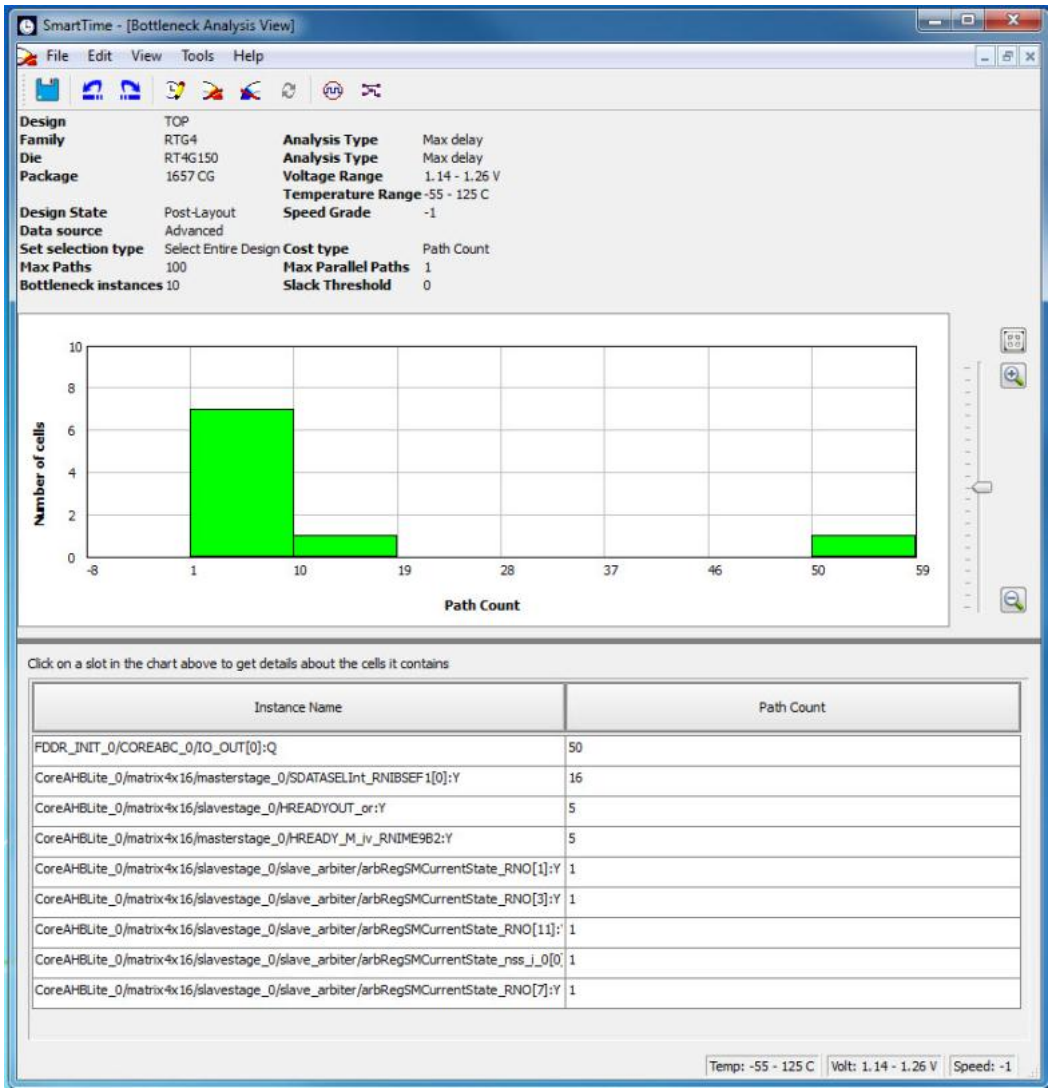


Figure 99 · Bottleneck Analysis View

A bottleneck is a point in the design that contributes to multiple timing violations. The Bottleneck Analysis View contains two sections:

- Device Description
- Bottleneck Description

Device Description

The device section contains general information about the design and the parameters that define the bottleneck computation:

- Design name
- Family
- Die
- Package
- Design state
- Data source
- Set selection type
- Max paths
- Bottleneck instances
- Analysis type
- Analysis max case
- Voltage
- Temperature
- Speed grade
- Cost type
- Max parallel paths
- Slack threshold

Bottleneck Description

This section displays a graphic representation of the bottleneck analysis and lists the core of the bottleneck information for the bar selected in the chart above. If no bar is selected, the grid lists all bottleneck information.

Click the controls on the right to zoom in or out the contents in the chart.

Right-click the chart to export the chart or to copy the chart to the clipboard.

The list is divided into two columns:

- Instance name: refers to the output pin name of the instance.
- Bottleneck cost: displays the pin's cost given the chosen cost type. Pin names are listed in decreasing order of their cost type.

See Also

Timing Bottleneck Analysis Options dialog box (SmartTime)

Managing Clock Domains

In SmartTime, timing paths are organized by clock domains. By default, SmartTime displays domains with explicit clocks. Each clock domain includes at least three path sets:

- Register to Register
- External Setup (in the Maximum Analysis View) or External Hold (in the Minimum Analysis View)
- Clock to Out

You must select a path set to display a list of paths in that specific set.

To manage the clock domains:

1. Right-click anywhere in the Domain Browser, and choose **Manage Clock Domains**. The Manage Clock Domains dialog box appears (as shown below).

Tip: You can click the  icon in the SmartTime window bar to display the **Manage Clock Domains** dialog box.

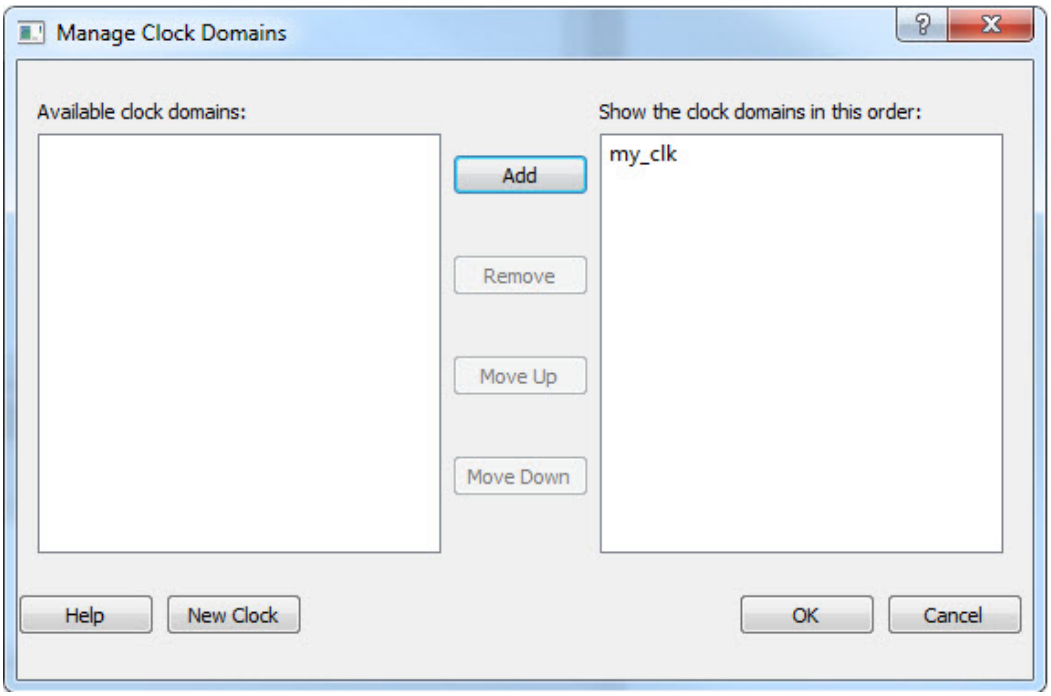


Figure 100 · Manage Clock Domains Dialog Box

2. To add a new domain, select a clock domain from the **Available clock domains** list, and click either **Add** or **New Clock** to add a non-explicit clock domain.
3. To remove a displayed domain, select a clock domain from the **Show the clock domain in this order** list, and click **Remove**.
4. To change the display order in the Domain Browser, select a clock domain from the **Show the clock domainin this order** list, and then use the **Move Up** or **Move Down** to change the order in the list.
5. Click **OK**. SmartTime updates the Domain Browser based on your specifications. If you have added a new clock domain, then it will include at least the three path sets as mentioned above.

See Also

Manage Clock Domains Dialog Box

Managing Path Sets

You can create and manage custom path sets for timing analysis and tracking purposes. Path sets are displayed under the **Custom Path Sets** at the bottom of the Domain Browser.

To add a new path set:

1. Right-click anywhere in the Domain Browser, and choose **Add Set**. The Add Path Analysis Set Dialog Box dialog box appears (as shown below).

Tip: You can click the  icon in the SmartTime window bar to display the **Add Path Analysis Set** dialog box.

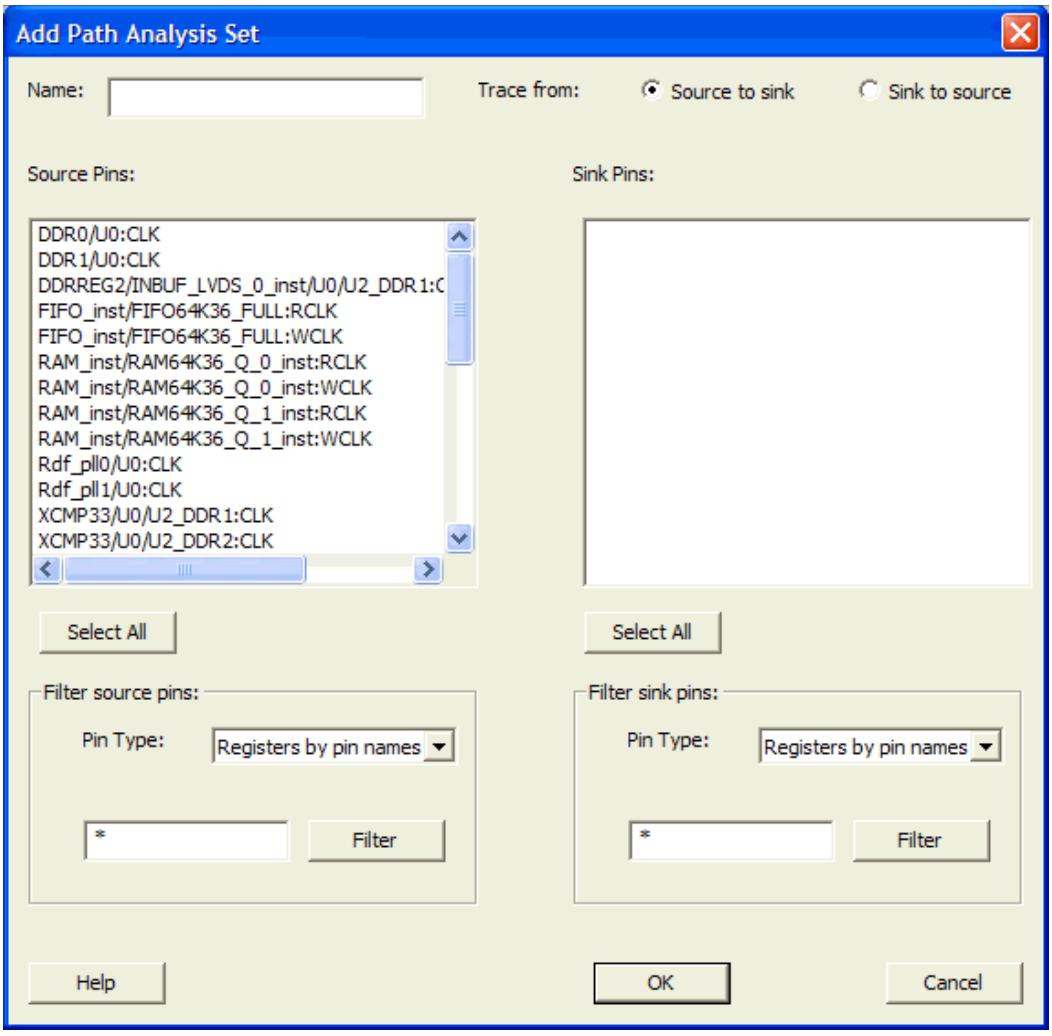


Figure 101 · Add Path Analysis Set Dialog Box

2. Enter a name for the path set.
3. Select the source and sink pins. You can use the filters to control the type of pins displayed.
4. Click **OK**. The new path set appears under **Custom Path Sets** in the Domain Browser (as shown below).

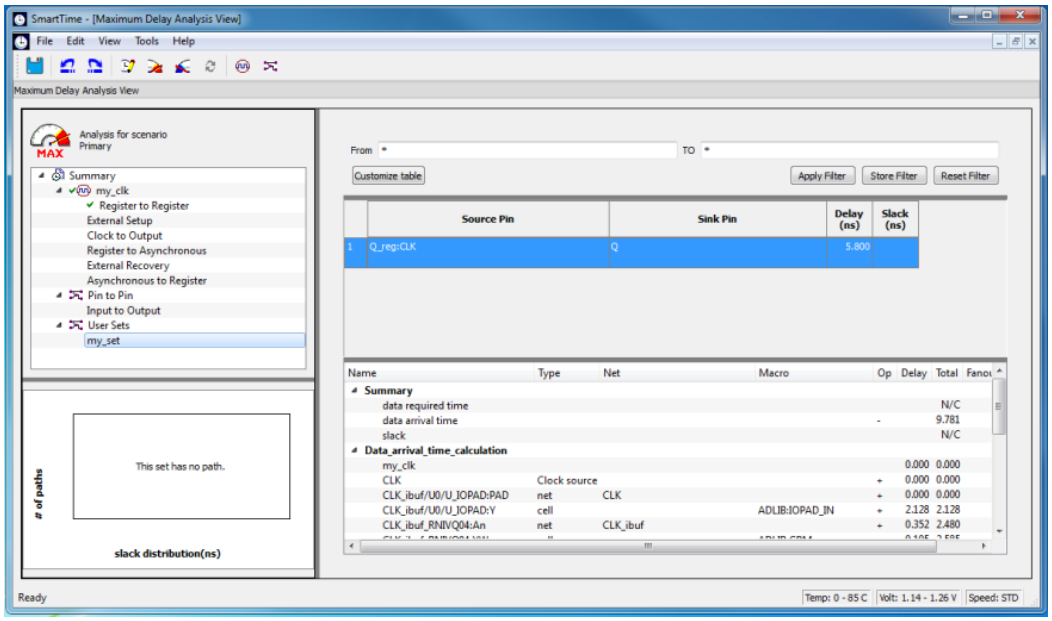


Figure 102 · Updated Domain Browser with User Sets

To remove an existing path set:

1. Select the path set from the **User Sets** in the Domain Browser.
2. Right-click the set to delete, and then choose **Delete Set** from the right-click menu.

To rename an existing path set:

1. Select the path set from **User Set** in the Domain Browser.
2. Right-click the set to rename, and then choose **Rename Set** from the right-click menu.
3. Edit the name directly in the Domain Browser.

See Also

Add Path Analysis Set Dialog Box

Using Filters

Exporting Files

Displaying Path List Timing Information

The Path List in the Timing Analysis View displays the timing information required to verify the timing requirements and identify violating paths. The Path List is organized in a grid where each row represents a timing path with the corresponding timing information displayed in columns. Timing information is customizable; you can add or remove columns for each type of set.

By default, each type of set displays a subset of columns as follows:

- Register to Register: Source Pin, Sink Pin, Delay, Slack, Arrival, Required, Setup, Minimum Period, and Skew.
- External Setup: Source Pin, Sink Pin, Delay, Slack, Arrival, Required, Setup, and External Setup.
- Clock to Out: Source Pin, Sink Pin, Delay, Slack, Arrival, Required, and Clock to Out.
- Input to Output: Source Pin, Sink Pin, Delay, and Slack.
- Custom Path Sets: Source Pin, Sink Pin, Delay, and Slack.

You can add the following columns for each type of set:

- Register to Register: Clock, Source Clock Edge, Destination Clock Edge, Logic Stage Count, Max Fanout, Clock Constraint, Maximum Delay Constraint, and Multicycle Constraint.

- External Setup: Clock, Destination Clock Edge, Logic Stage Count, Max Fanout, Clock Constraint, Input Delay Constraint, Required External Setup, Maximum Delay Constraint, and Multicycle Constraint.
- Clock to Out: Clock, Source Clock Edge, Logic Stage Count, Max Fanout, Clock Constraint, Output Delay Constraint, Required Maximum Clock to Out, Maximum Delay Constraint, and Multicycle Constraint.
- Input to Output: Arrival, Required, Setup, Hold, Logic Stage Count, and Max Fanout.
- Custom Path Sets.

To customize the set of timing information in the Path List:

1. Select the set to customize.
2. Select the whole Paths List by clicking in the upper-left corner.
3. Right-click anywhere on the column headings, and then choose **Customize table** from the right-click menu. The Customize Analysis View Dialog Box dialog box appears (as shown below).

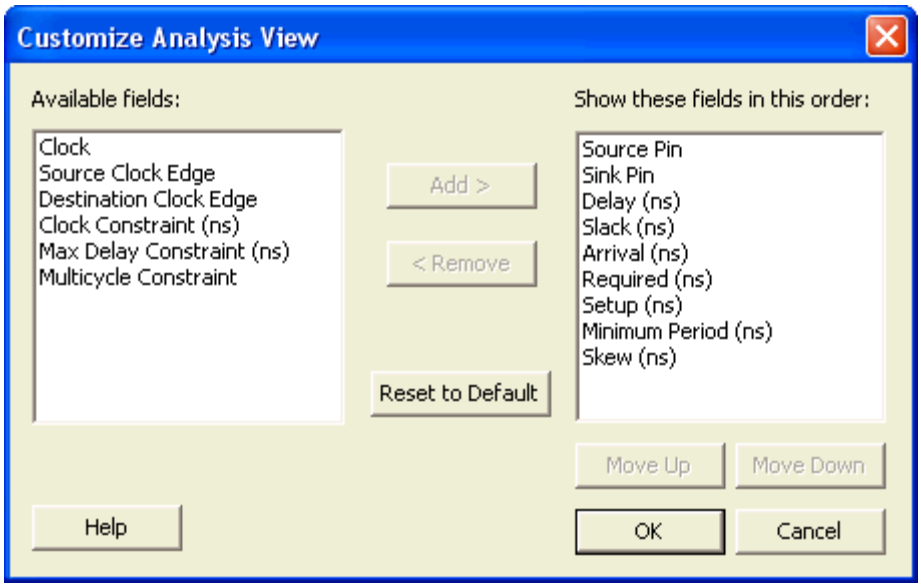


Figure 103 · Customize Analysis View Dialog Box

4. To add one or more columns, select the fields to add from the **Available fields** list, and click **Add**.
5. To remove one or more columns, select the fields to remove from the **Show these fields in this order** list, and click **Remove**.
6. Click **OK** to add or remove the selected columns. SmartTime updates the Timing Analysis View.

See Also

Customize Analysis View

Displaying Expanded Path Timing Information

SmartTime displays the list of paths and the path details for all parallel paths.

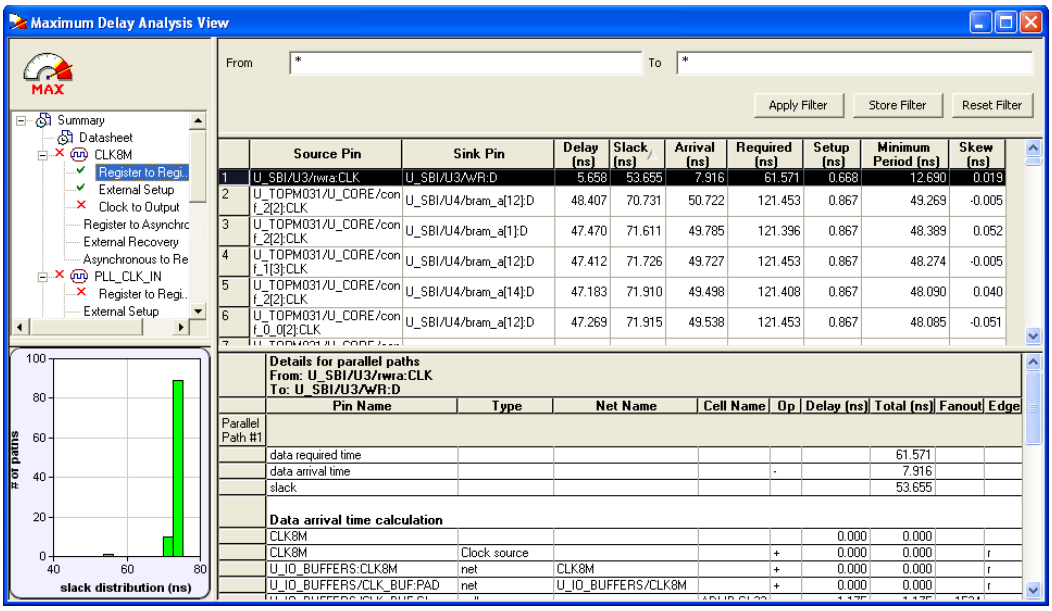


Figure 104 · Expanded Path View

The Path List displays all parallel paths in your design. The Path Details grid displays the path details for all parallel paths.

To display the Expanded Path View:

From the Path List: double-click the path, or right-click a path and select **expand selected paths**.

From the Expanded Path View: double-click the path, or right-click the path and select **expand path**.

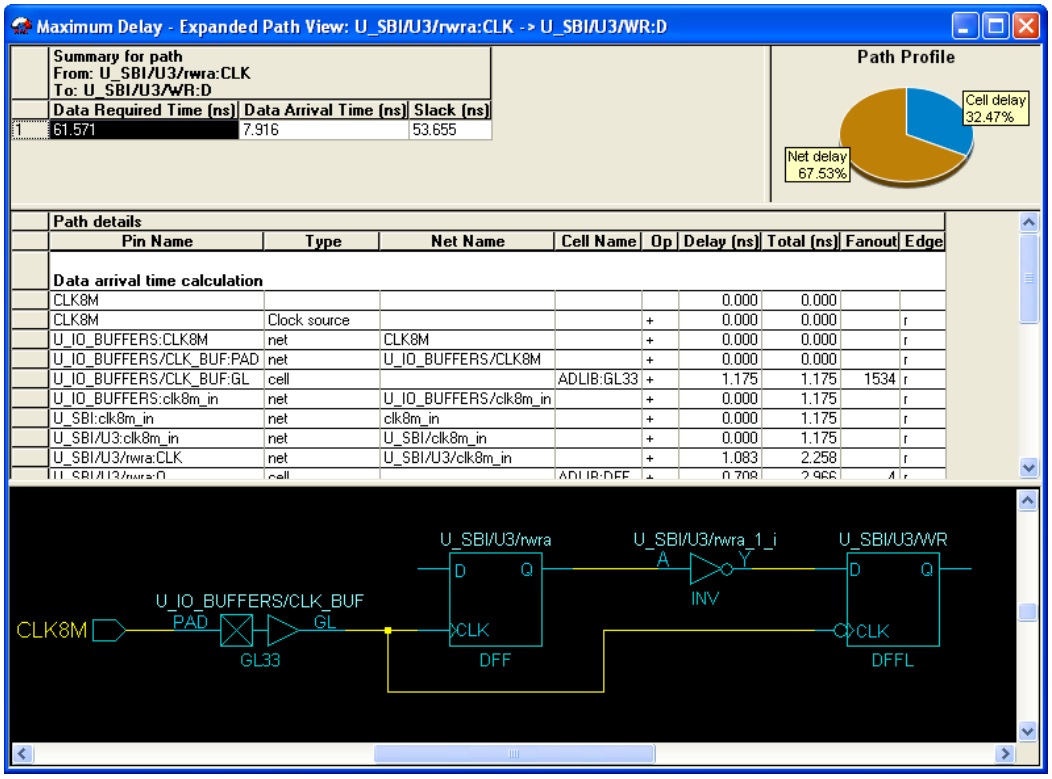


Figure 105 · Expanded Path View

The Expanded Path Summary provides a summary of all parallel paths for the selected path. The Path Profile chart displays the percentage of time taken by cells and nets for the selected path. If no parallel path

is selected in this view, the Path Profile shows the percentage for all paths. By default, SmartTime only shows one path for each Expanded Path. You can change this default in the SmartTime Options dialog box.

The Expanded Path View also includes a schematic of the path and a path profile chart for the paths selected in the Expanded Path Summary.

Using Filters

You can use filters in SmartTime to limit the Path List content (that is, create a filtered list on the source and sink pin names). The filtering options appear on the top of the Timing Analysis View. You can save these filters one level below the set under which it has been created.

To use the filter:

1. Select a set in the Domain Browser to display a given number of paths, depending on your SmartTime Options settings (100 paths by default).
2. Enter the filter criteria in both the **From** and **To** fields and click **Apply Filter**. This limits the display to the paths that match your filter criteria.

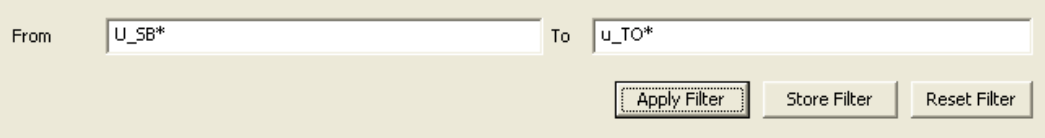


Figure 106 · Maximum Delay Analysis View

3. Click **Store Filter** to save your filter criteria with a special name. The **Create Filter Set** dialog box appears (as shown below).

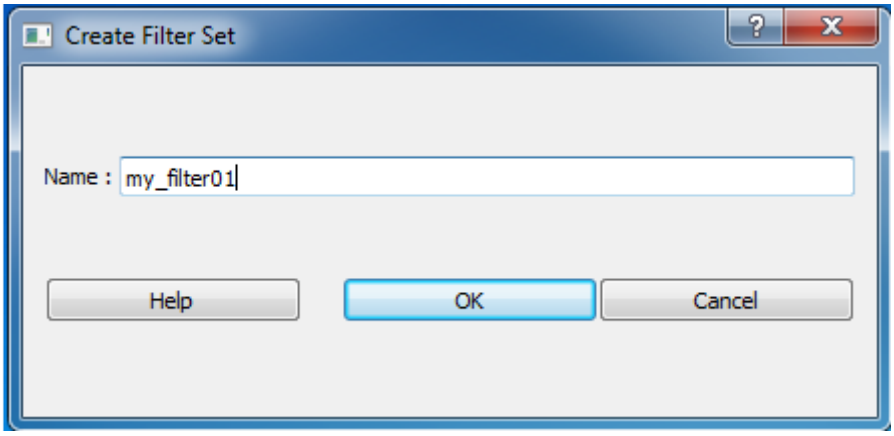


Figure 107 · Create Filter Set Dialog Box

4. Enter a name for the filter, such as myfilter01, and click **OK**. Your new filter name appears below the set under which it was created.

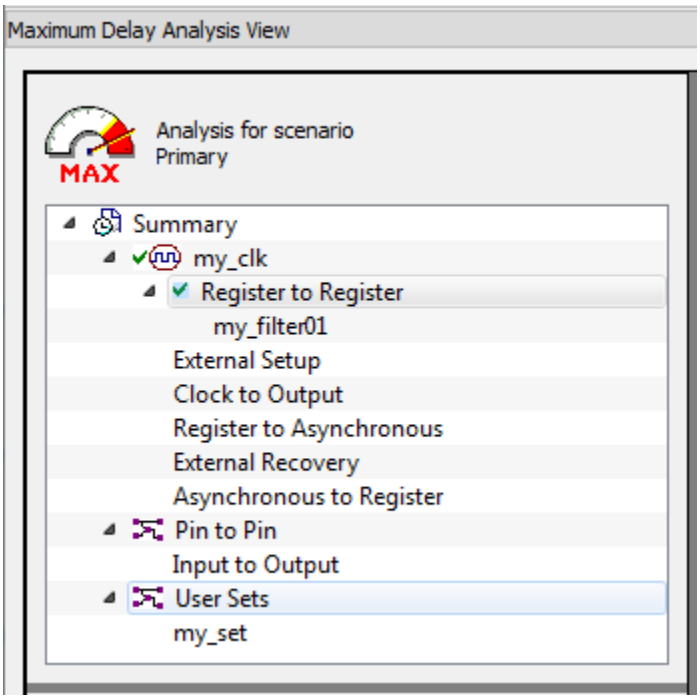


Figure 108 · my_filter01

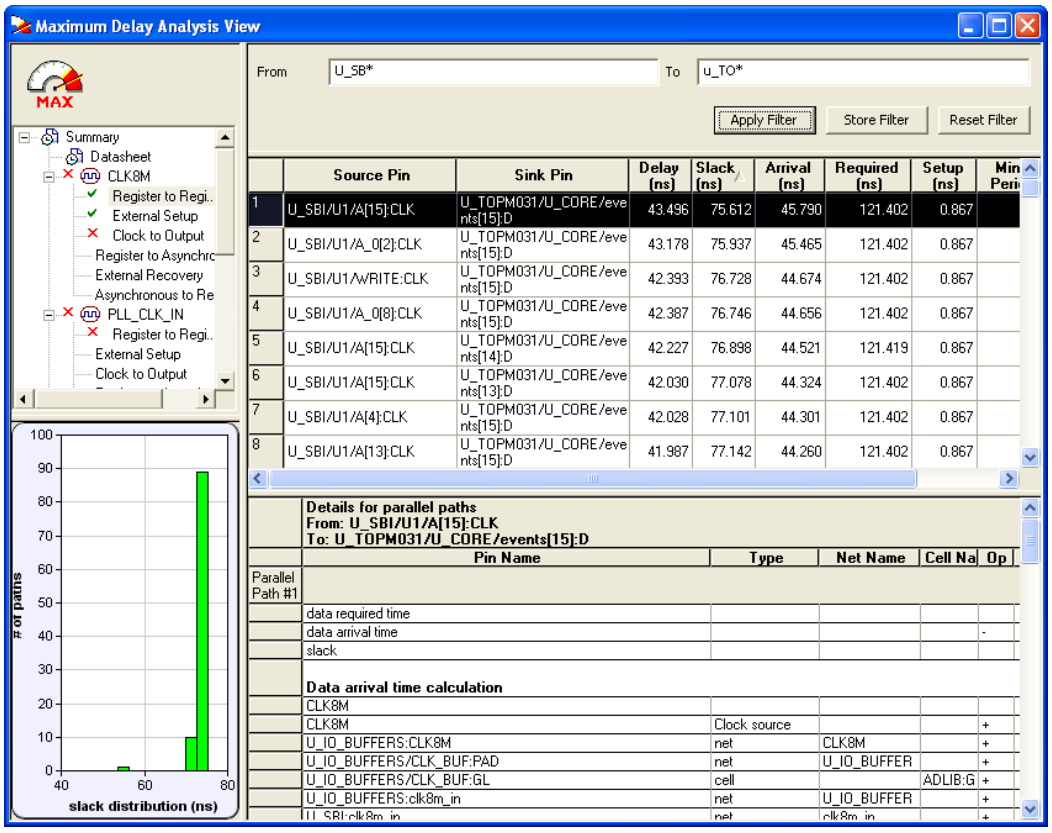


Figure 109 · Updated Maximum Delay Analysis View

Repeat the above steps and cascade as many sets as you need using the filtering mechanism.

To remove a set created with filters:

1. Select the set that uses filters.
2. Right-click the set, and choose **Delete Set** from the shortcut menu.

To rename a set created with filters:

1. Select the set that uses filters.
2. Right-click the set, and choose **Rename Set** from the shortcut menu.
3. Edit the name directly in the Domain Browser.

To edit a specific filter in the set:

1. Select the filter to edit.
2. Right-click the filter, and choose **Edit Set** from the shortcut menu.

See Also

SmartTime Options

Store Filter as Analysis Set

Edit Set dialog box

Exporting Files

Advanced Timing Analysis

Understanding Inter-Clock Domain Analysis

When functional paths exist across two clock domains (the register launching the data and the one capturing it are clocked by two different clock sources), you must provide accurate specification of both clocks to allow a valid inter-clock domain timing check. This is important especially when the clocks are specified with different waveforms and frequencies.

When you specify multiple clocks in your design, the first step is to consider whether the inter-clock domain paths are false or functional. If these paths are functional, then you must perform setup and hold checks between the clock domains in SmartTime. Unless specified otherwise, SmartTime considers the inter-clock domain as false, and therefore does not perform setup or hold checks between the clock domains.

If you have several clock domains that are subset of a single clock (such as if you want to measure clock tree delay from an input clock to a generated clock), you must configure Generated Clock Constraints for each of the clock domains in order for SmartTime to do execute the calculation and show timing for each of the inter-clock-domain paths.

Once you include the inter-clock domains for timing analysis, SmartTime analyzes for each inter-clock domain the relationship between all the active clock edges over a common period equal to the least common multiple of the two clock periods. The new common period represents a full repeating cycle (or pattern) of the two clock waveforms (as shown below).

For setup check, SmartTime considers the tightest relation launch-capture to ensure that the data arrives before the capture edge. The hold check verifies that a setup relationship is not overwritten by a following data launch.

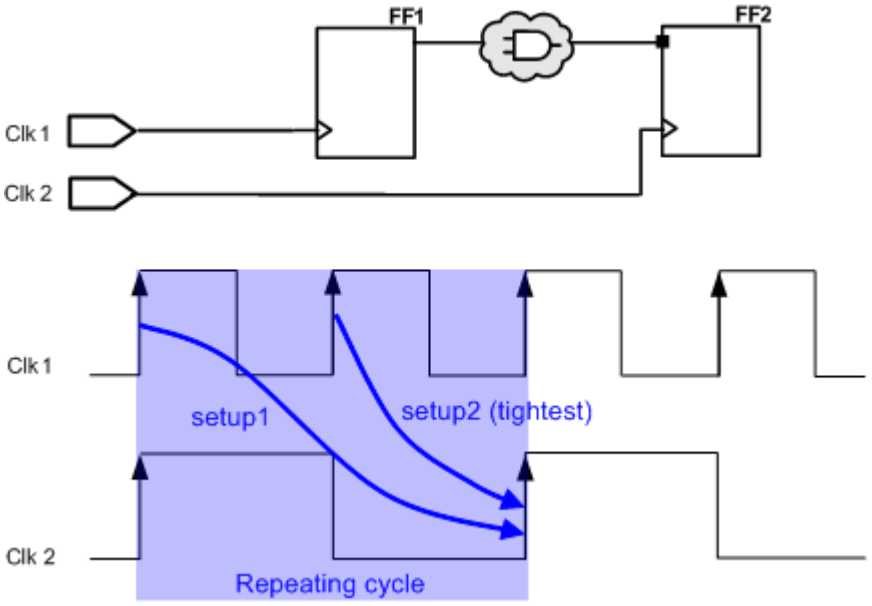


Figure 110 · Example Showing Inter-Clock Domains

See Also

Activating inter-clock domain analysis

Deactivating a specific inter-clock domain

Displaying inter-clock domain paths

Activating Inter-Clock Domain Analysis

To activate the inter-clock domain checking:

1. In SmartTime, from the **Tools** menu choose **Options**. The SmartTime Options Dialog Box dialog box appears (as shown below).
2. In the general category, check the **Include inter-clock domains in calculations for timing analysis**.

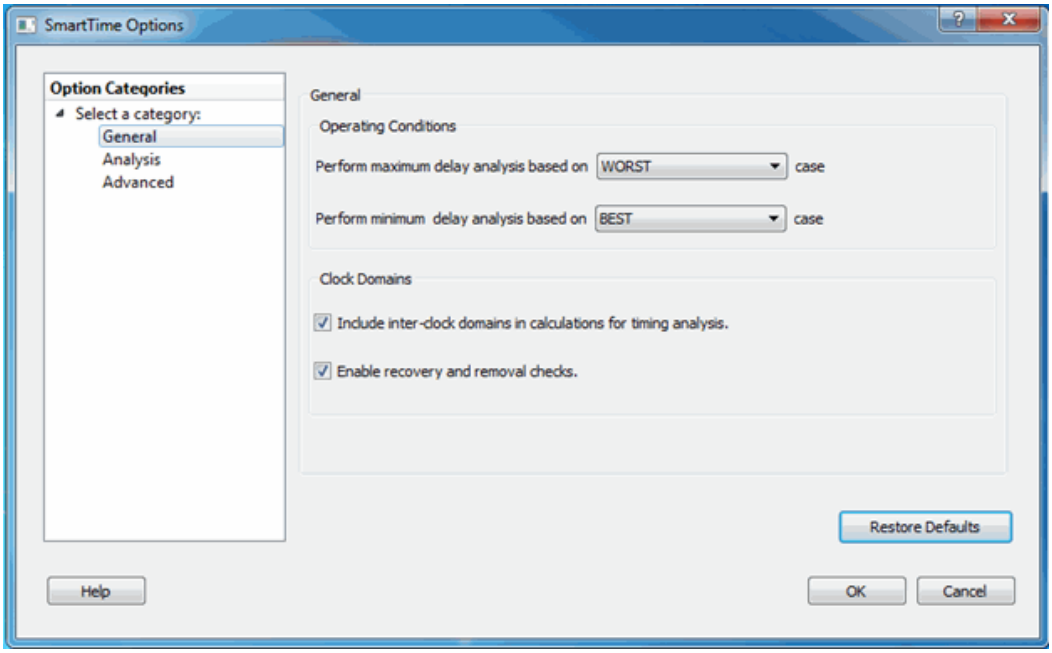


Figure 111 · SmartTime Options Dialog Box

3. Click **OK** to save the dialog box settings.

See Also

Inter-Clock Domain Analysis

Deactivating a Specific Inter-Clock Domain

Displaying Inter-Clock Domain Paths

Displaying Inter-Clock Domain Paths

Once you activate the inter-clock domain checking for a given clock domain CK1, SmartTime automatically detects all other domains CKn with paths ending at CK1. SmartTime creates inter-clock domain sets CKn to CK1 under the domain CK1. Each of these sets enables you to display the inter-clock domain paths between a given clock domain and CK1.

To display an inter-clock domain set:

1. Expand the receiving clock domain of the inter-clock domain in the Domain Browser to display its related sets. For the inter-clock domain CK1 to CK2, expand clock domain CK2.
2. Select the inter-clock domain that you want to see expanded from these sets. Once selected, all paths between the related two domains are displayed in Paths List in the same way as any register to register set.

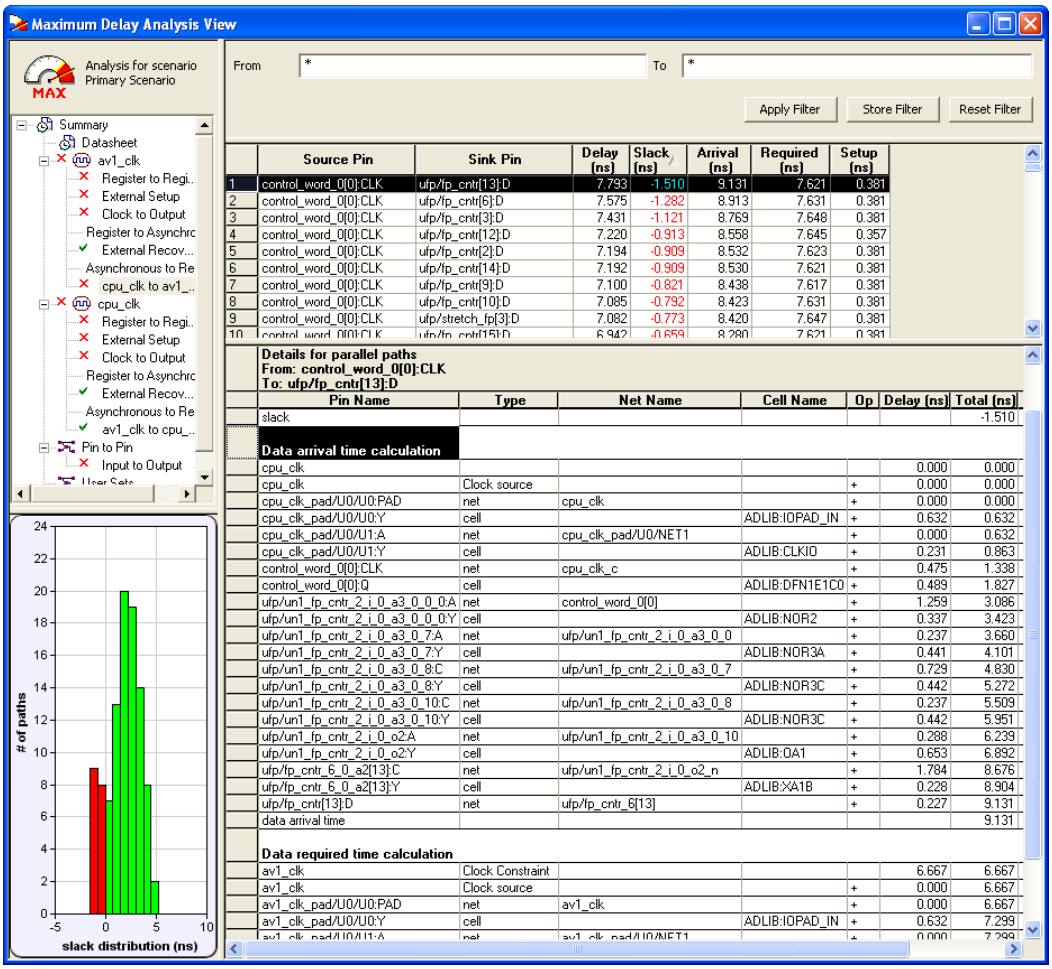


Figure 112 · Maximum Delay Analysis View

See Also

Inter-Clock Domain Analysis

Activating Inter-Clock Domain Analysis

Deactivating a Specific Inter-Clock Domain

Deactivating a Specific Inter-Clock Domain

To deactivate the inter-clock domain checking for the specific clock domains clk2->clk1, without disabling this option for the other clock domains:

1. From the **Tools** menu, choose **Constraints Editor** to open the Constraints Editor View.
2. In the Constraints Browser, double-click **False Path** under **Exceptions**. The Set False Path Constraint dialog box appears.
3. Click the **Browse** button to the right of the **From** text box. The **Select Source Pins for False Path Constraint** dialog box appears.
4. For **Specify pins**, select **by keyword and wildcard**.
5. For **Pin Type**, select **Registers by clock names** from the **Pin Type** drop-down list.
6. Type the inter-clock domain name, for example **Clk2**, in the filter box and click **Filter**.
7. Click **OK** to begin filtering the pins by your criteria. In this example, [get_clocks {Clk2}] appears in the **From** text box in the Set False Path Constraint dialog box.

8. Repeat steps 3 to 7 for the **To** option in the Set False Path Constraint dialog box, and type Clk2 in the filter box.
9. Click **OK** to validate the new false path and display it in the Paths List of the Constraints Editor.
10. Click the Recalculate All button in the toolbar.
11. Select the inter-clock domain set clk2 -> clk1 in the Domain Browser (as shown below).
12. Verify that the set does not contain any paths.

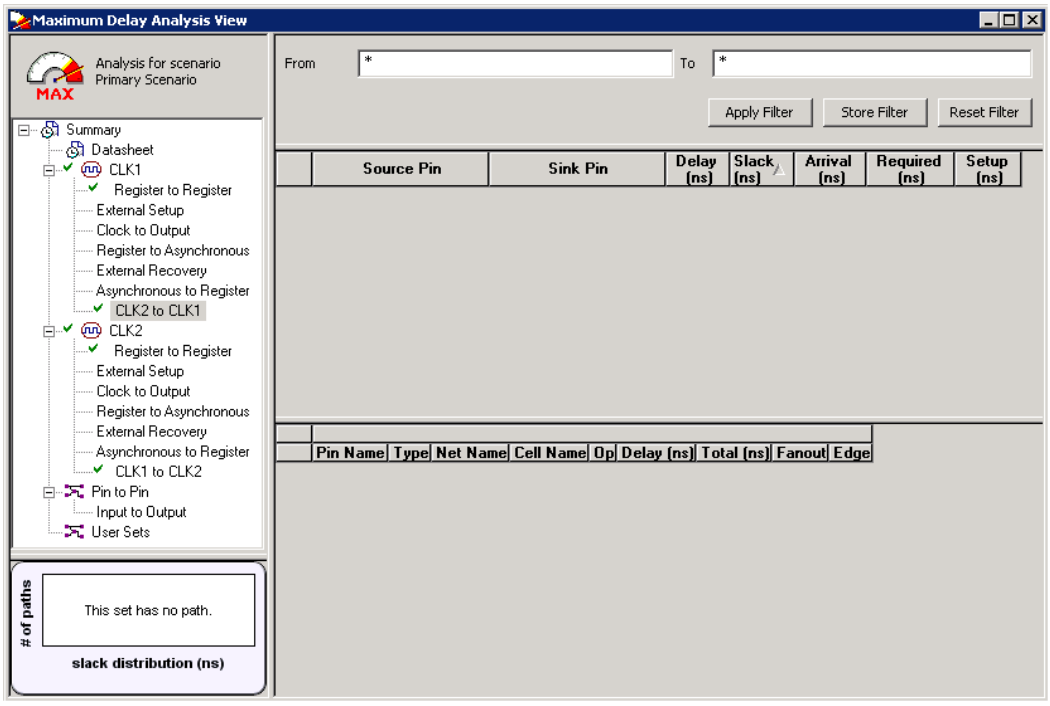


Figure 113 · Maximum Delay Analysis View

See Also

Inter-Clock Domain Analysis

Activating Inter-Clock Domain Analysis

Displaying Inter-Clock Domain Paths

Select Source or Destination Pins for Constraint Dialog Box

Set False Path Constraint Dialog Box

Timing Exceptions Overview

Use timing exceptions to overwrite the default behavior of the design path. Timing exceptions include:

- Setting multicycle constraint to specify paths that (by design) will take more than one cycle.
- Setting a false path constraint to identify paths that must not be included in the timing analysis or the optimization flow.
- Setting a maximum delay constraint on specific paths to relax or to tighten the original clock constraint requirement.

See Also

Specifying a Maximum Delay Constraint

Specifying a Minimum Delay Constraint

Specifying a Multicycle Constraint

Specifying a False Path Constraint

Changing Output Port Capacitance

Specifying a Maximum Delay Constraint

You set options in the Set Maximum Delay Constraint dialog box to relax or to tighten the original clock constraint requirement on specific paths.

To specify Max delay constraints:

1. Add the constraint in the Editable Constraints Grid or open the Set Maximum Delay Constraint dialog box using one of the following methods:
   - Click the  icon in the Constraints Editor.
   - From the Constraints Editor, right-click the Constraints menu and choose **Max delay**.

     The Set Maximum Delay Constraint dialog box appears (as shown below).

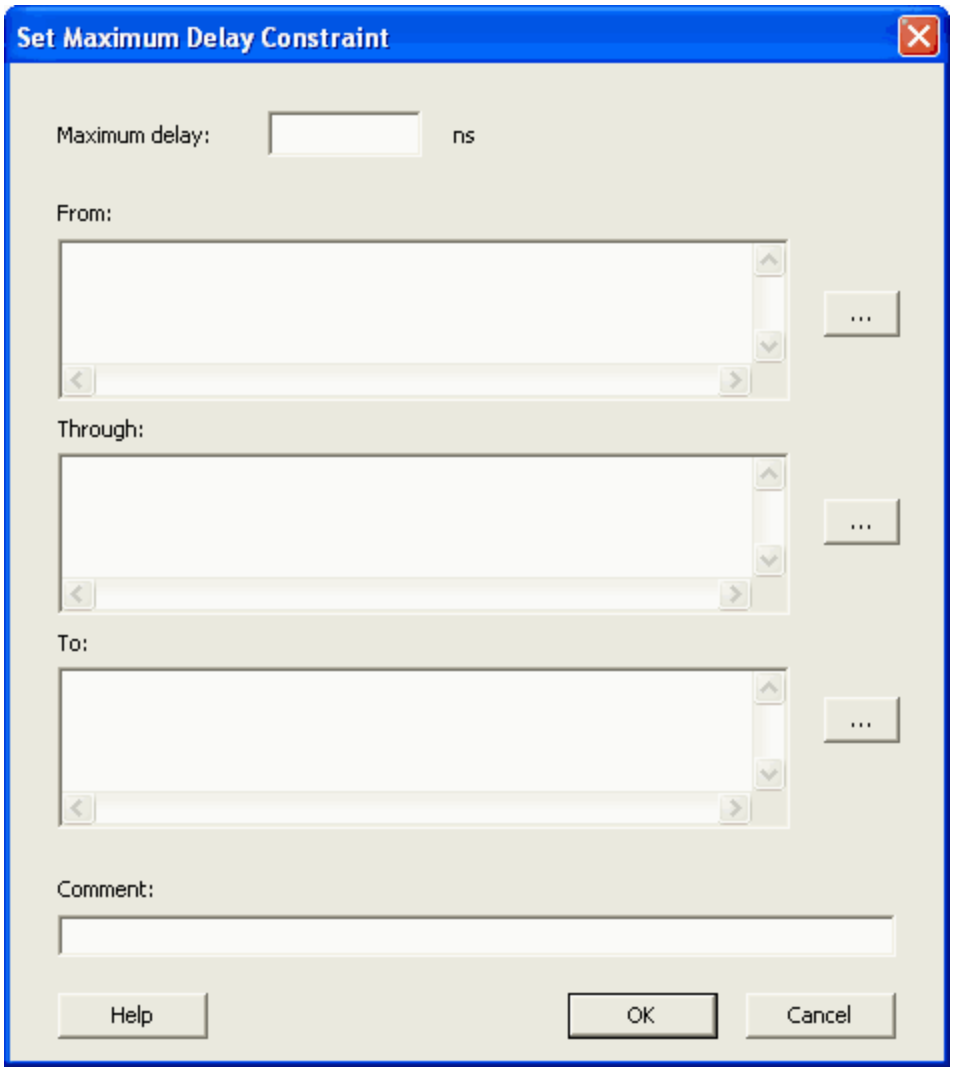


Figure 114 · Set Maximum Delay Constraint Dialog Box

2. Specify the delay in the **Maximum delay** field.
3. Specify the **From** pin(s). Click the **Browse** button next to **From** to open the Select Source Pins for Max Delay Constraint dialog box (as shown below).

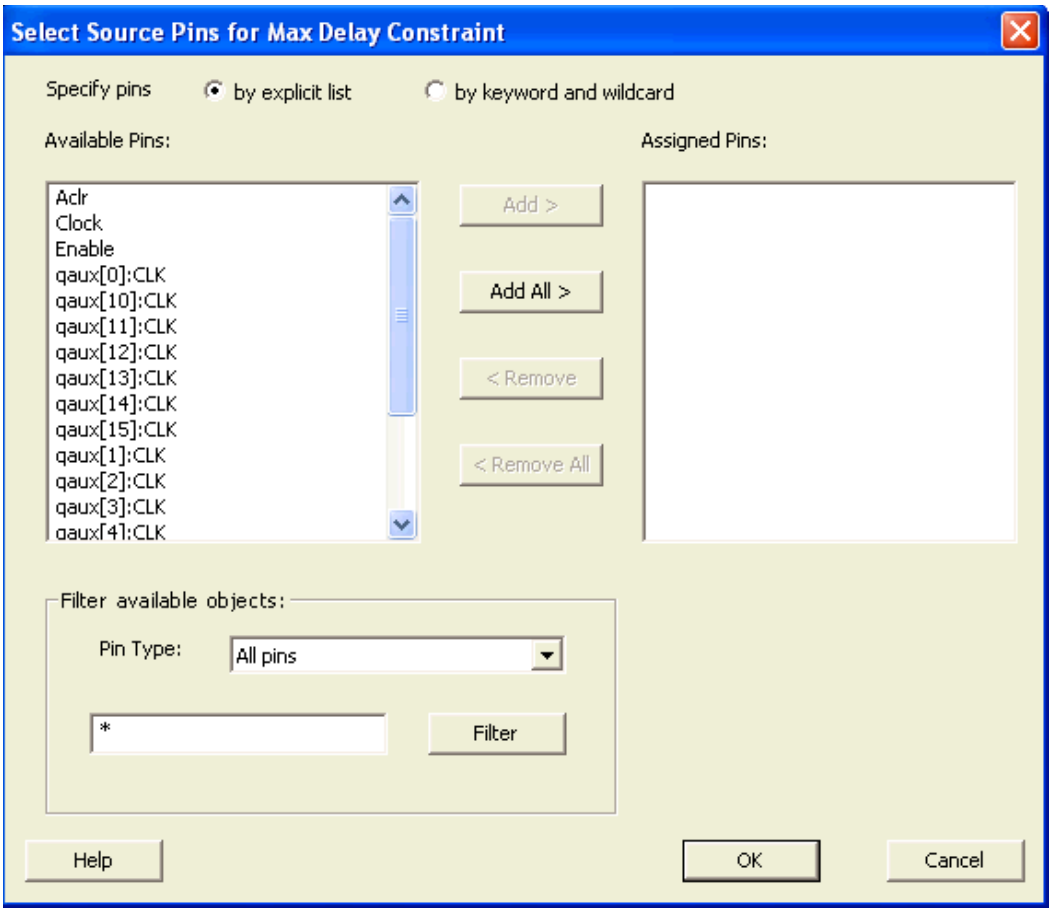


Figure 115 · Select Source Pins for Max Delay Constraint

4. Select **by explicit list**. (Alternatively, you can select **by keyword and wildcard**.)
5. Select the input pin(s) from the **Available Pins** list. You can also use **Filter available objects** to narrow the pin list. You can select multiple ports in this window.
6. Click **Add** or **Add All**. The input pin(s) move from the **Available Pins** list to the **Assigned Pins** list.
7. Click **OK**.The **Set Maximum Delay Constraint** dialog box displays the updated **From** pin(s) list.
8. Click the **Browse** button for **Through** and **To** and add the appropriate pin(s). The displayed list shows the pins reachable from the previously selected pin(s) list.
9. Enter comments in the **Comment** section.
10. Click **OK**.

SmartTime adds the maximum delay constraints to the Constraints List in the SmartTime Constraints Editor.

See Also

Timing Exceptions Overview

Set Maximum Delay Constraint dialog box

Specifying Maximum Delay Constraint

Specifying Multicycle Constraint

Specifying False Path Constraint

Changing Output Port Capacitance

Specifying a Minimum Delay Constraint

You set options in the Set Minimum Delay Constraint dialog box to relax or to tighten the original clock constraint requirement on specific paths.

To specify Min delay constraints:

1. Open the Set Minimum Delay Constraint dialog box using one of the following methods:
   - Click the  icon in the Constraints Editor.
   - From the Constraints Editor, right-click the Constraints Menu and choose **Min delay**.

   The Set Minimum Delay Constraint dialog box appears (as shown below).

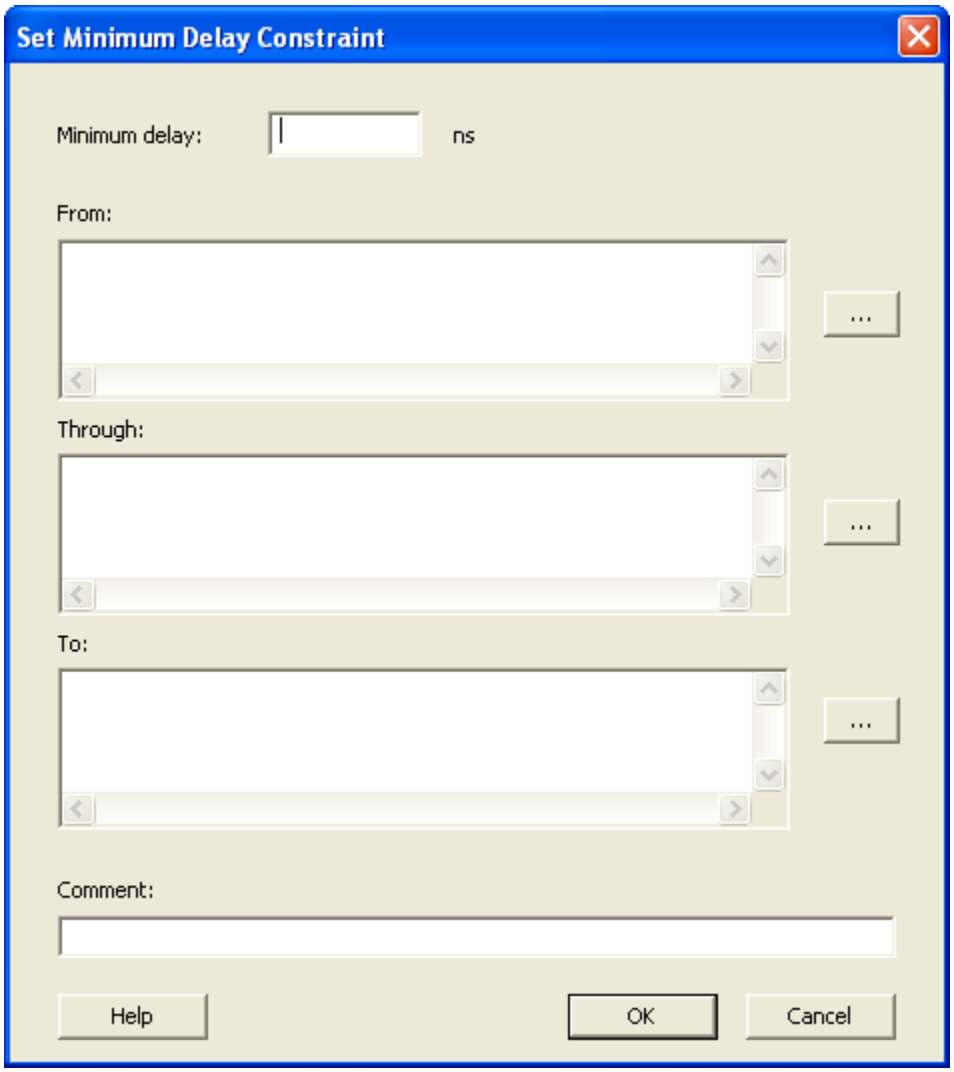


Figure 116 · Set Minimum Delay Constraint Dialog Box

2. Specify the delay in the **Minimum delay** field.
3. Specify the **From** pin(s). Click the **Browse** button next to **From** to open the Select Source Pins for Min Delay Constraint dialog box (as shown below).

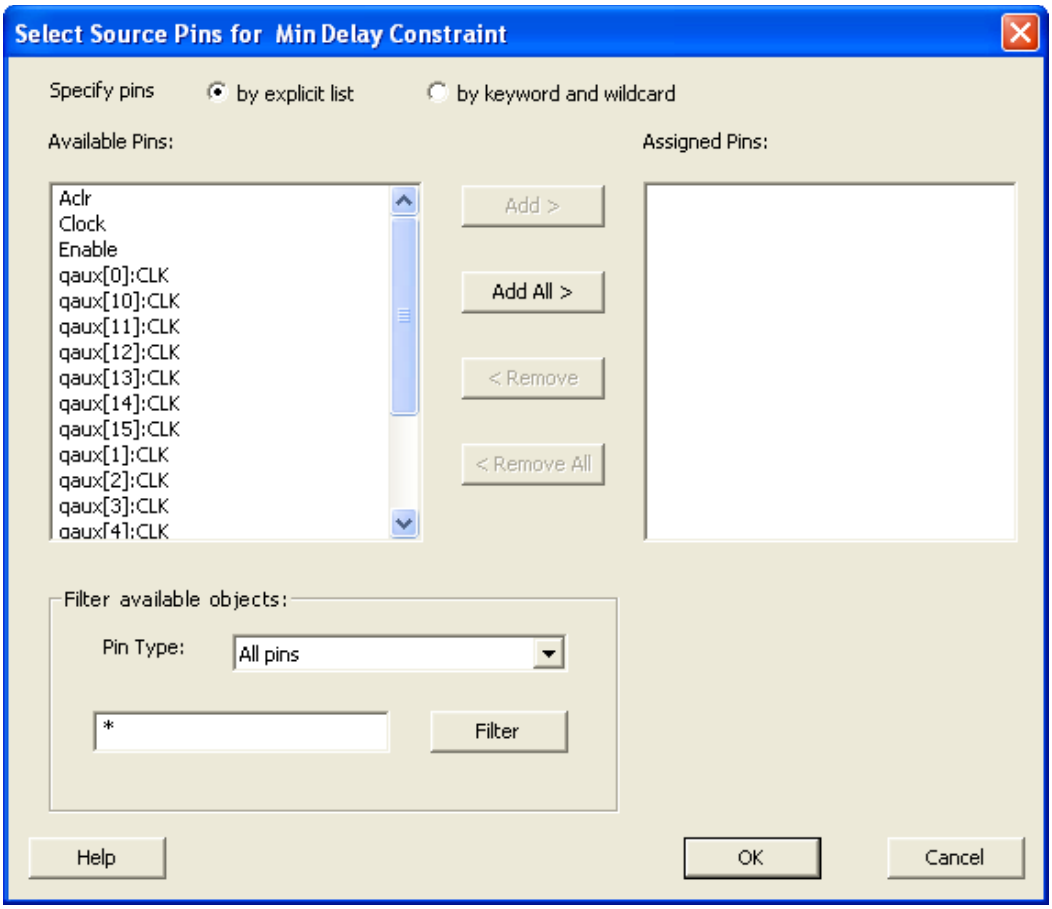


Figure 117 · Select Source Pins for Min Delay Constraint

4. Select **by explicit list**. (Alternatively, you can select **by keyword and wildcard**.)
5. Select the input pin(s) from the **Available Pins** list. You can also use **Filter available objects** to narrow the pin list. You can select multiple ports in this window.
6. Click **Add** or **Add All**. The input pin(s) move from the **Available Pins** list to the **Assigned Pins** list.
7. Click **OK**. The Set Minimum Delay Constraint dialog box displays the updated **From** pin(s) list.
8. Click the **Browse** button for **Through** and **To** and add the appropriate pin(s). The displayed list shows the pins reachable from the previously selected pin(s) list.
9. Enter comments in the **Comment** section.
10. Click **OK**.

SmartTime adds the minimum delay constraints to the Constraints List in the SmartTime Constraints Editor.

See Also

Timing Exceptions Overview

Set Maximum Delay Constraint dialog box

Specifying Maximum Delay Constraint

Specifying Multicycle Constraint

Specifying False Path Constraint

Changing Output Port Capacitance

Specifying a Multicycle Constraint

You set options in the Set Multicycle Constraint dialog box to specify paths that take multiple clock cycles in the current design.

To specify multicycle constraints:

1. Add the constraint in the Editable Constraints Grid or open the Set Multicycle Constraint dialog box using one of the following methods:
   - From the SmartTime Constraints Editor, choose **Constraint > MultiCycle**.
   - Click the  icon.
   - Right-click the **Multicycle** option in the Constraint Browser and select **Add Multicycle Path Constraint**.

     The Set Multicycle Constraint dialog box appears (as shown below).

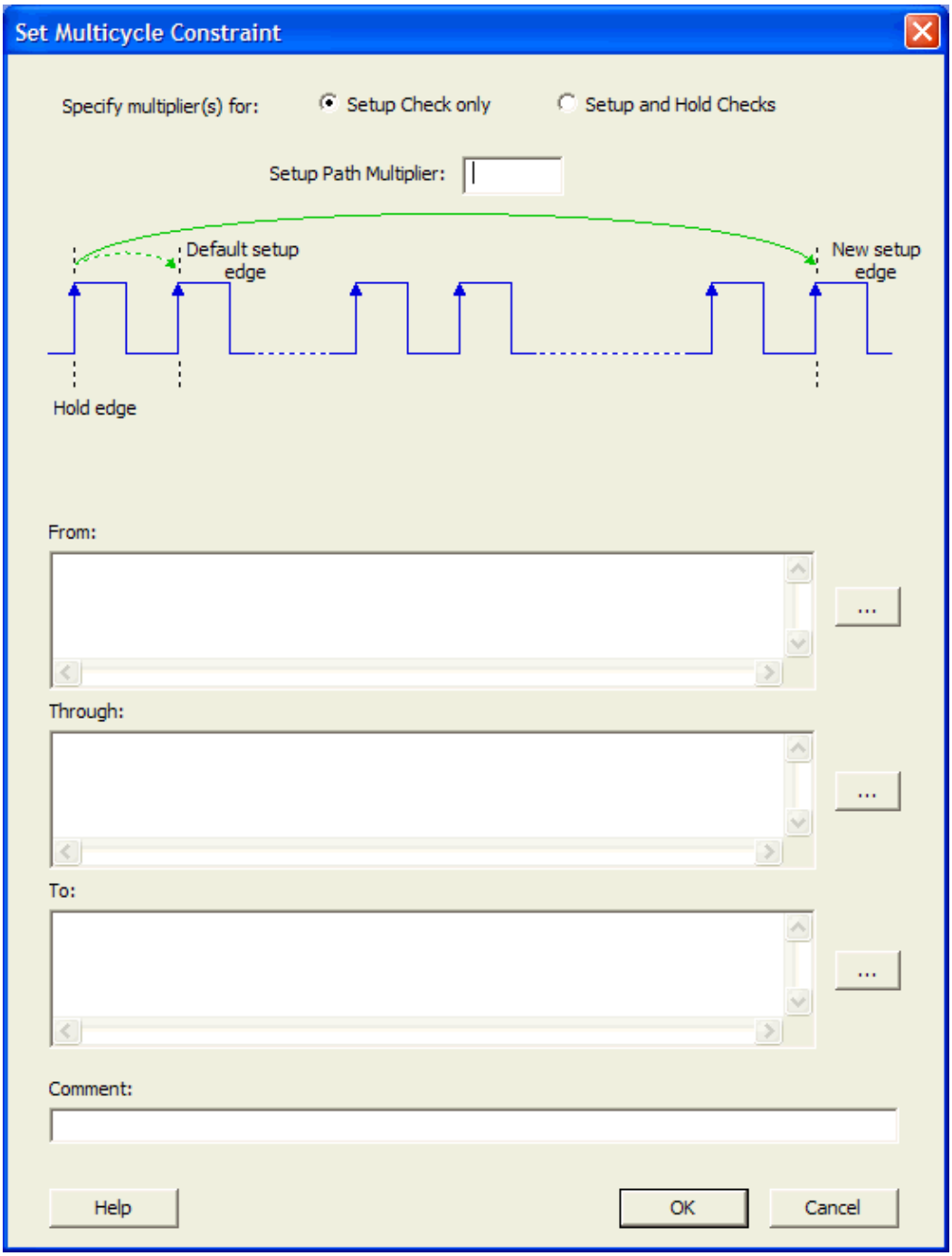


Figure 118 · Set Multicycle Constraint Dialog Box

2. Specify the number of cycles in the **Setup Path Multiplier**.
3. Specify the **From** pin(s). Click the **Browse** button next to **From** to open the Select Source Pins for Constraint dialog box (as shown below).

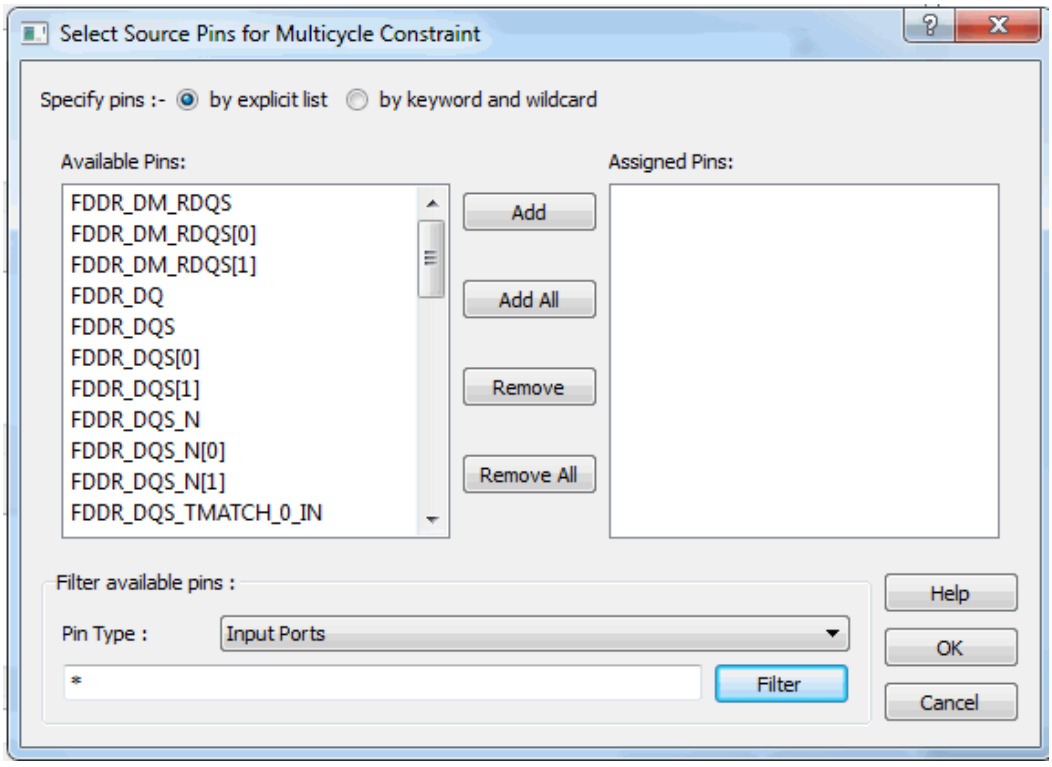


Figure 119 · Select Source Pins for Multicycle Constraint

4. Select **by explicit list**. (Alternatively, you can select **by keyword and wildcard**. For details, refer to the Select Source or Destination Pins for Constraint Dialog Box.)
5. Select the input pin(s) from the **Available Pin** list. You can use **Filter available objects** to narrow the pin list. You can select multiple ports in this window.
6. Click **Add** or **Add All** to move theinput pin(s) move from the **Available pins** list to the **Assigned Pins** list.
7. Click **OK**.

The Set Multicycle Constraint dialog box displays the updated representation of the **From** pin(s) (as shown below).

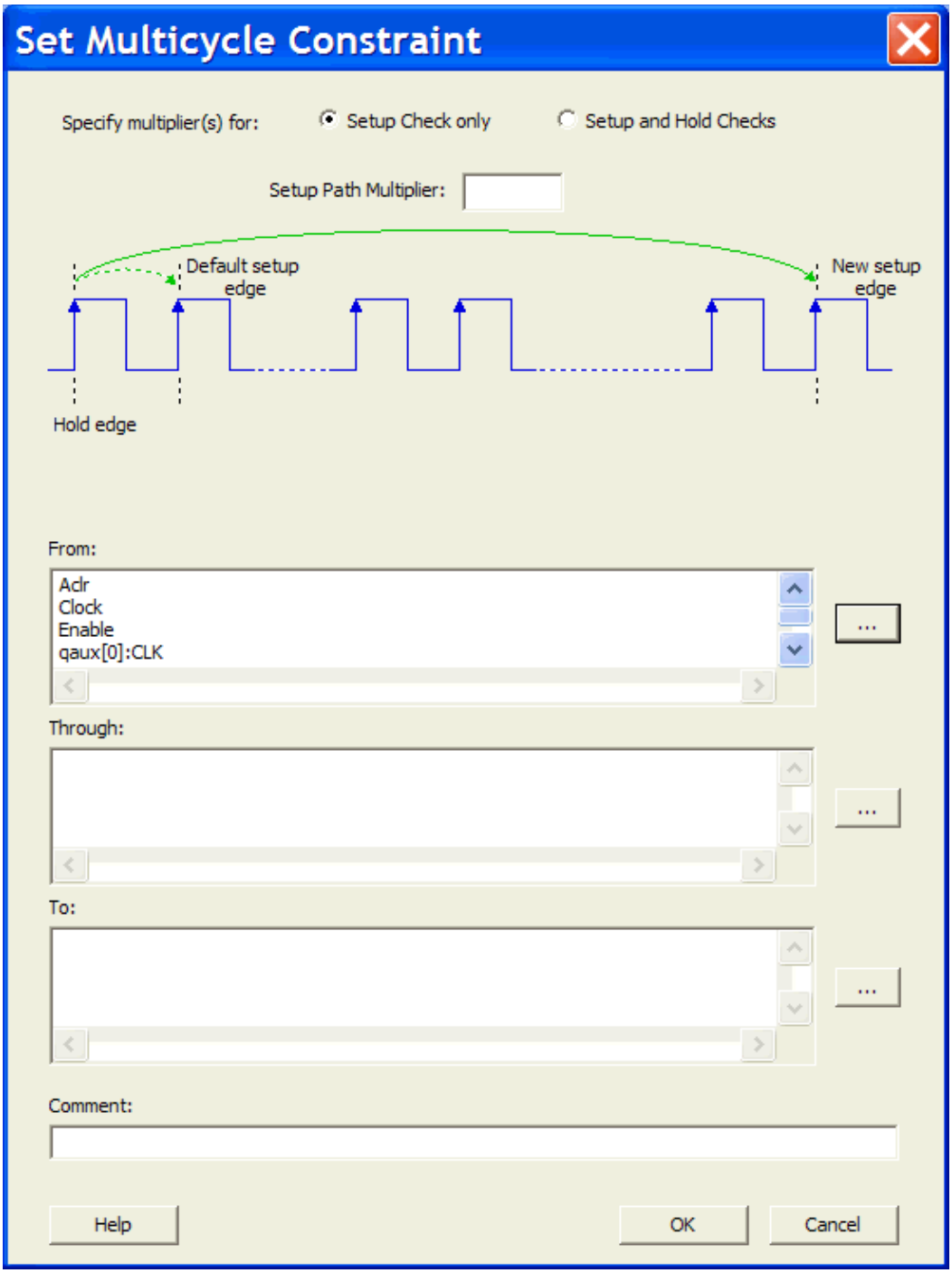


Figure 120 · Set Multicycle Constraint Dialog Box

8. Click the browse button for **Through** and **To** and add the appropriate pins. The displayed list shows the pins reachable from the previously selected pin(s) list.
9. Enter comments in the **Comment** section.
10. Click **OK**. SmartTime adds the multicycle constraints to the Constraints List in the SmartTime Constraints Editor.

See Also

Set Multicycle Constraint Dialog Box

Specifying a False Path Constraint

You set options in the Set False Path Constraint dialog box to define specific timing paths as false.

To specify False Path constraints:

1. Add the constraint in the Editable Constraints Grid or open the Set False Path Constraint dialog box. You can do this by using one of the following methods:
   - From the SmartTime **Constraints** menu, choose **False Path**.
   - Click the  icon.
   - Right-click **False Path** in the Constraint Browser and choose **Add False Path Constraint**.

   The Set False Path Constraint dialog box appears (as shown below).

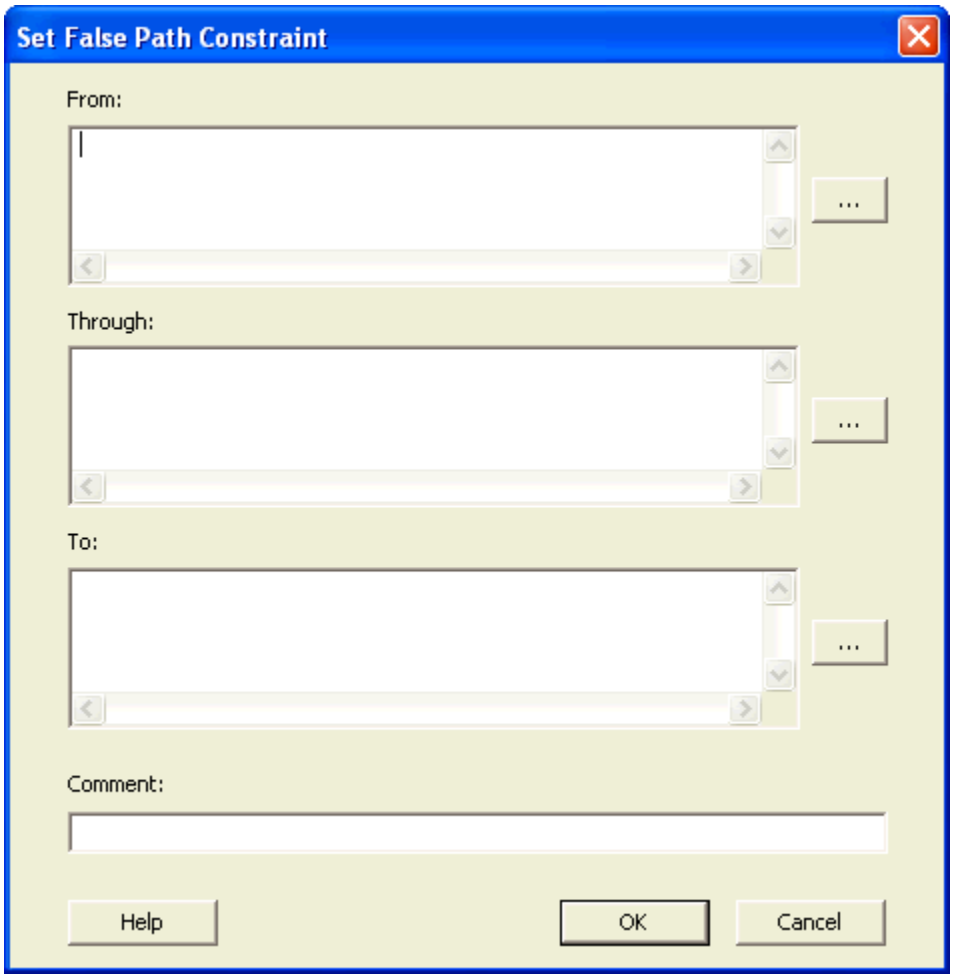


Figure 121 · Set False Path Constraint Dialog Box

2. Specify the **From** pin(s). Click the **Browse** button next to **From** to open the Select Source Pins for False Path Constraint dialog box (as shown below).

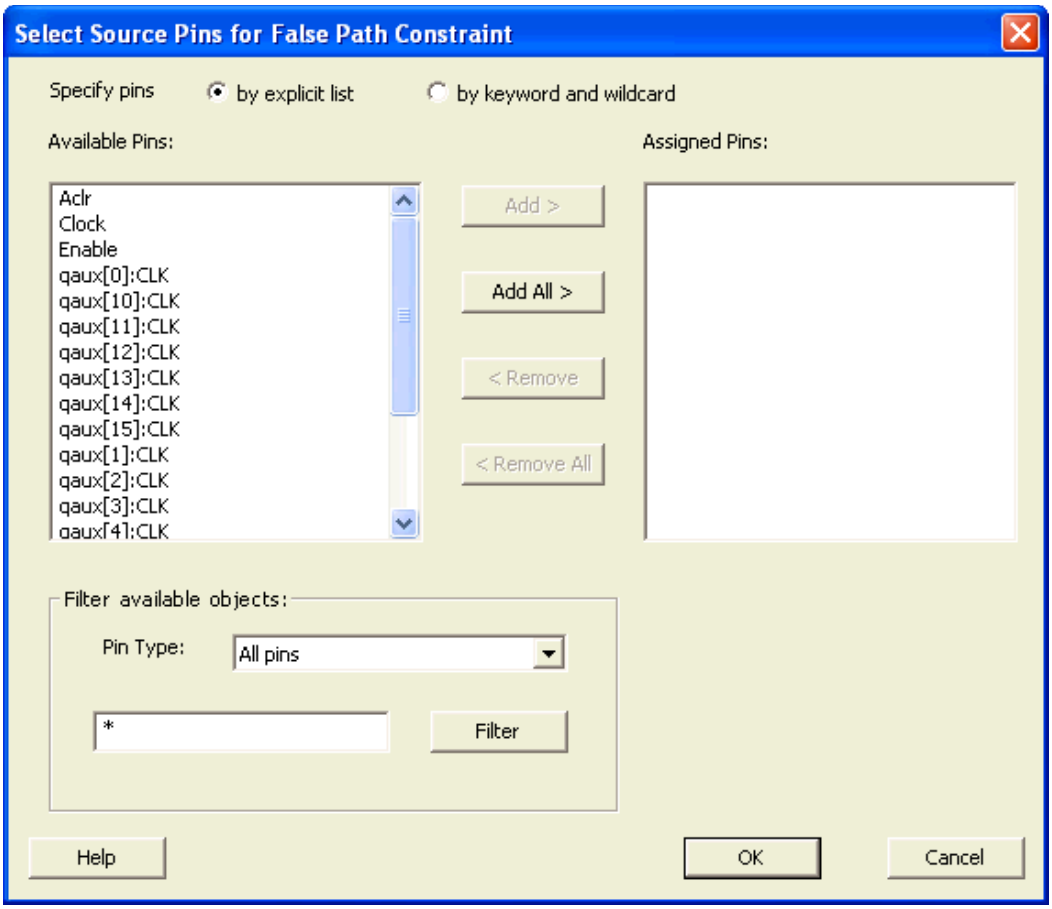


Figure 122 · Select Source Pins for False Path Constraint Dialog Box

3. Select **by explicit list**. (Alternatively, you can select **by keyword and wildcard**. For details, refer to Select Source or Destination Pins for Constraint Dialog Box.)
4. Select the input pin(s) from the **Available Pin** list. You can use **Filter available objects** to narrow the pin list. You can select multiple ports in this window.
5. Click **Add** or **Add All.** The input pin(s) move from the **Available Pins** list to the **Assigned Pins** list.
6. Click **OK**.

   The Set False Path Constraint dialog box displays the updated representation of the **From** pin(s).
7. Click the **Browse** button for **Through** and **To** and add the appropriate pin(s). The displayed list shows the pins reachable from the previously selected pin(s) list.
8. Enter comments in the **Comment** section.
9. Click **OK**.

SmartTime adds the False Path constraints to the Constraints List in the SmartTime Constraints Editor.

See Also

Set False Path Constraint Dialog Box

Changing Output Port Capacitance

Output propagation delay is affected by both the capacitive loading on the board and the I/O standard. The I/O Attribute Editor in ChipPlanner provides a mechanism for setting the expected capacitance to improve the propagation delay model. SmartTime automatically uses the modified delay model for delay calculations.

To change the output port capacitance and view the effect of this change in SmartTime Timing Analyzer, refer to the following example. The figure below shows the delay from FF3 to output port OUT2. It shows a delay of 6.603 ns based on the default loading of 35 pF.

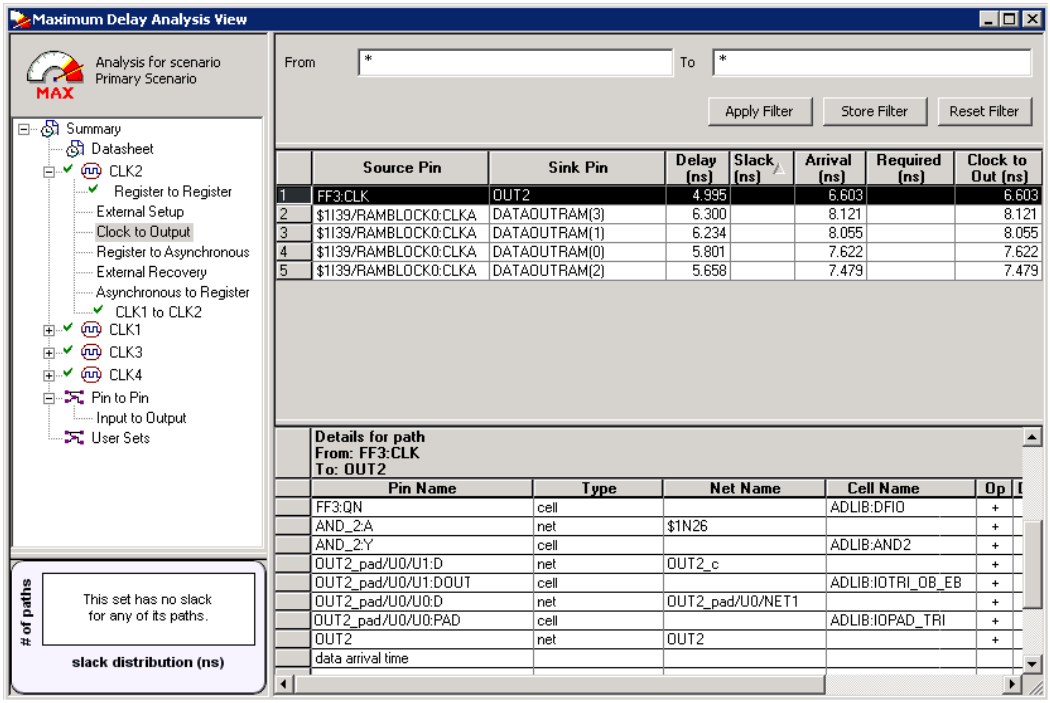


Figure 123 · Maximum Delay Analysis View

If your board has output capacitance of 75pf on OUT2, you must perform the following steps to update the timing number:

1. Open the I/O Attribute Editor and change the output load to 75pf.

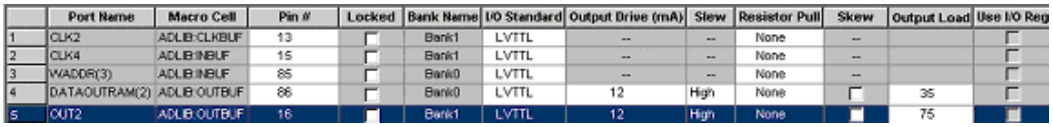


Figure 124 · I/O Attribute Editor View

2. Select **File > Save**.
3. Select **File > Close**.
4. Open the SmartTime Timing Analyzer.

You can see that the Clock to Output delay changed to 7.723 ns.

Specifying Clock Source Latency

Use clock source latency to specify the delay from the clock generation point to the clock definition point in the design.

Clock source latency defines the delay between an external clock source and the definition pin of a clock within SmartTime. It behaves much like an input delay constraint.

You can specify both an "early" delay and a "late" delay for this latency, providing an uncertainty which SmartTime propagates through its calculations. Rising and falling edges of the same clock can have different latencies. If only one value is provided for the clock source latency, it is taken as the exact latency value, for both rising and falling edges.

To specify the clock source latency:

1. Add the constraint in the Editable Constraints Grid or open the Set Clock Source Latency dialog box using one of the following methods:

- From the SmartTime Constraints Editor window, choose **Constraints > Clock Source Latency.**
- Click the  icon in the Constraints Editor.
- Click **Clock Source Latency** in the Constraint Browser.

2. Select a clock pin on which to set the source latency. You can only specify a clock source latency on a clock pin that has a clock constraint. Additionally, you may apply only one clock source latency constraint to each constrained clock pin.
3. Enter the Late Rise, Early Rise, Late Fall, and Early Fall values as required for your design.

   Note: An 'early' value larger than a 'late' value can result in optimistic timing analysis.
4. Select the **Falling Same As Rising** check box to indicate that falling clock edges have the same latency as rising clock edges.
5. Select the **Early Same As Late** check box to use a single value for the clock latency, rather than a range, by clicking the checkbox.
6. Enter any comments to be attached to the constraint.
7. Click **OK**. The new constraint appears in the constraints list.

   Note: When you choose Save from the File menu, SmartTime saves the newly-created constraint in the database.

See Also

Set Clock Source Latency Dialog Box

Specifying a False Path Constraint

You set options in the Set False Path Constraint dialog box to define specific timing paths as false.

To specify False Path constraints:

1. Add the constraint in the Editable Constraints Grid or open the Set False Path Constraint dialog box. You can do this by using one of the following methods:
   - From the SmartTime **Constraints** menu, choose **False Path**.
   - Click the  icon.
   - Right-click **False Path** in the Constraint Browser and choose **Add False Path Constraint**.

   The Set False Path Constraint dialog box appears (as shown below).

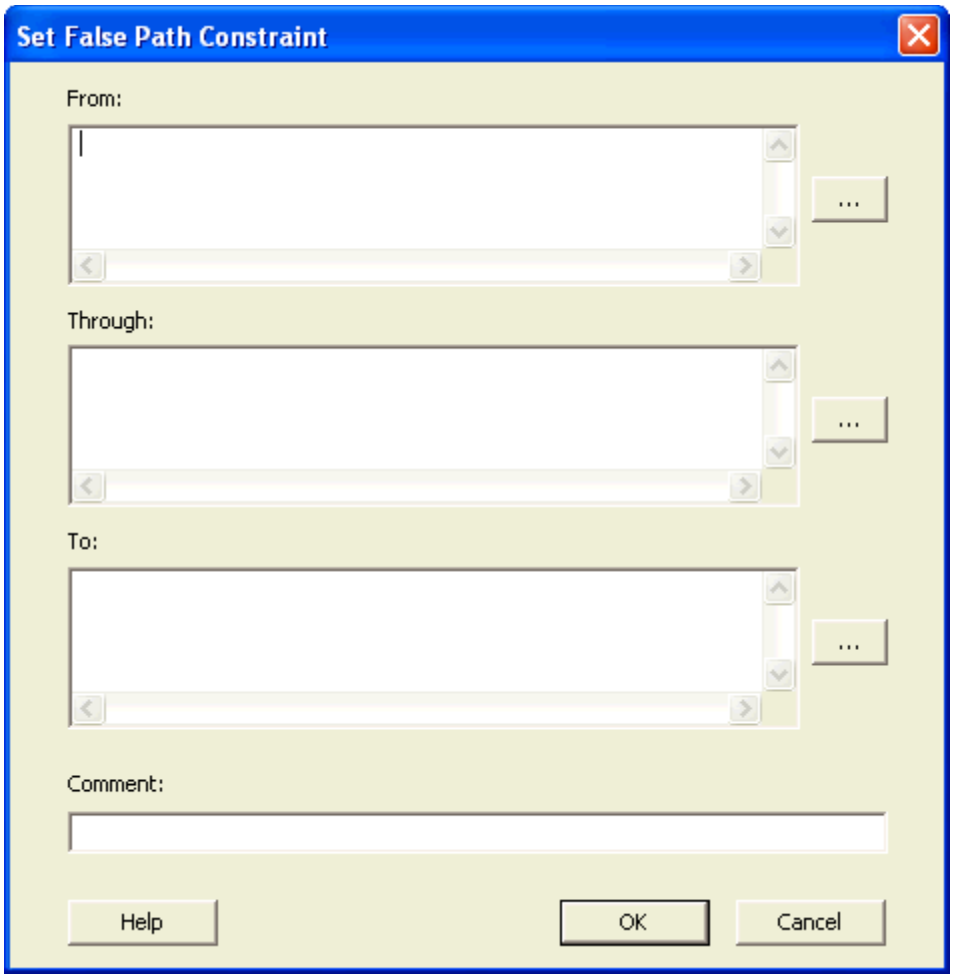


Figure 125 · Set False Path Constraint Dialog Box

2. Specify the **From** pin(s). Click the **Browse** button next to **From** to open the Select Source Pins for False Path Constraint dialog box (as shown below).

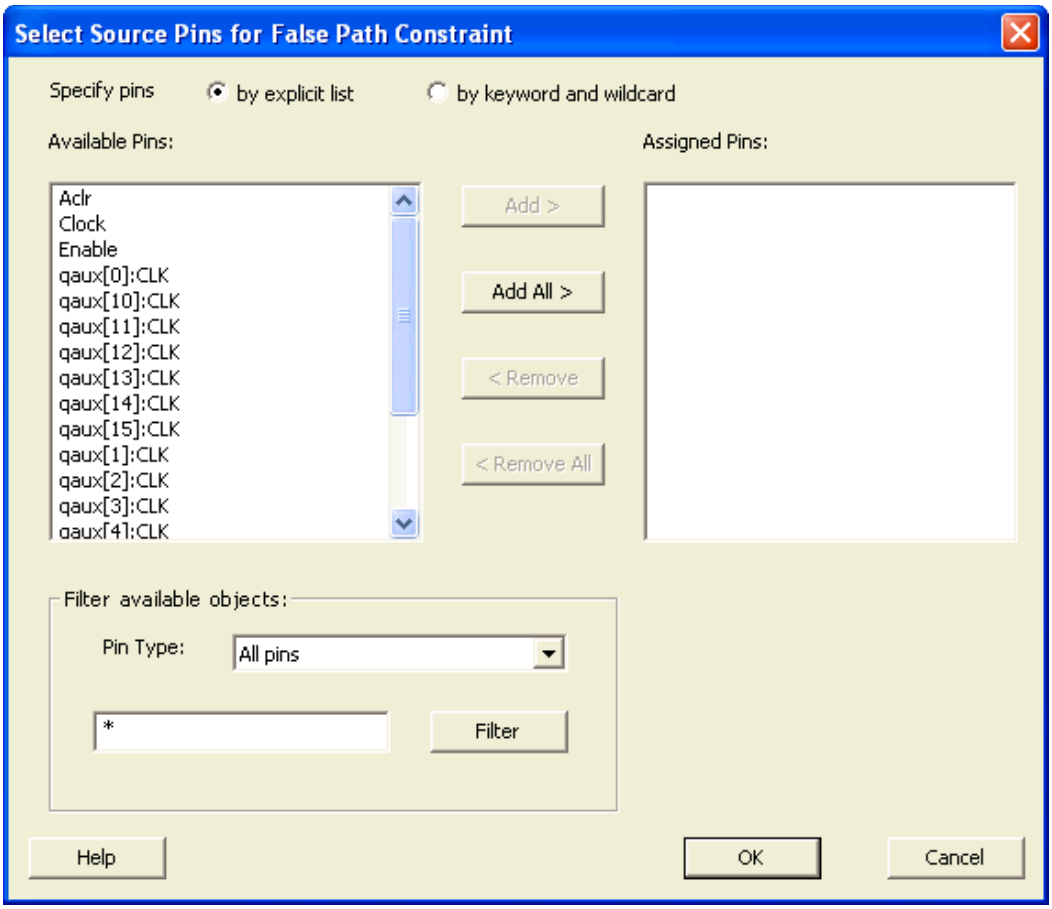


Figure 126 · Select Source Pins for False Path Constraint Dialog Box

3. Select **by explicit list**. (Alternatively, you can select **by keyword and wildcard**. For details, refer to Select Source or Destination Pins for Constraint Dialog Box.)
4. Select the input pin(s) from the **Available Pin** list. You can use **Filter available objects** to narrow the pin list. You can select multiple ports in this window.
5. Click **Add** or **Add All.** The input pin(s) move from the **Available Pins** list to the **Assigned Pins** list.
6. Click **OK**.

   The Set False Path Constraint dialog box displays the updated representation of the **From** pin(s).
7. Click the **Browse** button for **Through** and **To** and add the appropriate pin(s). The displayed list shows the pins reachable from the previously selected pin(s) list.
8. Enter comments in the **Comment** section.
9. Click **OK**.

SmartTime adds the False Path constraints to the Constraints List in the SmartTime Constraints Editor.

See Also

Set False Path Constraint Dialog Box

Specifying Clock-to-Clock Uncertainty Constraint

Use the clock-to-clock uncertainty constraint to model tracking jitter between two clocks in your design.

To specify the clock-to-clock uncertainty constraint:

1. Add the constraint in the Editable Constraints Grid or open the Set Clock-to-Clock Uncertainty Constraint Dialog Box using one of the following methods:
   - From the SmartTime Constraints Editor, choose **Constraints > Clock-to-Clock Uncertainty Constraint.**
   - Click the  icon.
   - Right-click **Clock-to-Clock Uncertainty Constraint** in the Constraint Browser.
2. Specify the *from* and *to* clocks and specify the uncertainty in ns.
4. Enter any comments to be attached to the constraint.
5. Click **OK**. The new constraint appears in the constraints list.

   Note: When you choose Save from the File menu, SmartTime saves the newly-created constraint in the database.

See Also

Set Clock-to-Clock Uncertainty Constraint Dialog Box

Generating Timing Reports

Types of Reports

Using SmartTime you can generate the following types of reports:

- **Timer report** – This report displays the timing information organized by clock domain.
- **Timing Violations report** – This flat slack report provides information about constraint violations.
- **Bottleneck report** – This report displays the points in the design that contribute to the most timing violations.
- **Datasheet report** – This report describes the characteristics of the pins, I/O technologies, and timing properties in the design.
- **Constraints Coverage report** – This report displays the overall coverage of the timing constraints set on the current design.
- **Combinational Loop report** – This report displays loops found during initialization.

See Also

Generating a Timing Report

Generating a Timing Violation Report

Generating a datasheet report

Generating a bottleneck report

Generating a constraints coverage report

Generating a Combinational Loop Report

Generating a Timing Report

The timing report enables you to quickly determine if any timing problems exist in your design. The Maximum Delay Analysis timing report lists the following information about your design:

- Maximum delay from input I/O to output I/O
- Maximum delay from input I/O to internal registers
- Maximum delay from internal registers to output I/O
- Maximum delays for each clock network
- Maximum delays for interactions between clock networks

To generate a timing report:

1. From the SmartTime Max/Min Delay Analysis View, choose **Reports > Timer**. The Timing Report Options Dialog Box appears.
2. Select the options you want to include in the report, and then click **OK**.

The timing report appears in a separate window.

See Also

Understanding Timing Reports

Timing Report Options Dialog Box

Understanding Timing Reports

The timing report contains the following sections:

Header

The header lists:

- The report type
- The version of Designer used to generate the report
- The date and time the report was generated
- General design information (name, family, etc.)

Summary

The summary section reports the timing information for each clock domain.

By default, the clock domains reported are the explicit clock domains that are shown in SmartTime. You can filter the domains and get only specific sections in the report (see Timing Report Options).

Path Sections

The paths section lists the timing information for different types of paths in the design. This section is reported by default. You can deselect this option in the Timing Report Options dialog box.

By default, the number of paths displayed per set is 5.

You can filter the domains using the Timing Report Options dialog box.

You can also view the stored filter sets in the generated report using the timing report options (see Timing Report Options). The filter sets are listed by name in their appropriate section, and the number of paths reported for the filter set is the same as for the main sets.

By default, the filter sets are not reported.

Clock domains

The paths are organized by clock domain.

Register to Register set

This set reports the paths from the registers clock pins to the registers data pins in the current clock domain.

External Setup set

This set reports the paths from the top level design input ports to the registers in the current clock domain.

Clock to output set

This set reports the paths from the registers clock pins to the top level design output ports in the current clock domain.

Register to Asynchronous set

This set reports the paths from registers to asynchronous control signals (like asynchronous set/reset).

External Recovery set

This set reports the external recovery check timing for asynchronous control signals (like asynchronous set/reset).

Asynchronous to Register set

This set reports the paths from asynchronous control signals (like asynchronous set/reset) to registers

Inter-clock domain

This set reports the paths from the registers clock pins of the specified clock domain to the registers data pins in the current clock domain. Inter-domain paths are not reported by default.

Pin to pin

This set lists input to output paths and user sets. Input to output paths are reported by default. To see the user-defined sets, use the Timing Report Options dialog box.

Input to output set

This set reports the paths from the top level design input ports to top level design output ports.

Expanded Paths

Expanded paths can be reported for each set. By default, the number of expanded paths to report is set to 1. You can select and change the number when you specify Timing Report Options.

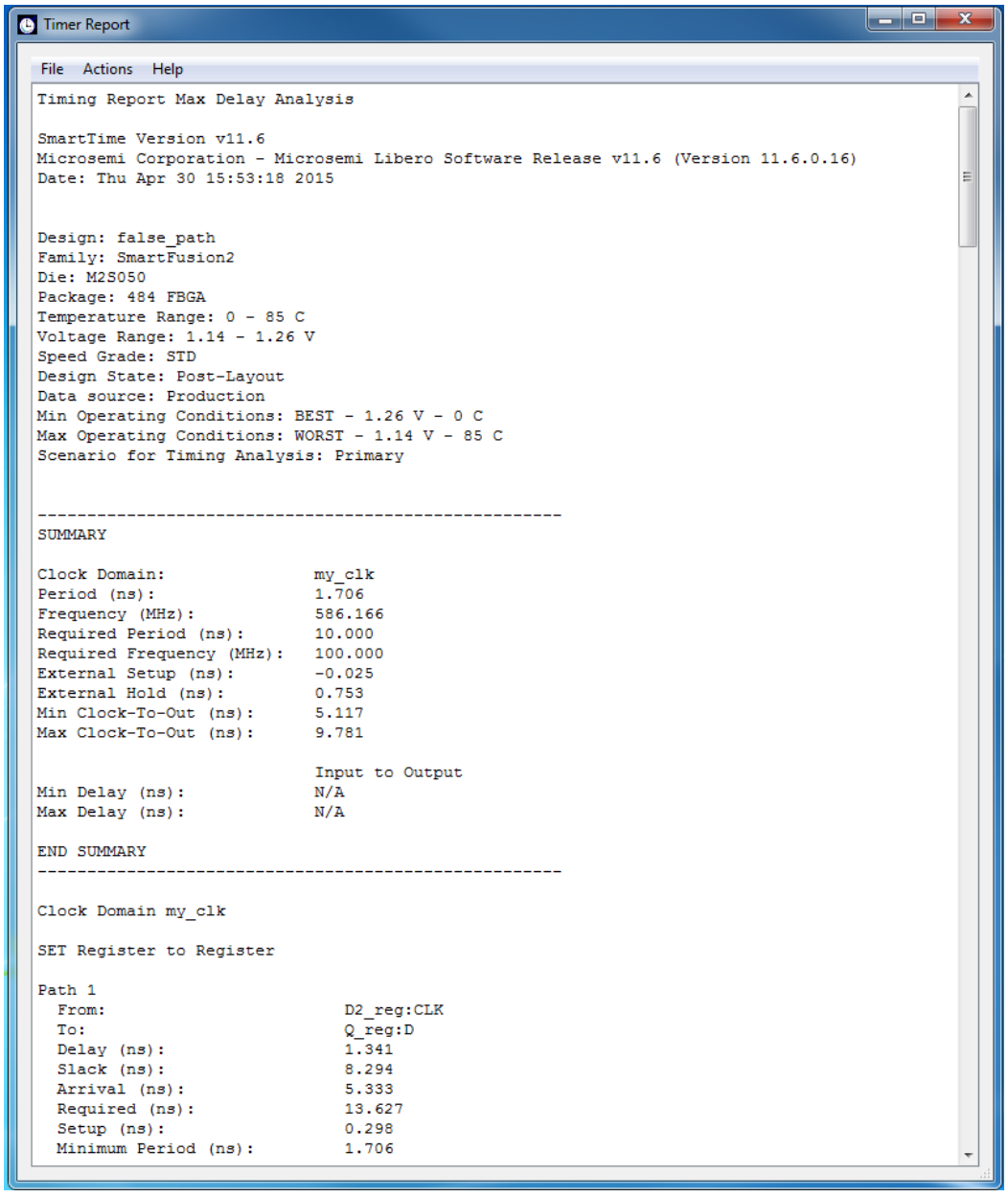


Figure 127 · Timing Report

See Also

Generating timing report

Timing Report Options dialog box

Generating a Timing Violation Report

The timing violations report provides a flat slack report centered around constraint violations.

To generate a timing violation report

1. From the SmartTime Max/Min Delay Analysis View window, choose **Tools > Reports > Timing Violations**. The Timing Violations Report Options Dialog Box appears.
2. Select the options you want to include in the report, and then click **OK**. The timing violations report appears in a separate window.

See Also

Understanding Timing Violation Reports

Understanding Timing Violation Reports

The timing violation report contains the following sections:

Header

The header lists:

- The report type
- The version of Designer used to generate the report
- The date and time the report was generated
- General design information (name, family, etc.)

Paths

The paths section lists the timing information for the violated paths in the design.

The number of paths displayed is controlled by two parameters:

- A maximum slack threshold to report
- A maximum number or path to report

By default, the slack threshold is 0 and the number of paths is limited. The default maximum number of paths reported is 100.

All clocks domains are mixed in this report. The paths are listed by decreasing slack.

You can also choose to expand one or more paths. By default, no paths are expanded. For details, see the timing violation report options.

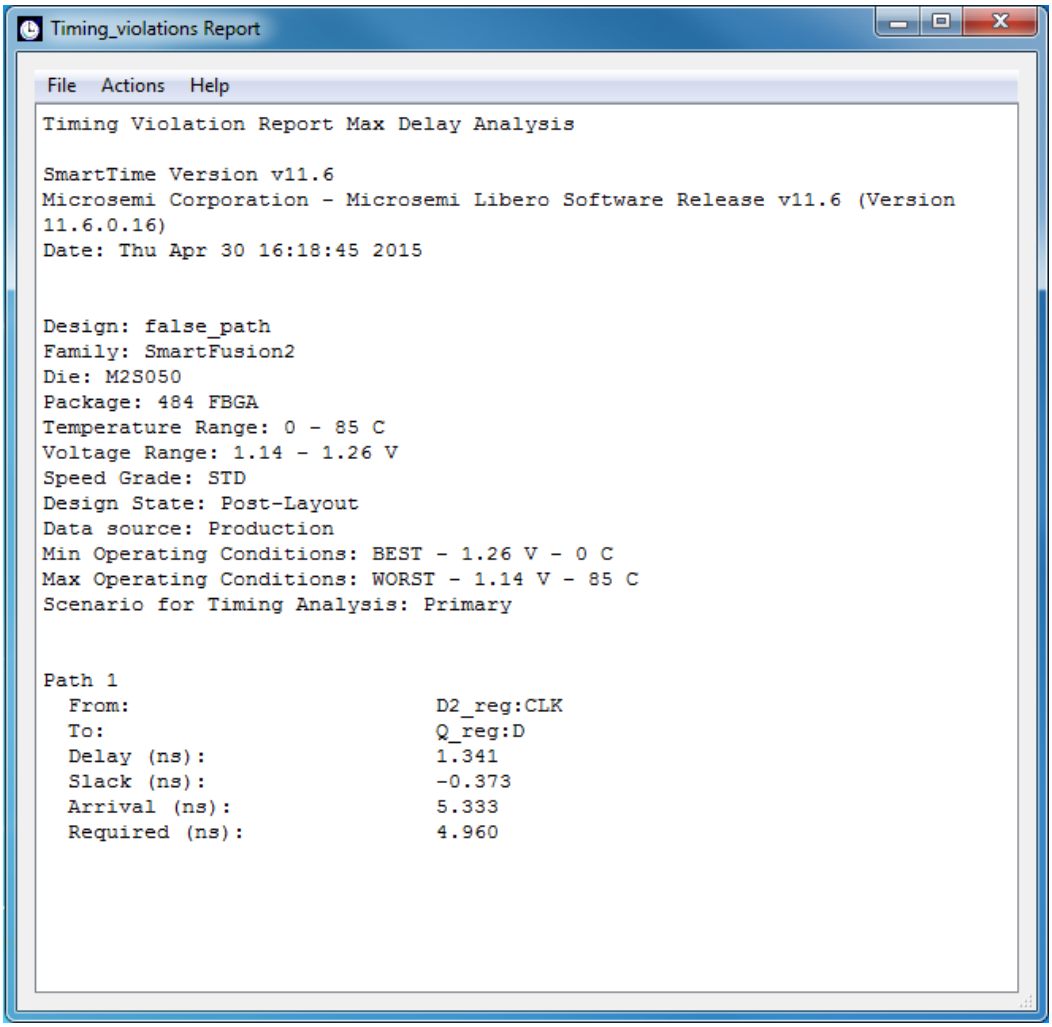


Figure 128 · Timing Violations Report

See Also

Generating a Timing Violation Report

Timing Violations Report Options Dialog Box

Generating a Constraints Coverage Report

The constraints coverage report contains information about the constraints in the design.

To generate a constraints coverage report, from the SmartTime Max/Min Delay Analysis View, choose **Tools > Reports > Constraints Coverage**. The report appears in a separate window.

See Also

Understanding Constraints Coverage Reports

Understanding Constraints Coverage Reports

The constraint coverage displays the overall coverage of the timing constraints set on the current design. You can generate this report either from within Designer or within SmartTime Analyzer. The report contains three sections:

- Coverage Summary

- Results by Clock Domain
- Enhancement Suggestions

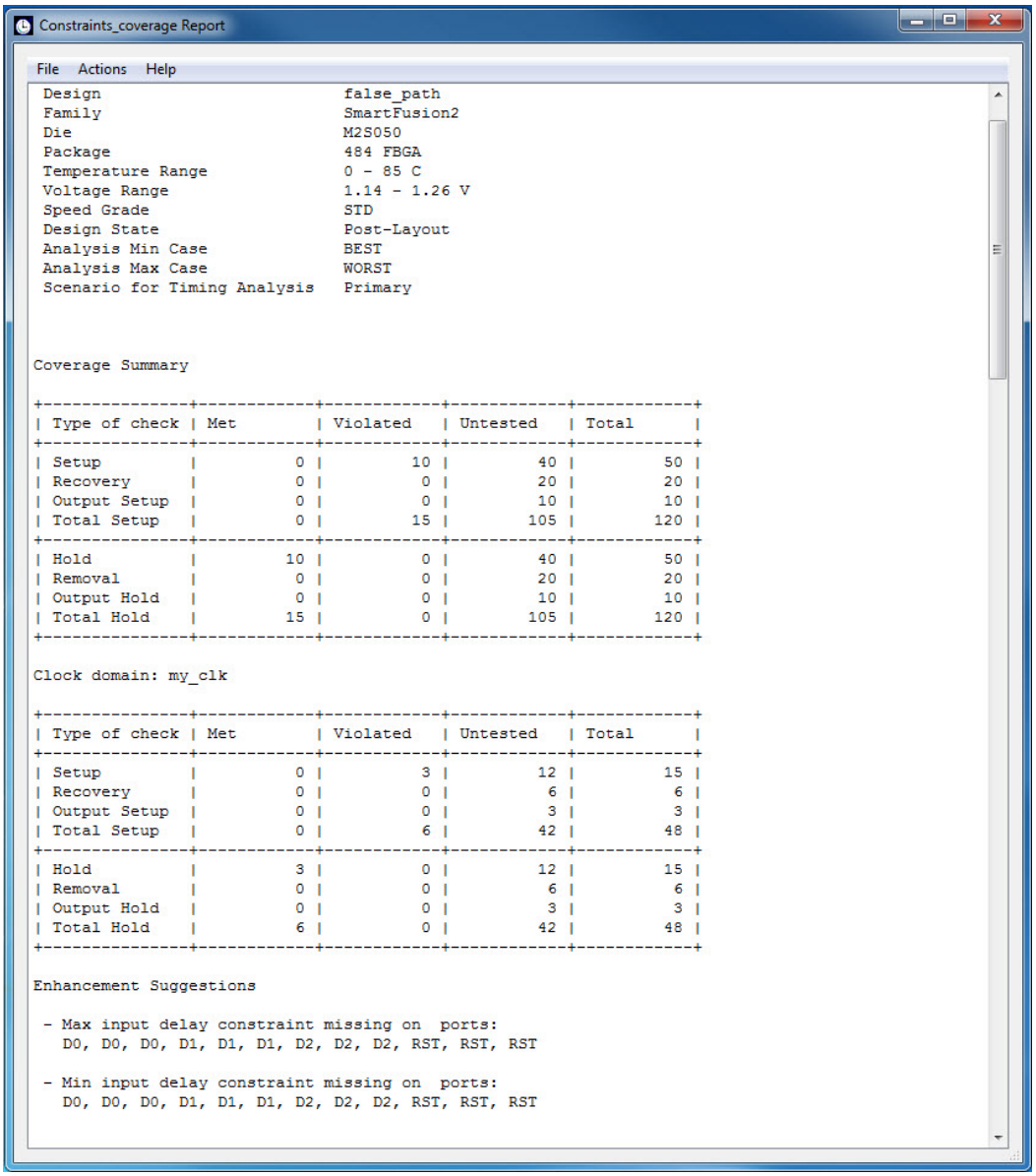


Figure 129 · Constraints Coverage Report

Coverage Summary

The coverage summary gives statistical information on the timing constraint in the design. For each type of timing checks (Setup, Recovery, Output, Hold and Removal), it specifies how many are Met (there is a constraint and it is satisfied), Violated (there is a constraint and it is not satisfied), or Untested (no constraint was found).

Clock Domain

This section provides a coverage summary for each clock domain.

Enhancement Suggestions

The enhancement suggestion reports, per clock domain, a list of constraints that can be added to the design to improve the coverage. It also reports if some options impacting the coverage can be changed.

Detailed Stats

This section provides detailed suggestions regarding specific clocks or I/O ports that may require to be constrained for every pin/port that requires checks.

See Also

Clock

Input delay

Output delay

Setting SmartTime Options

Generating a Bottleneck Report

The bottleneck report provides a list of the bottlenecks in the design.

To generate a bottleneck report, from the,SmartTime Max/Min Delay Analysis View, choose **Tools > Reports > Bottleneck**. The report appears in a separate window.

See Also

Understanding Bottleneck Reports

Timing Bottleneck Analysis Options Dialog Box

Understanding Bottleneck Reports - SmartFusion2, IGLOO2, RTG4

A bottleneck is a point in the design that contributes to multiple timing violations. The purpose of the bottleneck report is to provide a list of the bottlenecks in the design. You can generate this report either from SmartTime Analyzer. It contains two sections

- Device Description
- Bottleneck Analysis

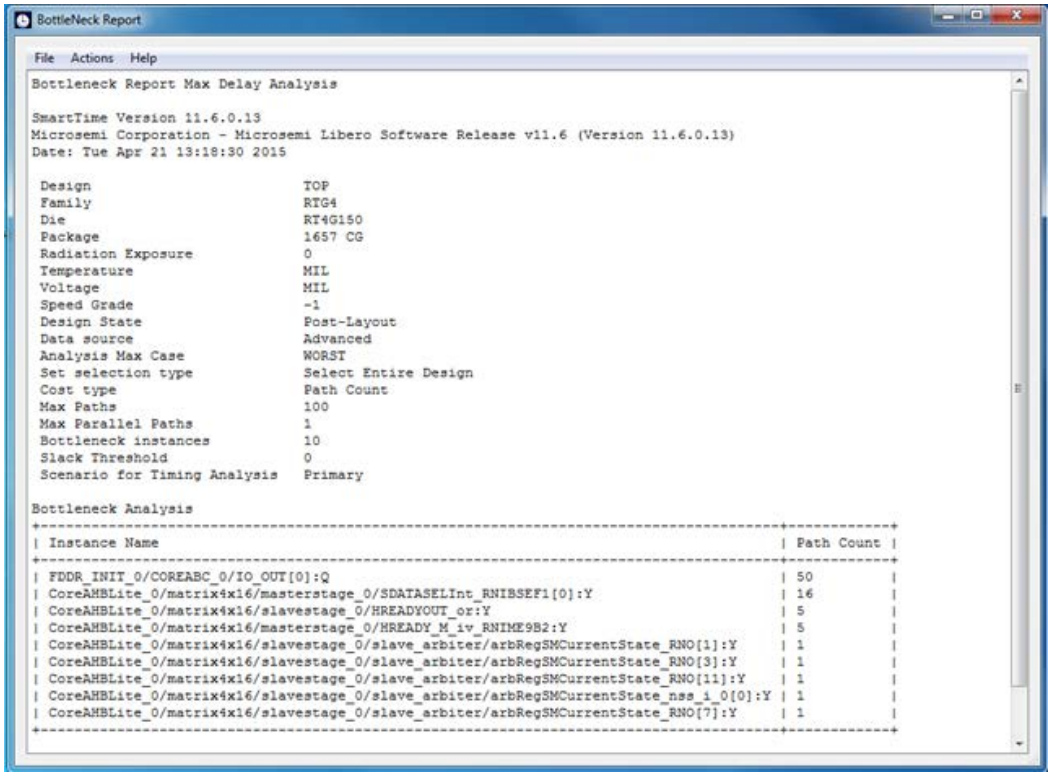


Figure 130 · Bottleneck Report

The bottleneck can only be computed if and only a cost type is defined. There are two options available:

- **Path count:** This cost type associates the severity of the bottleneck to the count of violating/critical paths that traverse the instance.
- **Path cost:** This cost type associates the severity of the bottleneck to the sum of the timing violations for the violating/critical paths that traverse the instance.

Device Description

The device section contains general information about the design, including:

- Design name
- Family
- Die
- Package
- Software version

Bottleneck Analysis

This section lists the core of the bottleneck information. It is divided into two columns:

- Instance name: refers to the output pin name of the instance.
- Path Count: Displays the number of violating paths which include the instance pin.

See Also

Generating a Bottleneck Report

Timing Bottleneck Analysis Options Dialog Box

Generating a Datasheet Report

The datasheet reports information about the external characteristics of the design.

To generate a datasheet report, from the SmartTime Max/Min Delay Analysis View, choose **Tools > Reports > Datasheet**. The report appears in a separate window.

See Also

Understanding Datasheet Reports

Timing Datasheet Report Options Dialog Box

Understanding Datasheet Reports

The datasheet report displays the external characteristics of the design. . You can generate this report from SmartTime Max/Min Delay Analysis View. It contains three tables:

- Pin Description
- DC Electrical Characteristics
- AC Electrical Characteristics

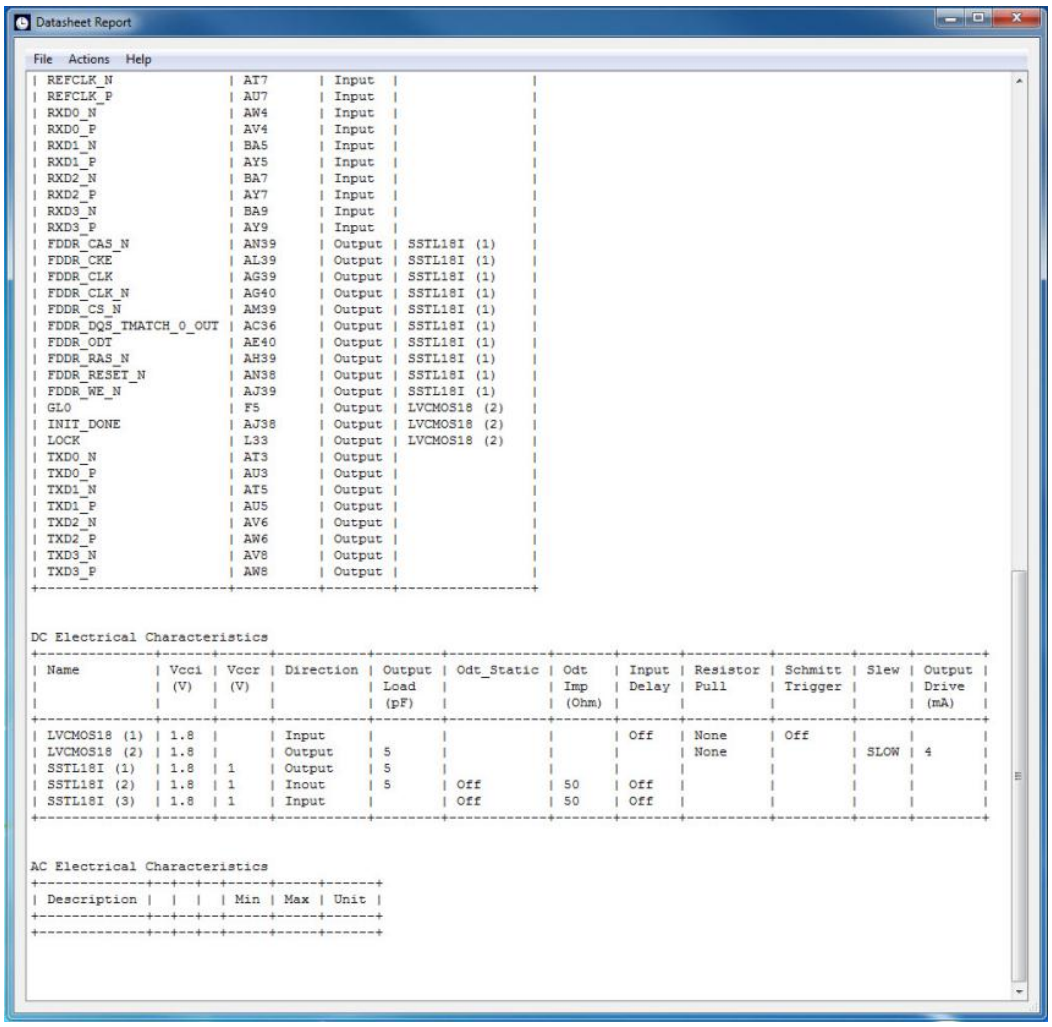


Figure 131 · Datasheet Report

Pin Description

Provides the port name in the netlist, location on the package, type of port, and I/O technology assigned to it. Types can be input, output, inout, or clock. Clock ports are ports shown as "clock" in the Clock domain browser.

DC Electrical Characteristics

Provides the parameters of the different I/O technologies used in the design. The number of parameters displayed depends on the family for which you have created the design.

AC Electrical Characteristics

Provides the timing properties of the ports of the design. For each clock, this section includes the maximum frequency. For each input, it includes the external setup, external hold, external recovery, and external removal for every clock where it applies. For each output, it includes the clock-to-out propagation time. This section also displays the input-to-output propagation time for combinational paths.

See Also

Generating a Datasheet Report

Timing Datasheet Report Options Dialog Box

Generating a Combinational Loop Report

The combinational loop report displays all loops found during initialization and reports pins associated with the loop(s), and the location where the loop is broken.

To generate a combinational loop report, from the Maximum (or Minimum) Delay Analysis **Tools** menu, choose **Reports > Combinational Loop**. The report appears in a separate window.

To generate the combinational loop report; from the **Tools** menu, choose **Reports > Combinational Loops ...**.

Select either the **Plain Text** or **Comma Separated Values** option in the Combinational_Loops Report Options dialog box and click **OK**.

The plain text report will pop up in a new window; you will be prompted to save the CSV in a directory of your choosing.

See Also

Understanding Combinational Loop Reports

Understanding Combinational Loop Reports

The combinational loop report displays all loops found during initialization and reports pins associated with the loop(s), and the location where the loop is broken.

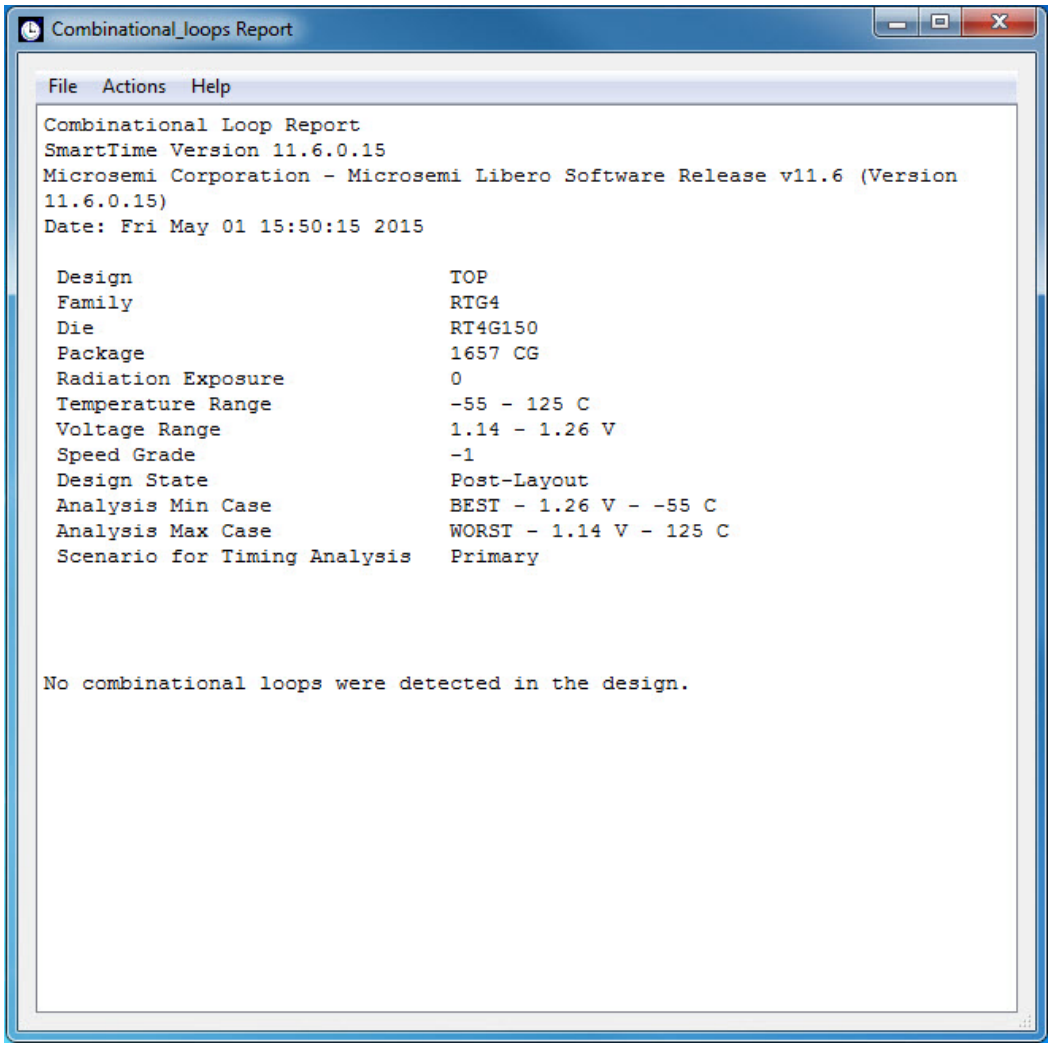


Figure 132 · Combinational Loop Report

To view a graphical representation of the broken loop, open MultiView Navigator, find each pin and add them as a logical cone. For more information on how to find each pin and how to set up the logical cones, refer to What is a LogicalCone.

See Also

Generating a Combinational Loop Report

Timing Concepts

Static Timing Analysis Versus Dynamic Simulation

Static timing analysis (STA) offers an efficient technique for identifying timing violations in your design and ensuring that it meets all your timing requirements. You can communicate timing requirements and timing exceptions to the system by setting timing constraints. A static timing analysis tool will then check and report setup and hold violations as well as violations on specific path requirements.

STA is particularly well suited for traditional synchronous designs. The main advantage of STA is that unlike dynamic simulation, it does not require input vectors. It covers all possible paths in the design and does all the above with relatively low run-time requirements. The major disadvantage of STA is that the STA tools do not automatically detect false paths in their algorithms.

Delay Models

The first step in timing analysis is the computation of single component delays. These components could be either a combinational gate or block or a single interconnect connecting two components.

Gates that are part of the library are pre-characterized with delays under different parameters, such as input-slew rates or capacitive loads. Traditional models provide delays between each pair of I/Os of the gate and between rising and falling edges.

The accuracy with which interconnect delays are computed depends on the design phase. These can be estimated using a simple Wire Load Model (WLM) at the pre-layout phase, or a more complex Resistor and Capacitor (RC) tree solver at the post-layout phase.

Timing Path Types

Path delays are computed by adding delay values across a chain of gates and interconnects. SmartTime uses this information to check for timing violations. Traditionally, timing paths are presented by static timing analysis tools in four categories or "sets":

- Paths between sequential components internal to the design. SmartTime displays this category under the Register to Register set of each displayed clock domain.
- Paths that start at input ports and end at sequential components internal to the design. SmartTime displays this category under the External Setup and External Hold sets of each displayed clock domain.
- Paths that start at sequential components internal to the design and end at output ports. SmartTime displays this category under the Clock to Out set of each displayed clock domain.
- Paths that start at input ports and end at output ports. SmartTime displays this category under the Input to Output set.

Maximum Clock Frequency

Generally, you set clock constraints on clocks for which you have a specified requirement. The absence of violations indicates that this clock will be able to run at least at the specified frequency. However, in the absence of such requirements, you may still be interested in computing the maximum frequency of a specific clock domain.

To obtain the maximum clock frequency, a static timing analysis tool computes the minimum period for each path between two sequential elements. To compute the maximum period, the tool evaluates the maximum data path delay and the minimum skew between the two elements, as well as the setup on the receiving sequential element. It also considers the polarity of each sequential element. The maximum frequency is the

inverse of the largest value among the maximum period of all the paths in the clock domain. The path responsible for limiting the frequency of a given clock is called the critical path.

Setup Check

The setup and hold check ensures that the design functions as specified at the required clock frequency.

Setup check specifies when data is required to be present at the input of a sequential component in order for the clock to capture this data effectively into the component. Timing analyzers evaluate the setup check as a maximum timing budget allowed between adjacent sequential elements. For more details on how setup check is processed, refer to Arrival Time, Required Time, and Slack.

See Also

Static Timing Analysis Versus Dynamic Simulation

Arrival Time, Required Time, and Slack

Arrival Time, Required Time and Slack

You can use arrival time and required time to verify timing requirements in the presence of constraints. Below is a simple example applied to verifying the clock requirement for setup between sequential elements in the design.

The arrival time represents the time at which the data arrives at the input of the receiving sequential element. In this example, the arrival time is considered from the setup launch edge at CK, taken as a time reference (instant zero). It follows the clock network along the blue line until the clock pin on FF1 (delay d1). Then it continues along the data path always following the blue line until the data pin D on FF2. Therefore,

$Arrival\_Time_{FF2:D} = d1 + d2$

The required time represents when the data is required to be present at the same pin FF2:D. Assume in this example that in the presence of an FF with the same polarity, the capturing edge is simply one cycle following the launch edge. Using the period T provided to the tool through the clock constraint, the event gets propagated through the clock network along the red line until the clock pin of FF2 (delay d3). Taking into account FF2 setup (delay d4), this means that the clock constraint requires the data to be present d4 time before the capturing clock edge on FF2. Therefore, the required time is:

$Required\_Time_{FF2:D} = T + d3 - d4$

The slack is simply the difference between the required time and arrival time:

$Slack_{FF2:D} = Required\_Time_{FF2:D} - Arrival\_TimeFF2:D$

If the slack is negative, the path is violating the setup relationship between the two sequential elements.

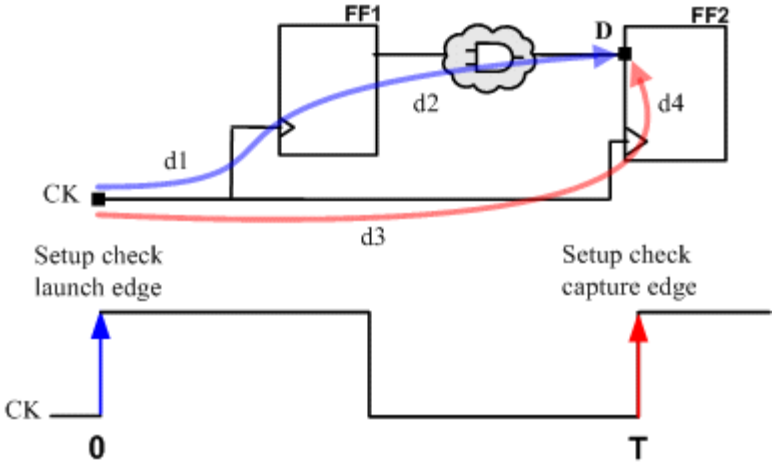


Figure 133 · Arrival Time and Required Time for Setup Check

Timing Exceptions Overview

Use timing exceptions to overwrite the default behavior of the design path. Timing exceptions include:

- Setting multicycle constraint to specify paths that (by design) will take more than one cycle.
- Setting a false path constraint to identify paths that must not be included in the timing analysis or the optimization flow.
- Setting a maximum delay constraint on specific paths to relax or to tighten the original clock constraint requirement.

See Also

Specifying a Maximum Delay Constraint

Specifying a Minimum Delay Constraint

Specifying a Multicycle Constraint

Specifying a False Path Constraint

Changing Output Port Capacitance

Clock Skew

The clock skew between two different sequential components is the difference between the insertion delays from the clock source to the clock pins of these components. SmartTime calculates the arrival time at the clock pin of each sequential component. Then it subtracts the arrival time at the receiving component from the arrival time at the launching component to obtain an accurate clock skew.

Both setup and hold checks must account for clock skew. However, for setup check, SmartTime looks for the smallest skew. This skew is computed by using the maximum insertion delay to the launching sequential component and the shortest insertion delay to the receiving component.

For hold check, SmartTime looks for the largest skew. This skew is computed by using the shortest insertion delay to the launching sequential component and the largest insertion delay to the receiving component. SmartTime makes this distinction automatically.

Add Input Delay Constraint Dialog Box

Use this dialog box to apply input delay constraints or external setup/hold constraints. This constraint defines the arrival time of an input relative to a clock.

To open the Add Input Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor Constraints menu choose **Input Delay** (**Constraints > Input Delay**).

External Setup/Hold

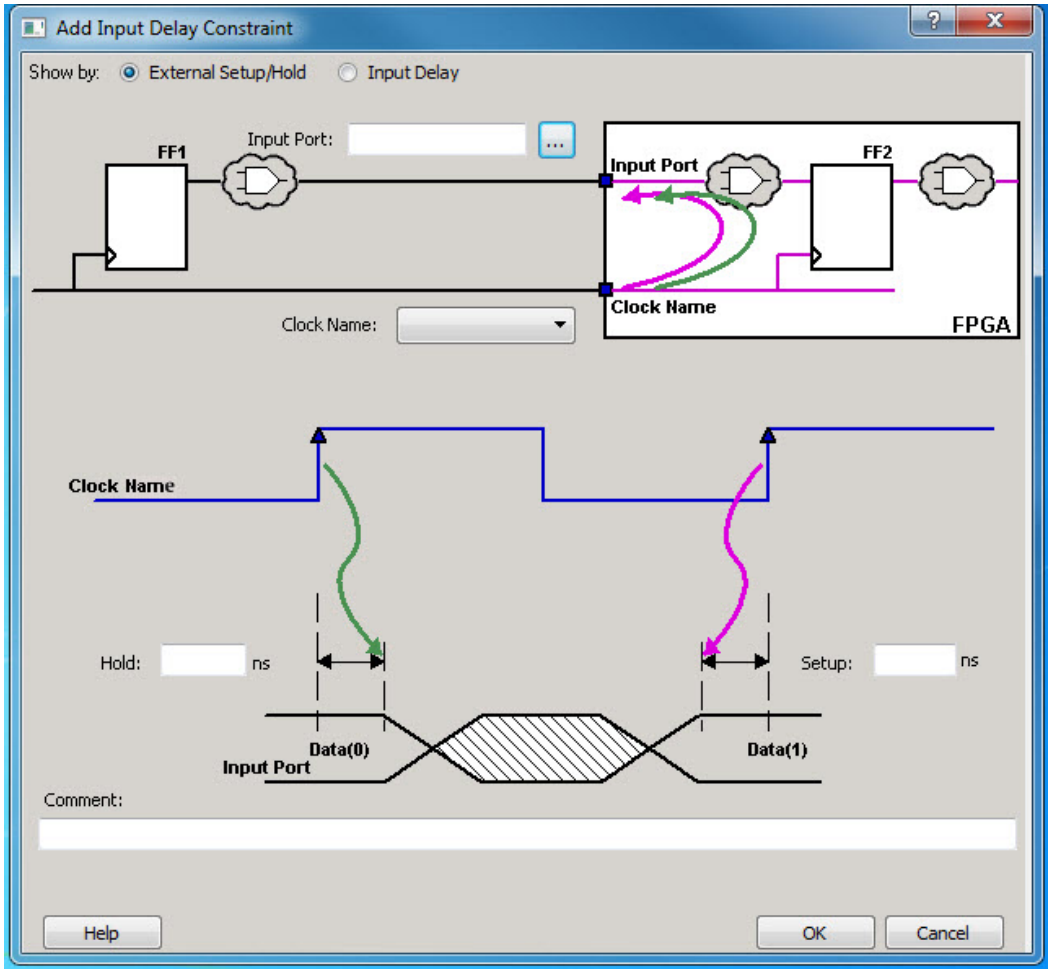


Figure 134 · Set Input Delay Constraint (Show by: External Setup/Hold) Dialog Box

Input Port

Specifies a list of input ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Name

Specifies the clock reference to which the specified External Setup/Hold is related.

External Hold

Specifies the external hold time requirement for the specified input ports.

External Setup

Specifies the external setup time requirement for the specified input ports.

Comment

Enables you to provide comments for this constraint.

Input Delay

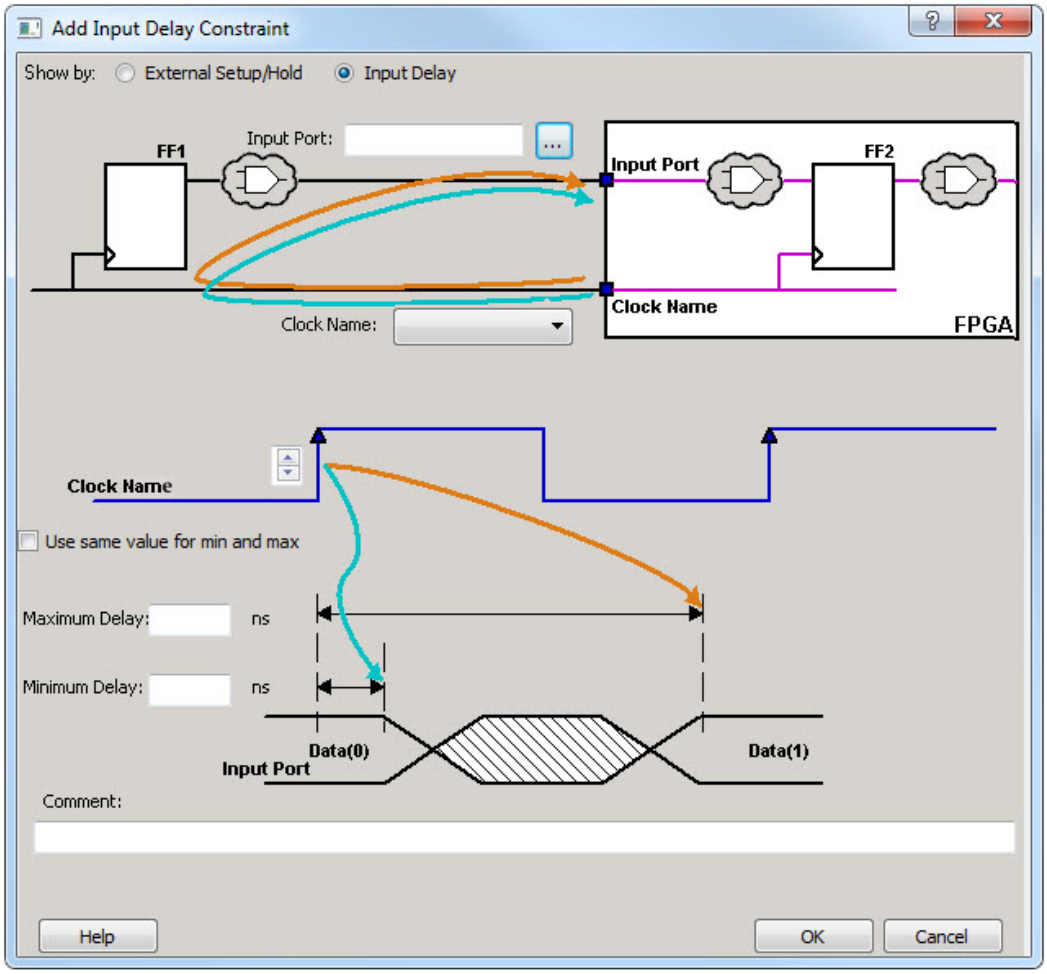


Figure 135 · Set Input Delay Constraint (Show by: Input Delay) Dialog Box

Input Port

Specifies a list of input ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Name

Specifies the clock reference to which the specified input delay is related.

Clock edge

Indicates the launching edge of the clock (rising or falling).

Use same delay for both min and max

Specifies that the minimum input delay uses the same value as the maximum input delay.

Maximum Delay

Specifies that the delay refers to the longest path arriving at the specified input.

Minimum Delay

Specifies that the delay refers to the shortest path arriving at the specified input.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying Input Timing Delay Constraint

Add Output Delay Constraint Dialog Box

Use this dialog box to apply output delay constraints. This constraint defines the output delay of an output relative to a clock.

To open the Set Output Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Output Delay**.

Clock-to-Output

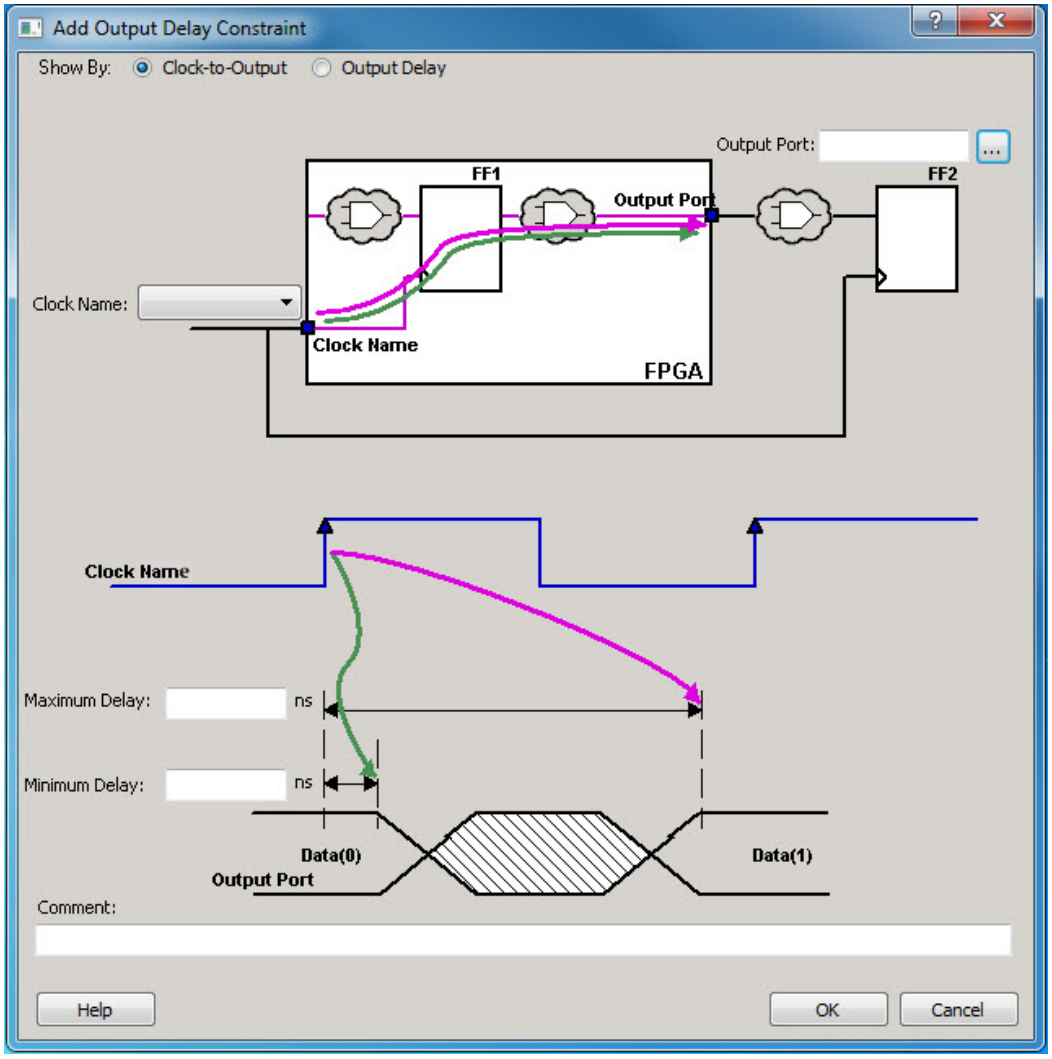


Figure 136 · Add Output Delay (Show By: Clock-to-Output) Dialog Box

Output Port

Specifies a list of output ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Name

Specifies the clock reference to which the specified Clock-to-Output is related.

Maximum Delay

Specifies the delay for the longest path from the clock port to the output port. This constraint includes the combinational path delay from output of the launched edge to the output port.

Minimum Delay

Specifies the delay for the shortest path from the clock port to the output port. This constraint includes the combinational path delay from output of the launched edge to the output port.

Comment

Enables you to provide comments for this constraint.

Output Delay

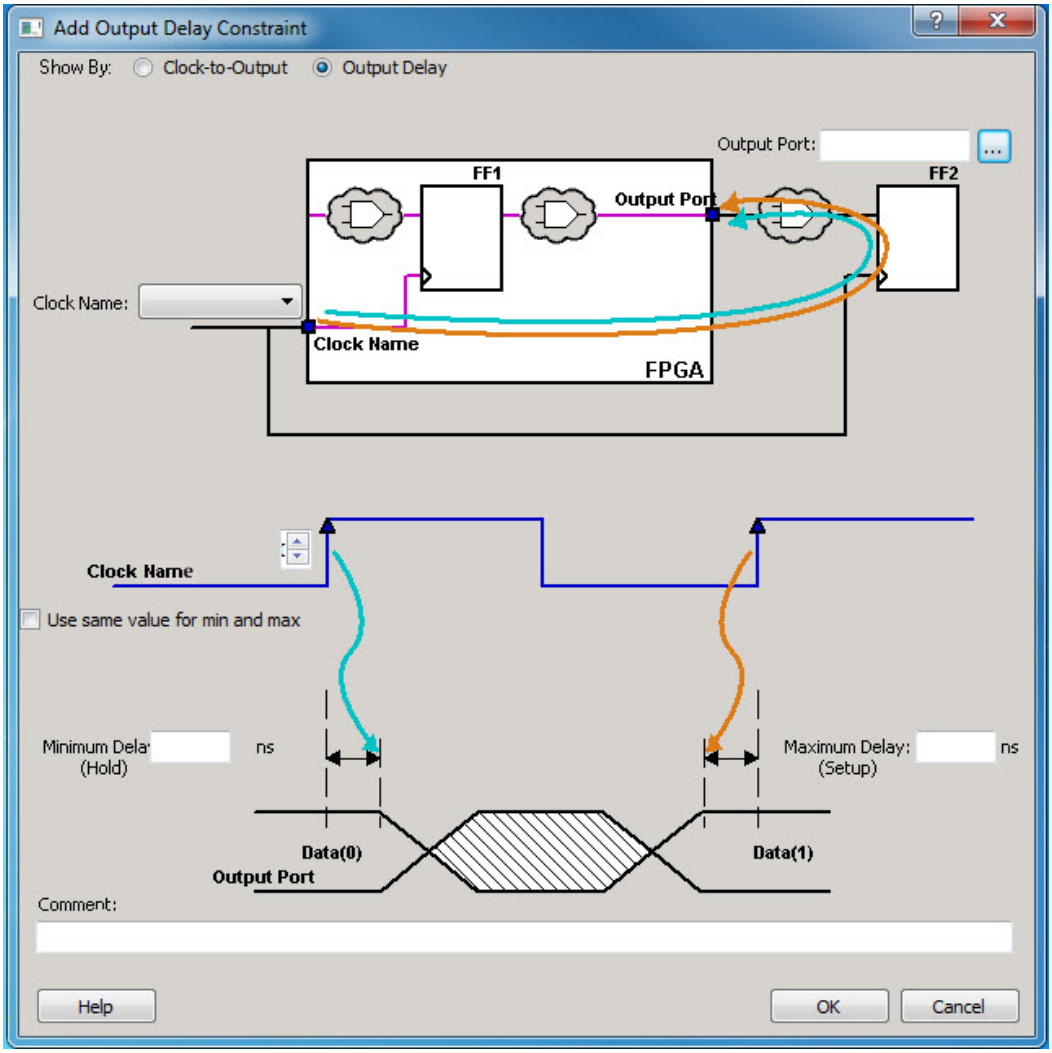


Figure 137 · Set Output Delay (Show By: Output Delay) Dialog Box

Output Port

Specifies a list of output ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Name

Specifies the clock reference to which the specified output delay is related.

Clock edge

Indicates the launching edge of the clock (rising or falling).

Maximum Delay

Specifies the delay for the longest path from the specified output to the captured edge. This represents a combinational path delay to a register outside the current design plus the library setup time.

Minimum Delay

Specifies the delay for the shortest path from the specified output to the captured edge. This represents a combinational path delay to a register outside the current design plus the library hold time.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying Output Timing Delay Constraint

Dialog Boxes

Add Path Analysis Set Dialog Box

Use this dialog box to specify a custom path analysis set.

Note: The Analysis menu is available only in Maximum or Minimum Delay Analysis view.

To open the Add Path Analysis Set dialog box (shown below) from the SmartTime Max/Min Delay Analysis View, right-click a path group in the Domain Browser and select **Add Set**.

Tip: You can also click the  icon in the SmartTime window bar to display the **Add Path Analysis Set** dialog box.

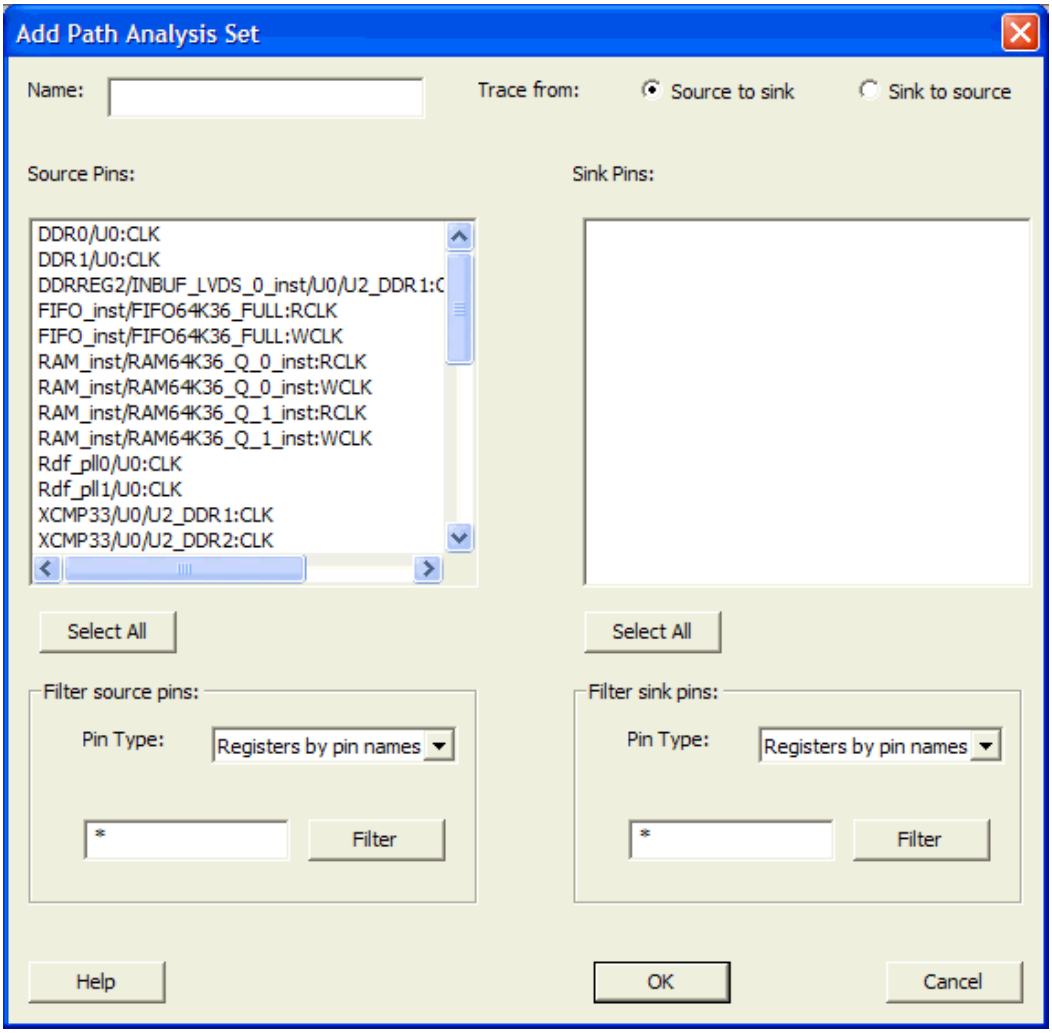


Figure 138 · Add Path Analysis Set Dialog Box

Name

Enter the name of your path set.

Trace from

Select whether you want to trace connected pins from **Source to sink** or from **Sink to source**. By default, the pins are traced Source to sink.

Source Pins

Displays a list of available and valid source pins. You can select multiple pins. To select all source pins, click the **Select All** button beneath the Source Pins list.

Select All

Selects all the pins in the Source Pins list to include in the path analysis set.

Filter Source Pins

Enables you to specify thesource **Pin Type** and the **Filter**. The default pin type is Registers by pin name. You can specify any string value for the **Filter**. If you change the pin type, the **Source Pins** shows the updated list of available source pins.

Sink Pins

Displays list of available and valid pins. You can select multiple pins. To select all source pins, click the **Select All** button beneath the **Sink Pins list.**

Select All

Selects all the pins in the Sink Pins list to include in the path analysis set.

Filter Sink Pins

Enables you to specify the sink **Pin Type** and the **Filter**. The default pin type is Registers (by pin). You can specify any string value for the **Filter**. If you change the pin type, the **Sink Pins** shows the updated list of available sink pins.

Analysis Set Properties Dialog Box

Use this dialog box to view information about the user created set.

To open the **Analysis Set Properties** dialog box (shown below) from the Timing Analysis View, right-click any user-created set in the Domain Browser, and choose **Properties** from the shortcut menu.

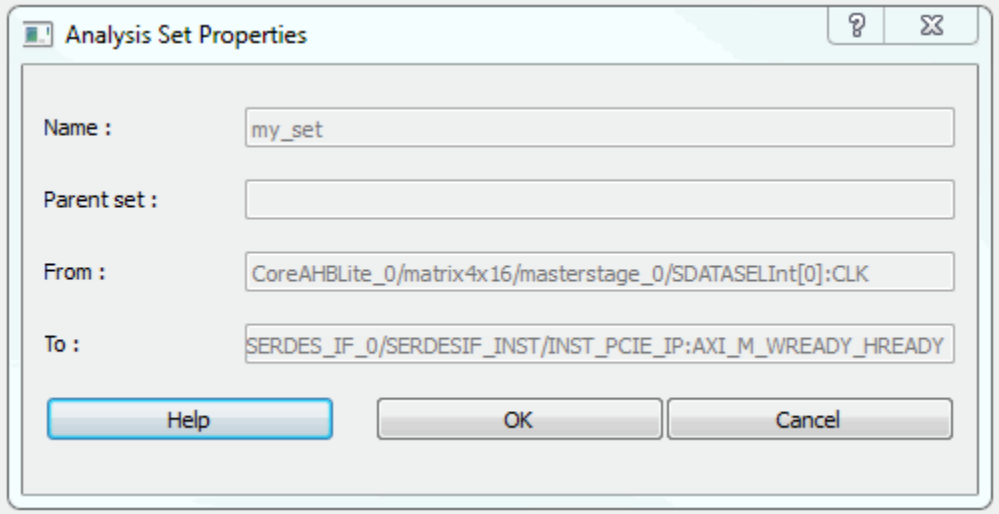


Figure 139 · Analysis Set Properties Dialog Box

Name

Specifies the name of the user-created path set.

Parent Set

Specifies the name of the parent path set to which the user-created path set belongs.

Creation filter

From

Specifies a list of source pins in the user-created path set.

To

Specifies a list of sink pins in the user-created path set.

See Also

Using filters

Edit Filter Set Dialog Box

Use this dialog box to specify a filter.

To open the **Edit Filter Set** dialog box (shown below) from the SmartTime Max/Min Delay Analysis view, right-click an existing filter set in the clock domain browser, and then choose **Edit Set** from the shortcut menu.

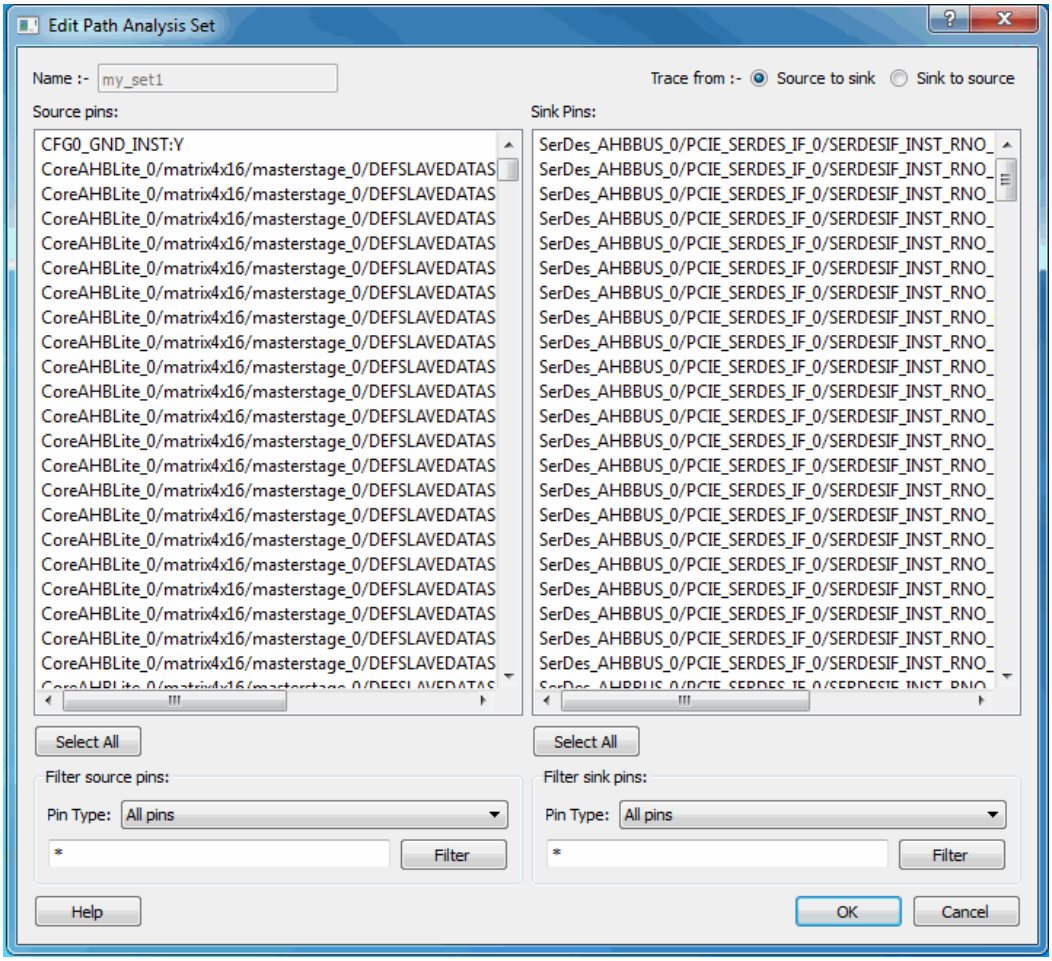


Figure 140 · Edit Path Analysis Set Dialog Box

Name

Specifies the name of the path you want to edit.

Creation filter

Source Pins - Displays a list of source pins in the user-created path set.

Sink Pins - Displays a list of sink pins in the user-created path set.

See Also

Using filters

Select Source Pins for Clock Constraint Dialog Box

Use this dialog box to find and choose the clock source from the list of available pins.

To open the **Select Source Pins for the Clock Constraint** dialog box (shown below) from the SmartTime Constraints Editor, click the **Browse** button to the right of the Clock source field in the **Create Clock Constraint** dialog box.

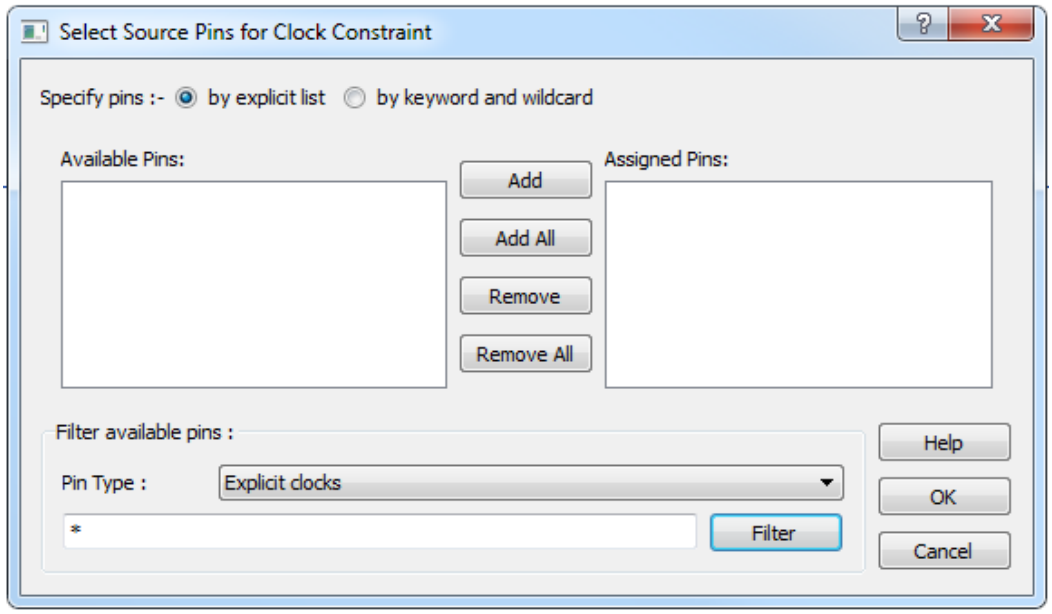


Figure 141 · Select Source Pins for Clock Constraint Dialog Box

Available Pins

Displays all available pins.

Filter Available Pins

Explicit clock pins for the design is the default value. To identify any other pins in the design as clock pins, right-click the **Pin Type** pull-down menu and select one of the following:

- Explicit clocks
- Potential clocks
- Input ports
- All Pins
- All Nets
- Pins on clock network
- Nets in clock network

You can also use the **Filter** to filter the clock source pin name in the displayed list

See Also

Specifying clock constraints

Create Clock Constraint Dialog Box

Use this dialog box to enter a clock constraint setting.

It displays a typical clock waveform with its associated clock information. You can enter or modify this information, and save the final settings as long as the constraint information is consistent and defines the clock waveform completely. The tool displays errors and warnings if information is missing or incorrect.

To open the Create Clock Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Clock**.

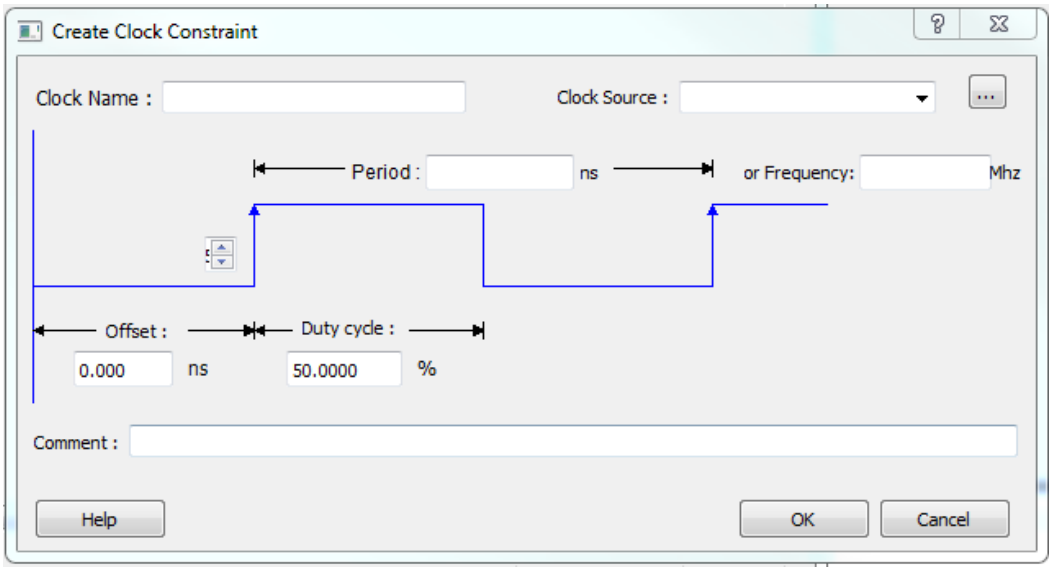


Figure 142 · Create Clock Constraint Dialog Box

Clock Source

Enables you to choose a pin from your design to use as the clock source.

The drop-down list is populated with all explicit clocks. You can also select the Browse button to access all potential clocks. The **Browse** button displays the Select Source Pins for Clock Constraint Dialog Box.

Clock Name

Specifies the name of the clock constraint. This field is required for virtual clocks when no clock source is provided.

T(zero) Label

Instant zero used as a common starting time to all clock constraints.

Period

When you edit the period, the tool automatically updates the frequency value.

The period must be a positive real number. Accuracy is up to 3 decimal places.

Frequency

When you edit the frequency, the tool automatically updates the period value.

The frequency must be a positive real number. Accuracy is up to 3 decimal places.

Offset (Starting Edge Selector)

Enables you to switch between rising and falling edges and updates the clock waveform.

If the current setting of starting edge is rising, you can change the starting edge from rising to falling.

If the current setting of starting edge is falling, you can change the starting edge from falling to rising.

Duty Cycle

This number specifies the percentage of the overall period that the clock pulse is high.

The duty cycle must be a positive real number. Accuracy is up to 4 decimal places. Default value is 50%.

Offset

The offset must be a positive real number. Accuracy is up to 2 decimal places. Default value is 0.

Comment

Enables you to save a single line of text that describes the clock constraints purpose.

See Also

create_clock (SDC)

Clock definition

Specifying Clock Constraints

Create Generated Clock Constraint Dialog Box

Use this dialog box to specify generated clock constraint settings.

It displays a relationship between the clock source and its reference clock. You can enter or modify this information, and save the final settings as long as the constraint information is consistent. The tool displays errors and warnings if the information is missing or incorrect.

To open the Create Generated Clock Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Generated Clock**.

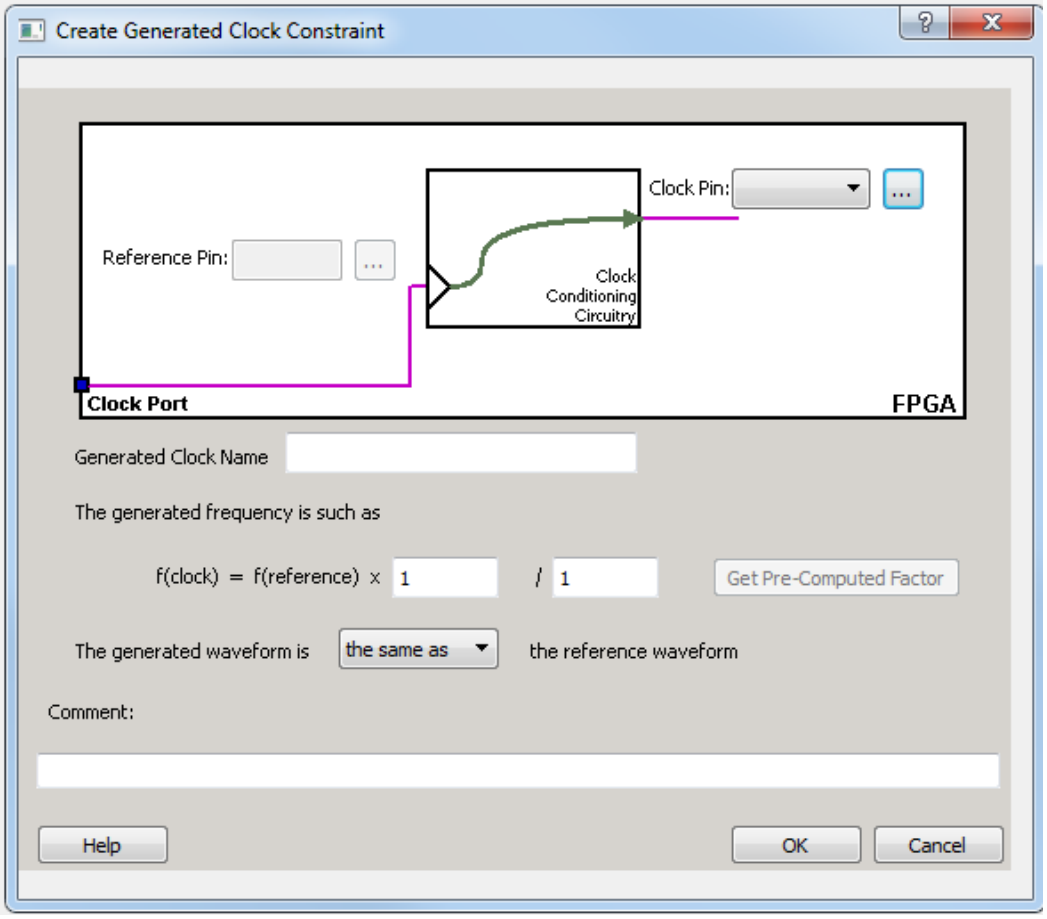


Figure 143 · Create Generated Clock Constraint

Clock Pin

Enables you to choose a pin from your design to use as a generated clock source.

The drop-down list is populated with all unconstrained explicit clocks. You can also select the Browse button to access all potential clocks and pins from the clock network. The Browse button displays the Select Generated Clock Source dialog box.

Reference Pin

Enables you to choose a pin from your design to use as a generated reference pin.

Generated Clock Name

Specifies the name of the clock constraint. This field is required for virtual clocks when no clock source is provided.

Generated Frequency

The generated frequency is a factor of reference frequency defined with a multiplication element and/or a division element.

Generated Waveform

The generated waveform could be either the same as or inverted w.r.t. the reference waveform.

Comment

Enables you to save a single line of text that describes the generated clock constraints purpose.

See Also

create_generated_clock (SDC)

Specifying Generated Clock Constraints

Select Generated Clock Source

Customize Analysis View Dialog Box

Use this dialog box to customize the timing analysis grid.

To open the **Customize Analysis View** dialog box (shown below) from the SmartTime Max/Min Delay Analysis View, click the **Customize table** button (circled in red in the figure below) in the Max/Min Delay Analysis View.

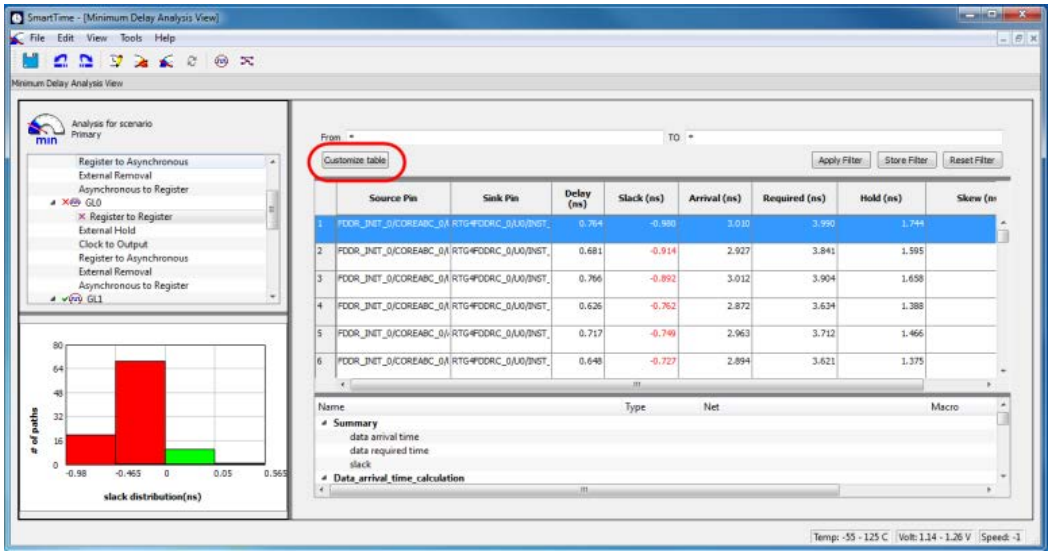


Figure 144 · Customize Table Button

The Customize Paths List Table Dialog Box appears.

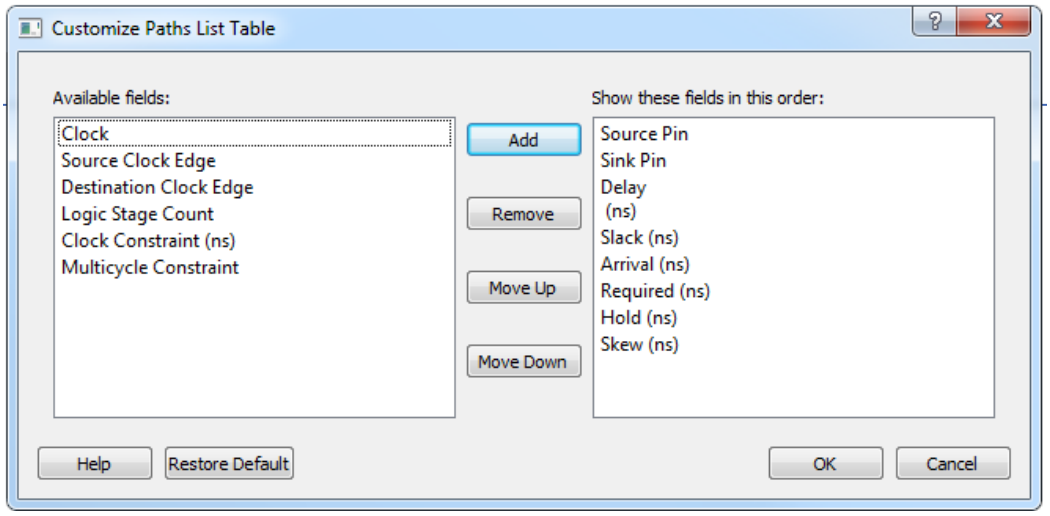


Figure 145 · Customize Paths List Dialog Box

Available Fields

Displays a list of all the available fields in the timing analysis grid.

Show These Fields in This Order

Shows the list of fields you want to see in the timing analysis grid. Use **Add** or **Remove** to move selected items from **Available fields** to **Show these fields in this order** or vice versa. You can change the order in which these fields are displayed by using **Move Up** or **Move Down**.

Restore Defaults

Resets all the options in the General panel to their default values.

Manage Clock Domains Dialog Box

Use this dialog box to specify the clock pins you want to see in the Expanded Path view.

To open the Manage Clock Domain dialog box (shown below) from the SmartTime Max/Min Delay Analysis view, click the  icon.

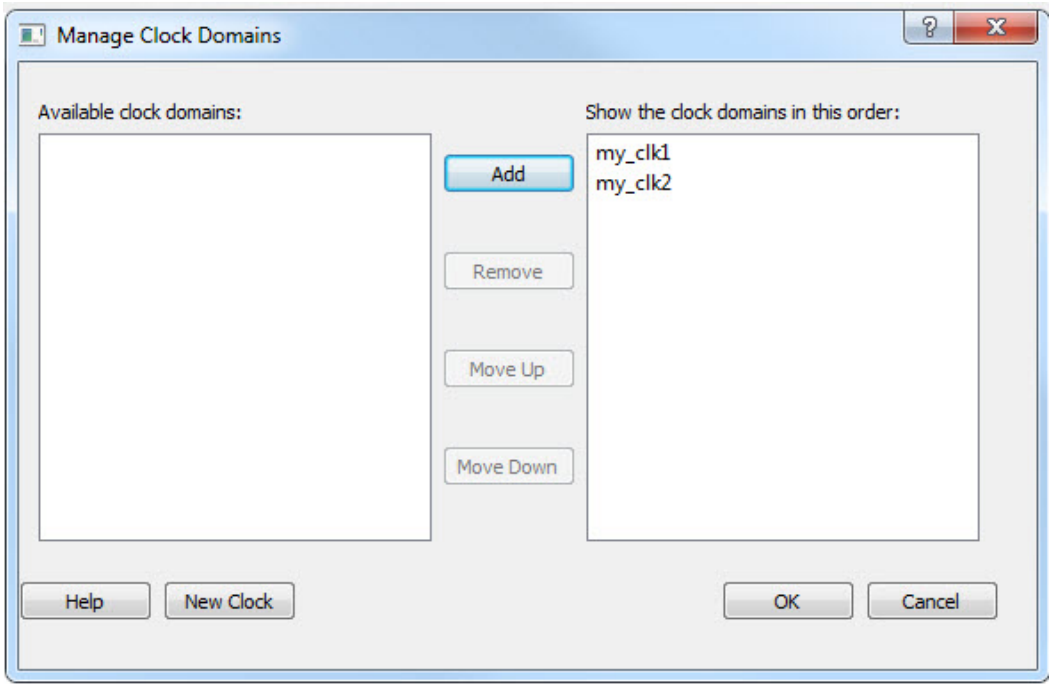


Figure 146 · Manage Clock Domains Dialog Box

Available Clock Domains

Displays alphanumerically sorted list of available clock pins. The first clock pin is selected by default.

Show the Clock Domains in this Order

Shows the clock pins you want to see in the Expanded Path view. Use **Add** or **Remove** to move selected items from **Available clock domains** to **Show the clock domains in this order** or vice versa. You can change the order in which these clock pins are displayed by using **Move Up** or **Move Down**.

New Clock

Invokes the Select Source Pins for Clock Constraint dialog box. The new clock gets added at the end of the **Show the clock domains in this order** list box.

See Also

Managing Clock Domains

Select Generated Clock Reference Dialog Box

Use this dialog box to find and choose the generated clock reference pin from the list of available pins.

To open the Select Generated Clock Reference dialog box (shown below) from the SmartTime Constraints Editor, open the Create Generated Clock Constraint Dialog Box dialog box and click the **Browse** button for the **Clock Reference**.

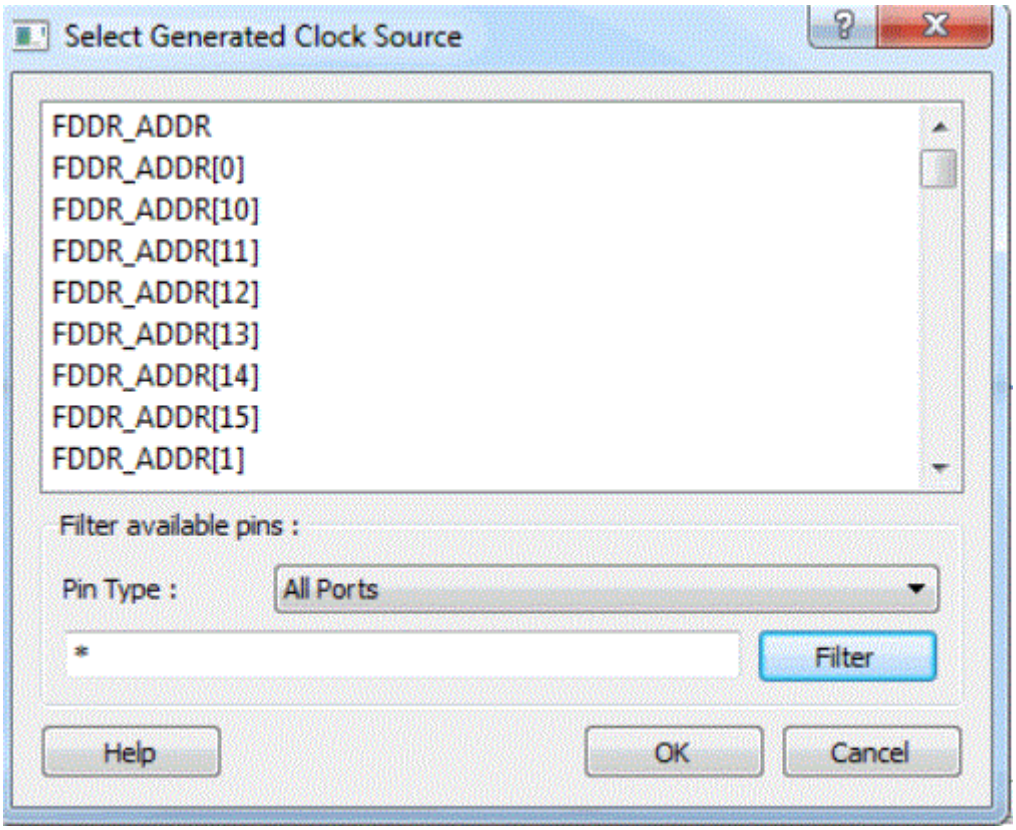


Figure 147 · Select Generated Clock Reference Dialog Box

Filter Available Pins

To identify any other pins in the design as the generated master pin, select **Filter available objects - Type** as **Clock Network**. You can also use the **Filter** to filter the generated reference clock pin name in the displayed list.

See Also

Specifying generated clock constraints

Select Generated Clock Source Dialog Box

Use this dialog box to find and choose the generated clock source from the list of available pins.

To open the Select Generated Clock Source dialog box (shown below) from the SmartTime Constraints Editor, open the Create Generated Clock Constraint dialog box and click the **Browse** button for the **Clock Pin**.

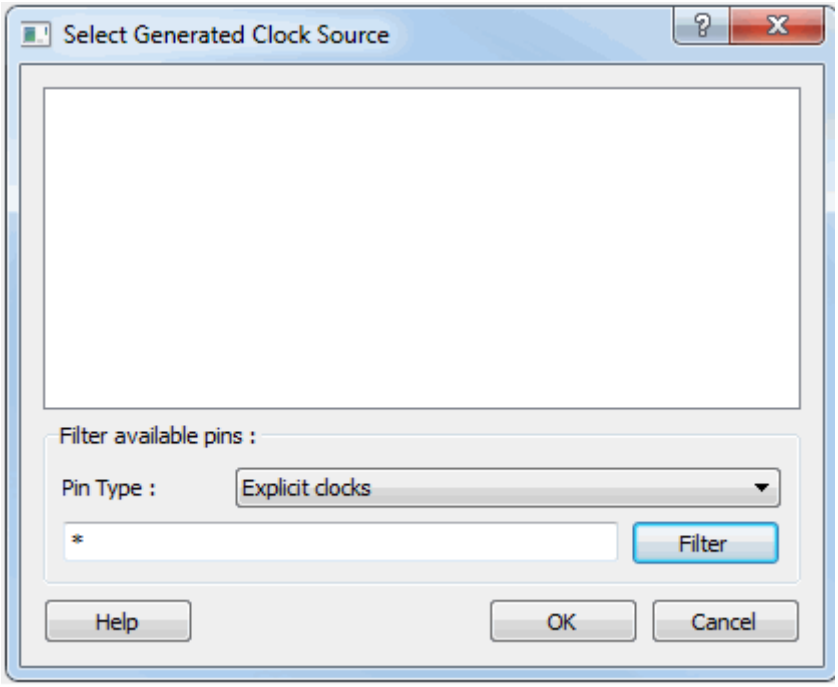


Figure 148 · Select Generated Clock Source Dialog Box

Filter Available Pins

Explicit clock pins for the design is the default value. To identify any other pins in the design as the generated clock source pins, from the **Pin Type** pull-down list, select **Explicit clocks**, **Potential clocks**, **All Ports**, **All Pins**, **All Nets**, **Pins on clock network**, or **Nets in clock network**. You can also use the **Filter** to filter the generated clock source pin name in the displayed list.

See Also

Specifying generated clock constraint (SDC)

Select Source or Destination Pins for Constraint Dialog Box

Use this dialog box to select pins or ports:

- By explicit list
- By keyword and wildcard

To open the Select Source or Destination Pins for Constraint dialog box from the SmartTime Constraints Editor, right-click the Constraint Type in the Contraint Browser to open the Add Constraint Dialog Box. From this dialog box, click the **Browse** button to open the Select Source or Destination Pins for the Constraint dialog box.

By Explicit List

This is the default. This mode stores the actual pin names. The following figure shows an example dialog box for **Select Source Pins for Multicycle Constraint** > **by explicit list**.

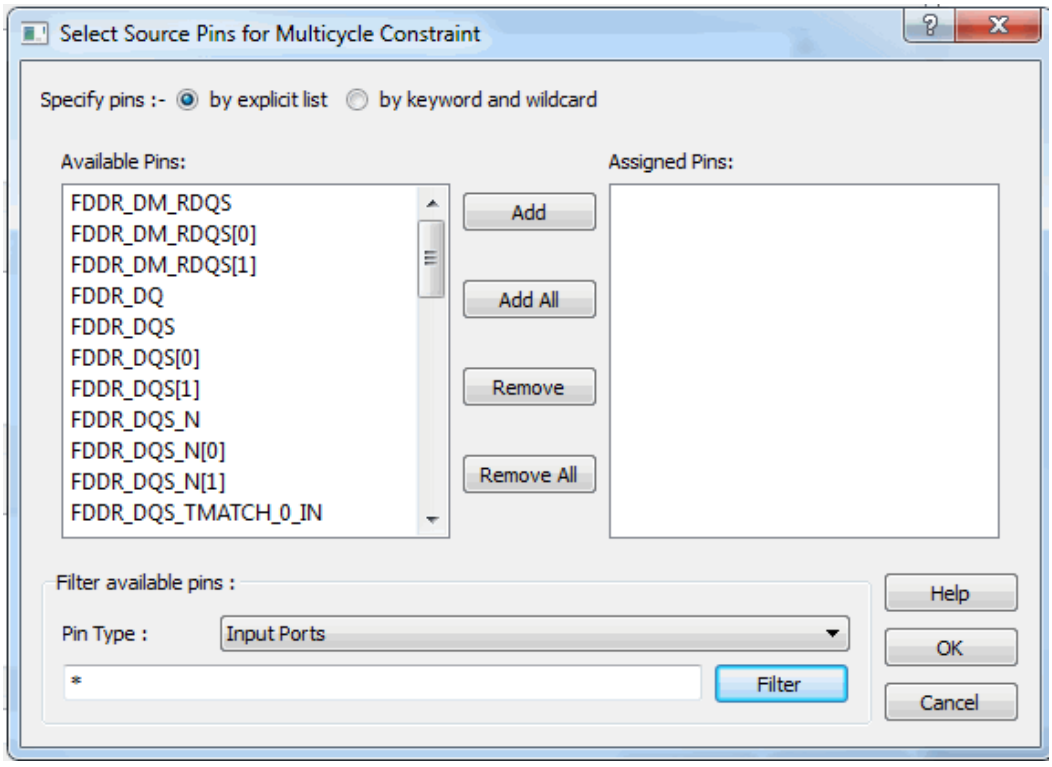


Figure 149 · Select Source Pins for Multicycle (Specify pins by explicit list) Dialog Box

Available Pins

The list box displays the available valid objects. If you change the filter value, the list box shows the available objects based on the filter.

Use **Add**, **Add All**, **Remove**, **Remove All** to add or delete pins from the **Assigned Pins** list.

Filter Available Pins

Pin type – Specifies the filter on the available object. This can be by **Explicit clocks**, **Potential clocks**, **Input ports**, **All Pins**, **All Nets**, **Pins on clock network**,or **Nets in clock network**

Filter

Specifies the filter based on which the **Available Pins** list shows the pin names. The default is *. You can specify any string value.

By Keyword and Wildcard

This mode stores the filter only. It does not store the actual pin names. The constraints created using this mode get exported with the SDC accessors (get_ports, get_pins, etc.). The following figure shows an example dialog box for **Select Source Pins for Multicycle Constraint** > **by keyword and wildcard**.

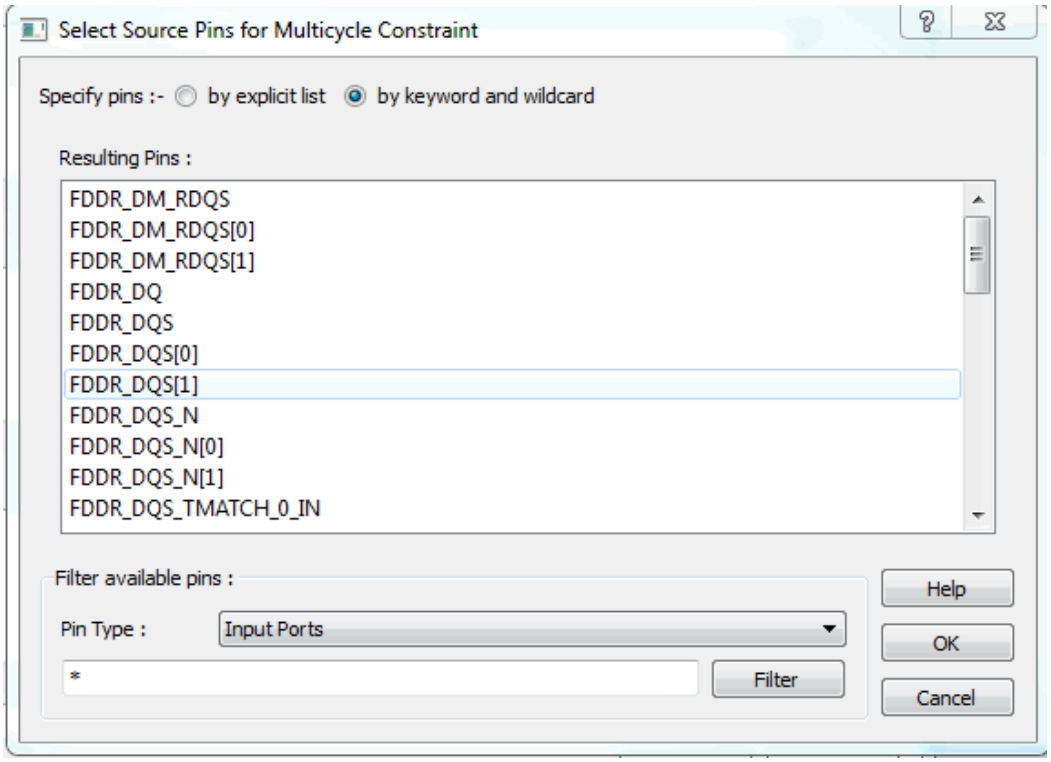


Figure 150 · Select Source Pins for Multicycle (Specify pins by keyword and wildcard) Dialog Box

Pin Type

Specifies the filter on the available pins. This can be Registers by pin names, Registers by clock names, Input Ports, or Output Ports. The default pin type is Registers by pin names.

Filter

Specifies the filter based on which the Available Pins list shows the pin names. The default is *. You can specify any string value.

Resulting Pins

Displays pins from the available pins based on the filter.

Set False Path Constraint Dialog Box

Use this dialog box to define specific timing paths as being false.

This constraint removes timing requirements on these false paths so that they are not considered during the timing analysis. The path starting points are the input ports or register clock pins and path ending points are the register data pins or output ports. This constraint disables setup and hold checking for the specified paths.

Note: The false path information always takes precedence over multiple cycle path information and overrides maximum delay constraints.

To open the Set False Path Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > False Path**.

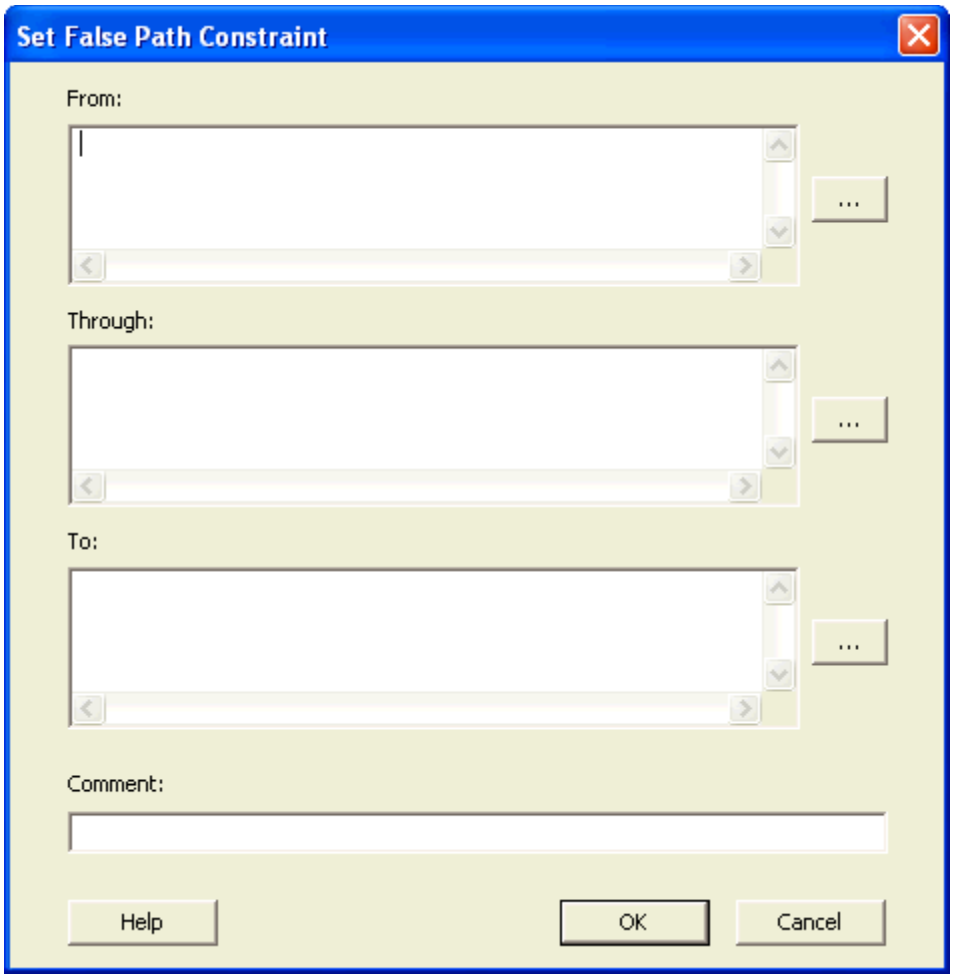


Figure 151 · Set False Path Constraint Dialog Box

From

Specifies the starting points for false path. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

Through

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

To

Specifies the ending points for false path. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying a False Path Constraint

Set Clock Source Latency Dialog Box

Use this dialog box to define the delay between an external clock source and the definition pin of a clock within SmartTime.

To open the Set Clock Source Latency dialog box (shown below) from the Timing Analysis View, you must first create a clock constraint. From the **Constraints** menu, choose **Clock Source Latency**.

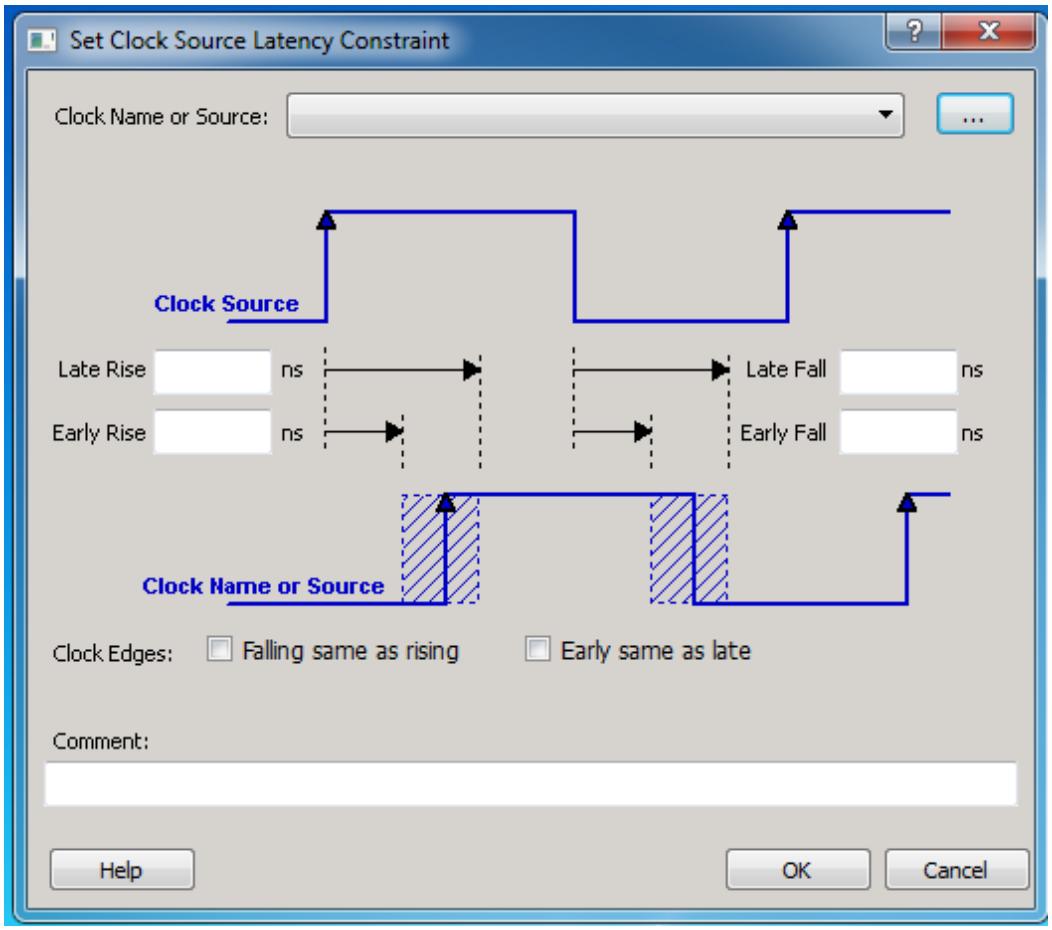


Figure 152 · Set Clock Source Latency Dialog Box

Clock Name or Source

Displays a list of clock ports or pins that do not already have a clock source latency specified. Select the clock name or source for which you are specifying the clock source latency.

Late Rise

Specifies the largest possible latency, in nanoseconds, of the rising clock edge at the clock port or pin selected, with respect to its source. Negative values are acceptable, but may lead to overly optimistic analysis.

Early Rise

Specifies the smallest possible latency, in nanoseconds, of the rising clock edge at the clock port or pin selected, with respect to its source. Negative values are acceptable, but may lead to overly optimistic analysis.

Late Fall

Specifies the largest possible latency, in nanoseconds, of the falling clock edge at the clock port or pin selected, with respect to its source. Negative values are acceptable, but may lead to overly optimistic analysis.

Early Fall

Specifies the smallest possible latency, in nanoseconds, of the falling clock edge at the clock port or pin selected, with respect to its source. Negative values are acceptable, but may lead to overly optimistic analysis.

Clock Edges

Select the latency for the rising and falling edges:

Falling same as rising: Specifies that Rising and Falling clock edges have the same latency.

Early same as late : Specifies that the clock source latency should be considered as a single value, not a range from "early" to '"late".

Comment

Enables you to save a single line of text that describes the clock source latency.

See Also

Specifying Clock Constraints

Set Constraint to Disable Timing Arcs Dialog Box

Use this dialog box to specify the timing arcs being disabled to fix the combinational loops in the design.

To open the Set Constraint to Disable Timing Arcs dialog box (shown below) from the Timing Analysis View, from the **Constraints** menu, choose **Disable Timing**.

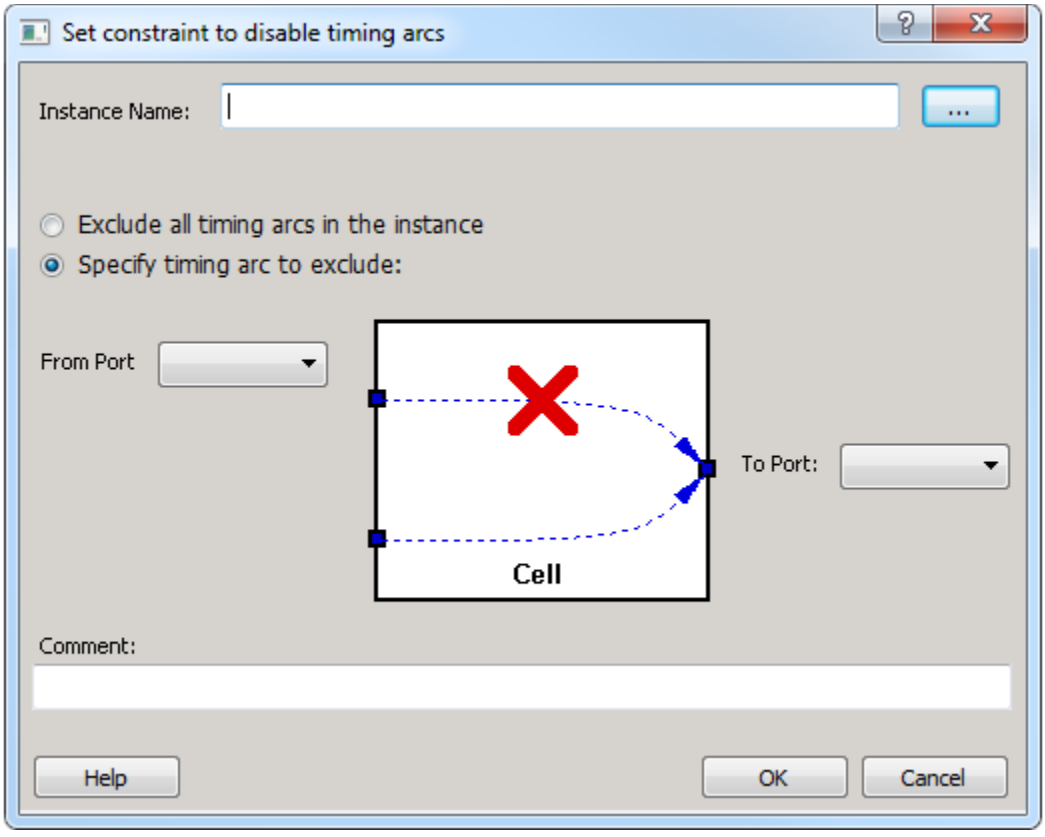


Figure 153 · Set Constraint to Disable Timing Arcs Dialog Box

Instance Name

Specifies the instance name for which the disable timing arc constraint will be created.

Exclude All Timing Arcs in the Instance

This option enables you to exclude all timing arcs in the specified instance.

Specify Timing Arc to Exclude

This option enables you to specify the timing arc to exclude. In this case, you need to specify the from and to ports:

From Port

Specifies the starting point for the timing arc.

To Port

Specifies the ending point for the timing arc.

Comment

Enables you to save a single line of text that describes the disable timing arc.

See Also

Specifying Disable Timing Constraint

Set Clock-to-Clock Uncertainty Constraint Dialog Box

Use this dialog box to model tracking jitter between two clocks in your design.

To open the Set Clock-to-Clock Uncertainty Constraint dialog box (shown below), from the **Constraints** menu, choose **Clock-to-Clock Uncertainty**.

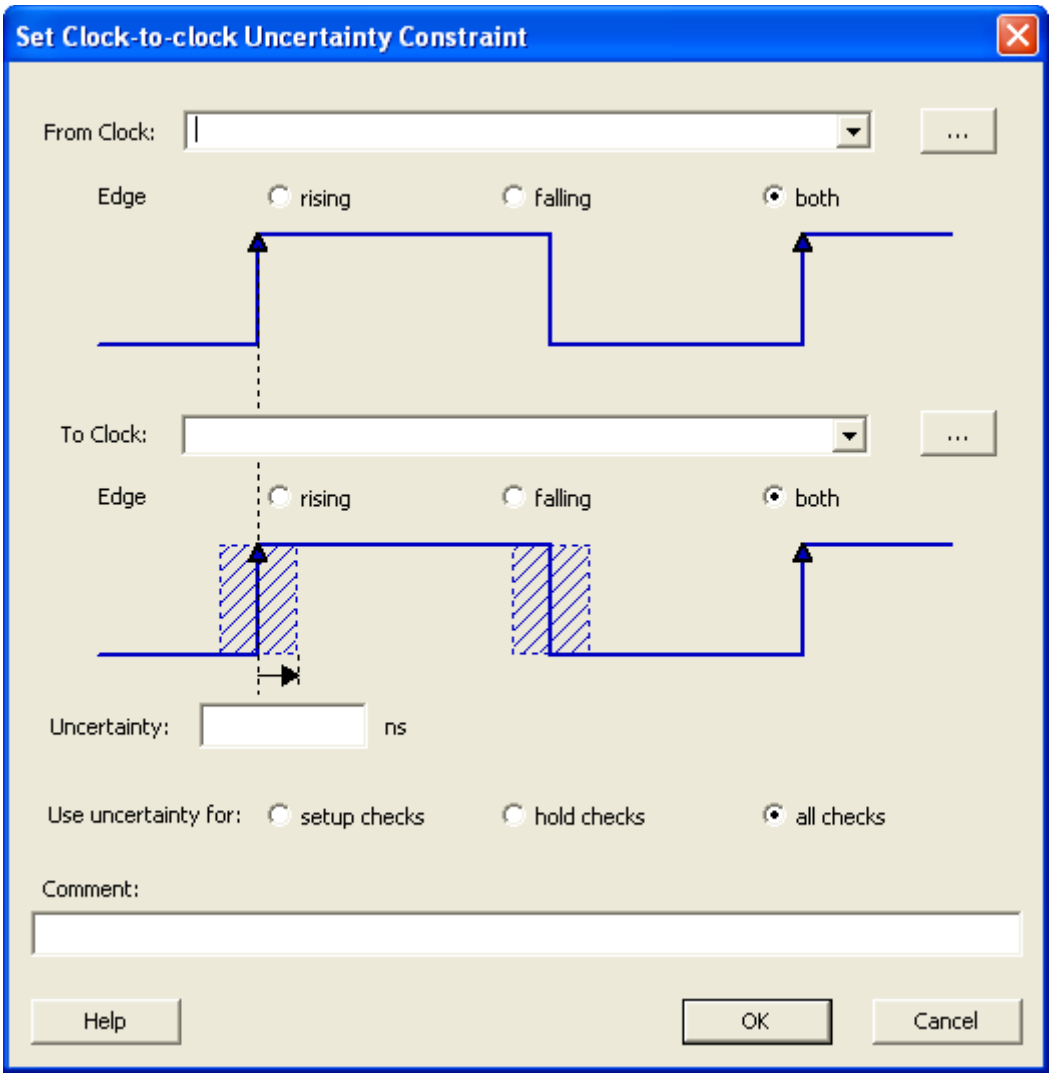


Figure 154 · Clock-to-Clock Uncertainty Constraint Dialog Box

From Clock

Specifies clock name as the uncertainty source.

Edge

This option enables you to select if the clock-to-clock uncertainty applies to rising, falling, or both edges.

To Clock

Specifies clock name as the uncertainty destination.

Edge

This option enables you to select if the clock-to-clock uncertainty applies to rising, falling, or both edges.

Uncertainty

Enter the time in ns that represents the amount of variation between two clock edges.

Use Uncertainty For

This option enables you select whether the uncertainty constraint applies to setup, hold, or all checks.

Comment

Enables you to save a single line of text that describes this constraint.

See Also

Specifying Disable Timing Constraint

Set Input Delay Constraint Dialog Box

Use this dialog box to apply input delay constraints or external setup/hold constraints. This constraint defines the arrival time of an input relative to a clock.

To open the Set Input Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Input Delay**.

External Setup/Hold

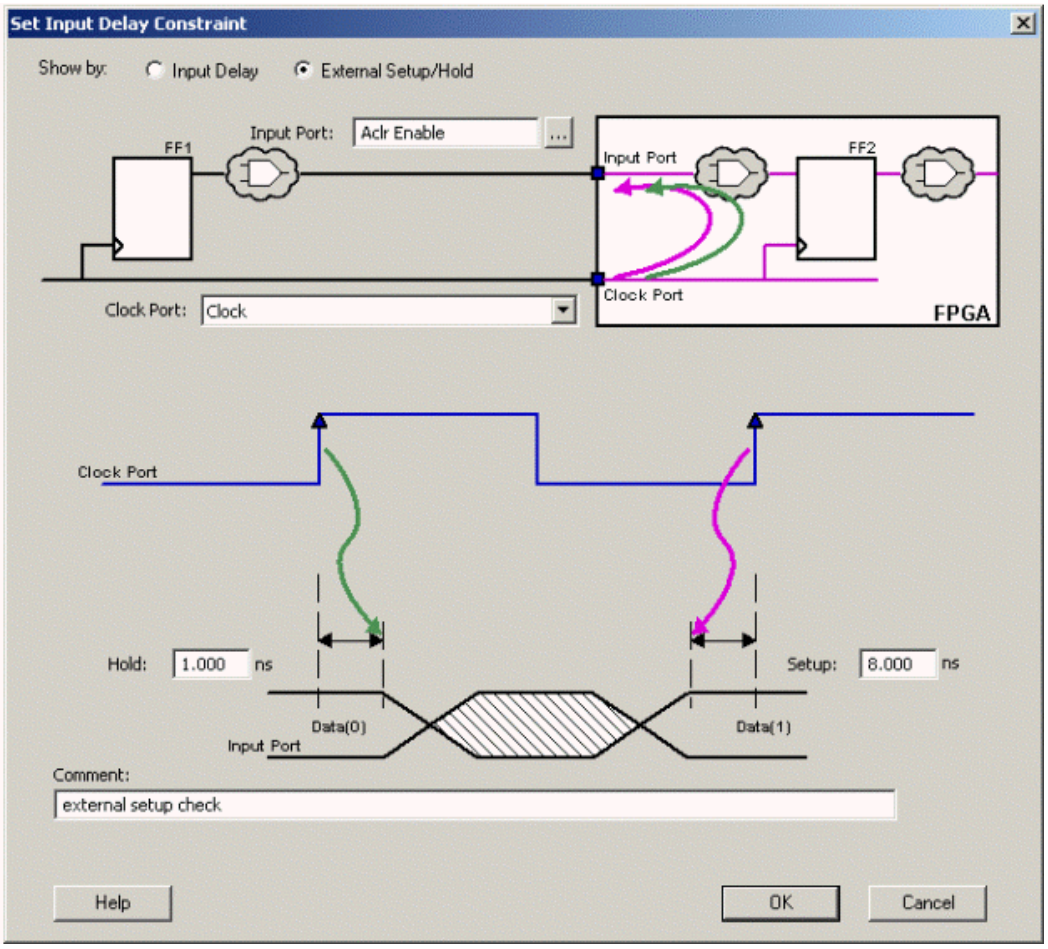


Figure 155 · Set Input Delay Constraint (Show by: External Setup/Hold) Dialog Box

Input Port

Specifies a list of input ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Port

Specifies the clock reference to which the specified External Setup/Hold is related.

External Hold

Specifies the external hold time requirement for the specified input ports.

External Setup

Specifies the external setup time requirement for the specified input ports.

Comment

Enables you to provide comments for this constraint.

Input Delay

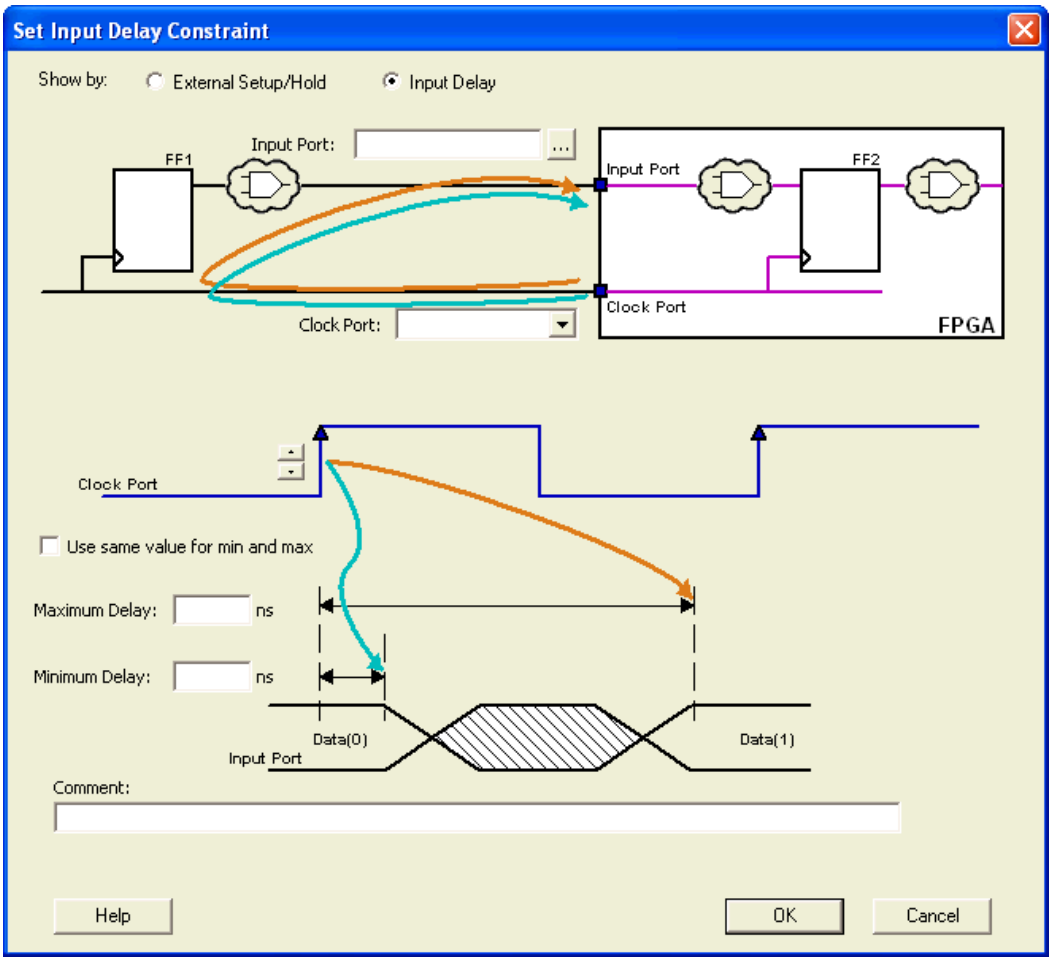


Figure 156 · Set Input Delay Constraint (Show by: Input Delay) Dialog Box

Input Port

Specifies a list of input ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Port

Specifies the clock reference to which the specified input delay is related.

Clock edge

Indicates the launching edge of the clock.

Use max delay for both min and max

Specifies that the minimum input delay uses the same value as the maximum input delay.

Maximum Delay

Specifies that the delay refers to the longest path arriving at the specified input.

Minimum Delay

Specifies that the delay refers to the shortest path arriving at the specified input.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying an Input Delay Constraint

Set Maximum Delay Constraint Dialog Box

Use this dialog box to specify the required maximum delay for timing paths in the current design.

SmartTime automatically derives the individual maximum delay targets from clock waveforms and port input or output delays. So the maximum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multiple cycle path constraint.

To open the Set Maximum Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor, click the **Constraints** menu and choose **Max Delay** (**Constraints > Max Delay**).

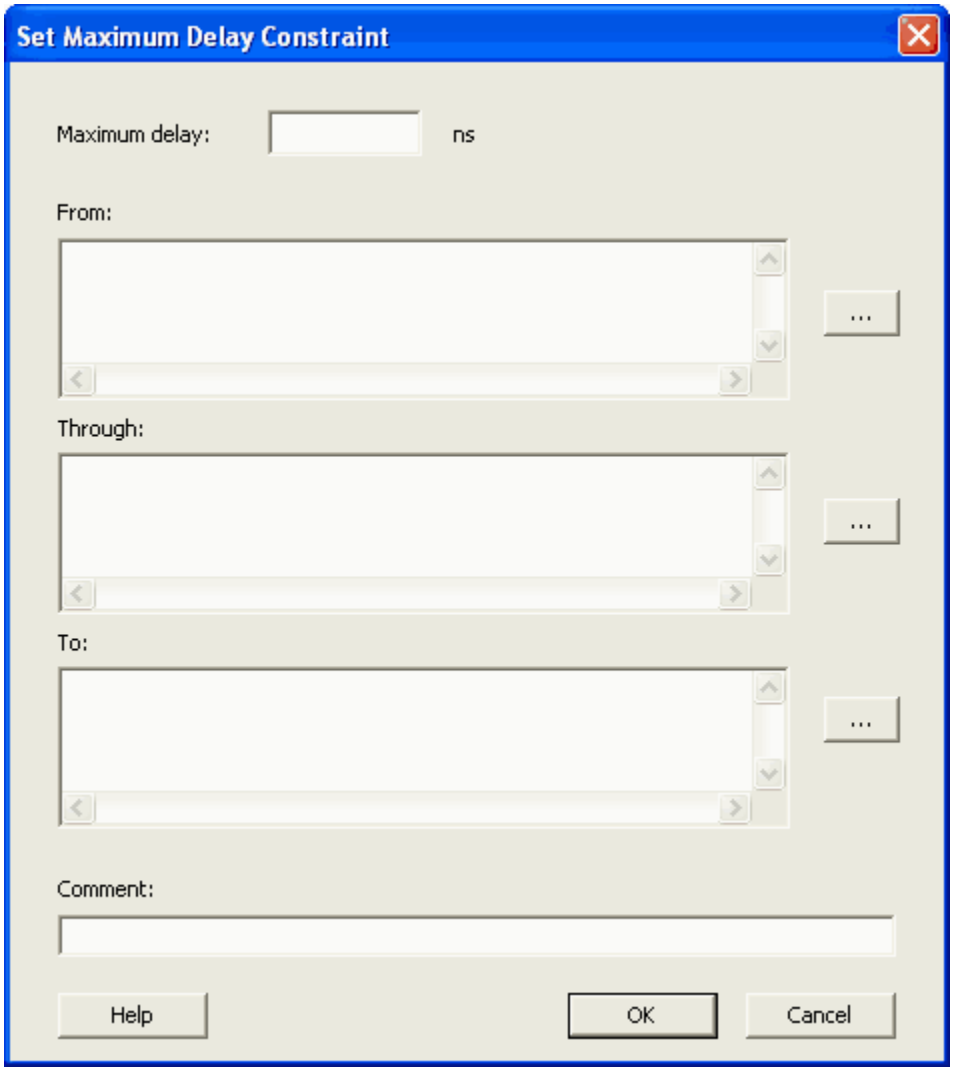


Figure 157 · Set Maximum Delay Constraint Dialog Box

Maximum Delay

Specifies a floating point number in nanoseconds that represents the required maximum delay value for specified paths.

If the path starting point is on a sequential device, SmartTime includes clock skew in the computed delay.

If the path starting point has an input delay specified, SmartTime adds that delay value to the path delay.

If the path ending point is on a sequential device, SmartTime includes clock skew and library setup time in the computed delay.

If the ending point has an output delay specified, SmartTime adds that delay to the path delay.

From

Specifies the starting points for max delay constraint. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

Through

Specifies the through points for the multiple cycle constraint.

To

Specifies the ending points for maximum delay constraint. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying a Maximum Delay Constraint

Set Minimum Delay Constraint Dialog Box

Use this dialog box to specify the required minimum delay for timing paths in the current design.

SmartTime automatically derives the individual minimum delay targets from clock waveforms and port input or output delays. So the minimum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multiple cycle path constraint.

To open the Set Minimum Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor, click the **Constraints** menu and choose **Min Delay** (**Constraints > Min Delay**).

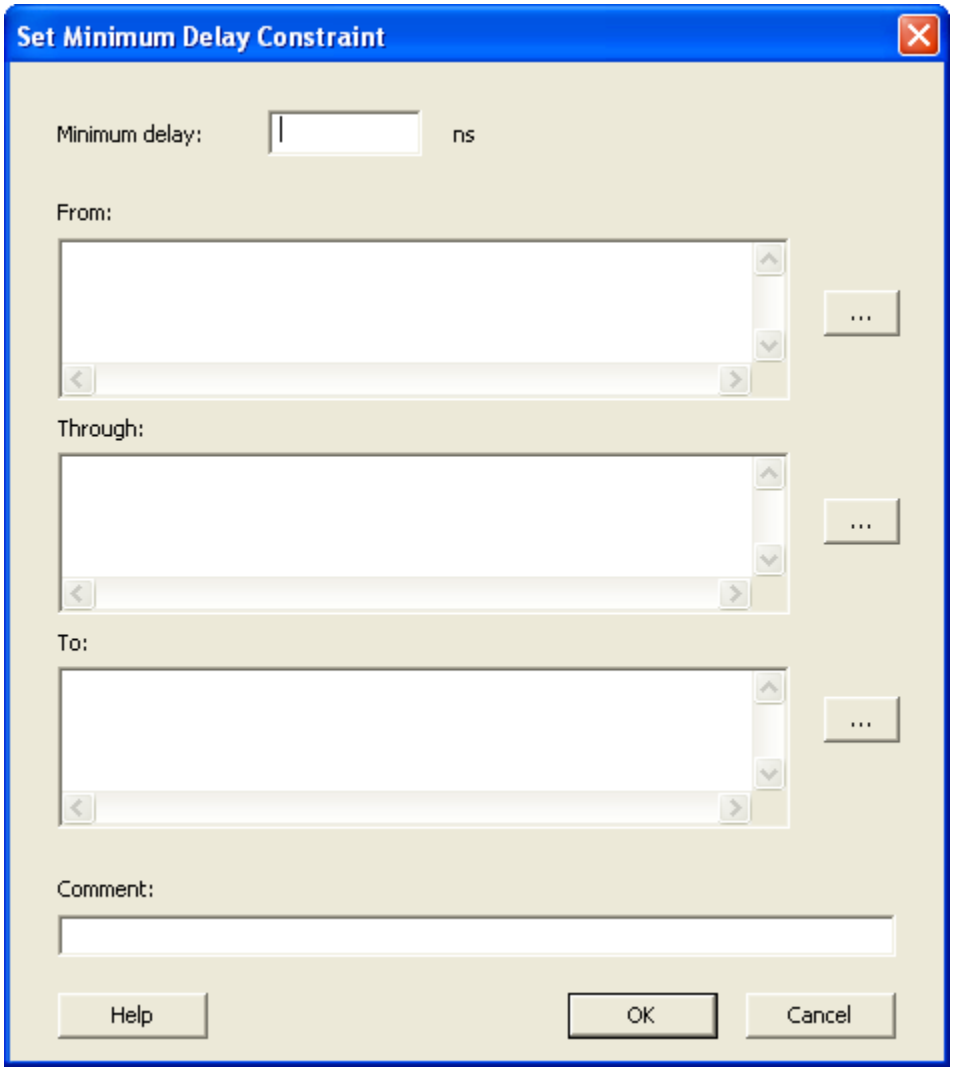


Figure 158 · Set Minimum Delay Constraint Dialog Box

Minimum Delay

Specifies a floating point number in nanoseconds that represents the required minimum delay value for specified paths.

If the path starting point is on a sequential device, SmartTime includes clock skew in the computed delay.

If the path starting point has an input delay specified, SmartTime adds that delay value to the path delay.

If the path ending point is on a sequential device, SmartTime includes clock skew and library setup time in the computed delay.

If the ending point has an output delay specified, SmartTime adds that delay to the path delay.

From

Specifies the starting points for minimum delay constraint. A valid timing starting point is a clock, a primary input, an input port, or a clock pin of a sequential cell.

Through

Specifies the through points for the multiple cycle constraint.

To

Specifies the ending points for minimum delay constraint. A valid timing ending point is a clock, a primary output, an input port, or a data pin of a sequential cell.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying a Minimum Delay Constraint

Set Multicycle Constraint Dialog Box

Use this dialog box to specify the paths that take multiple clock cycles in the current design.

Setting the multiple-cycle paths constraint overrides the single-cycle timing relationships between sequential elements by specifying the number of cycles that the data path must have for setup or hold checks.

Note: When multiple timing constraints are set on the same timing path, the false path constraint has the highest priority and always takes precedence over multiple cycle path constraint. A specific maximum delay constraint overrides a general multicycle path constraint.

To open the Set Multicycle Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Multicycle**.

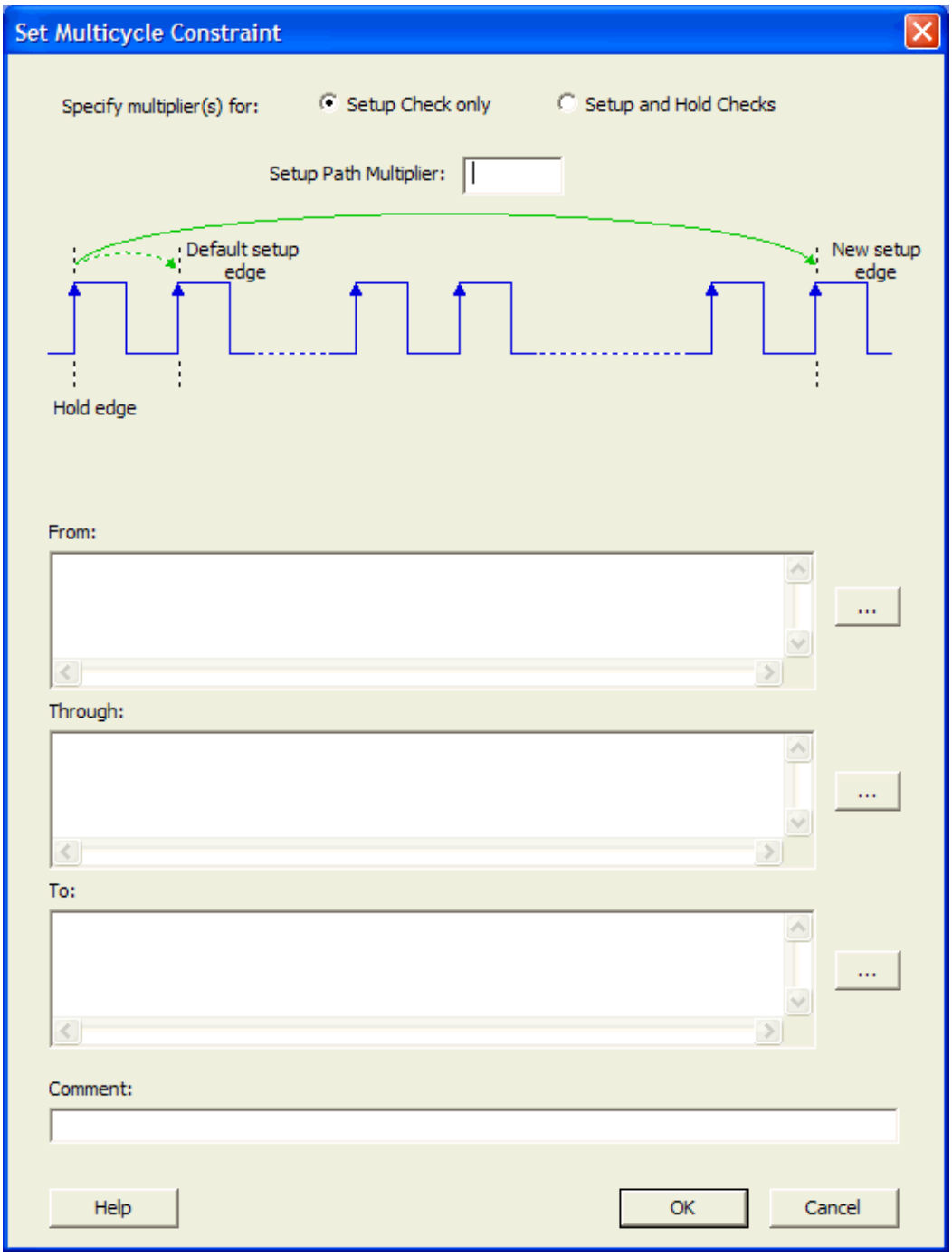


Figure 159 · Set Multicycle Constraint Dialog Box

Setup Path Multiplier

Specifies an integer value that represents a number of cycles the data path must have for a setup check. No hold check will be performed.

From

Specifies the starting points for the multiple cycle constraint. A valid timing starting point is a clock, a primary input, an inout port, or the clock pin of a sequential cell.

Through

Specifies the through points for the multiple cycle constraint.

To

Specifies the ending points for the multiple cycle constraint. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Comment

Enables you to provide comments for this constraint.

When you select the Setup and Hold Checks option, an additional field appears in this dialog box: **Hold Path Multiplier**.

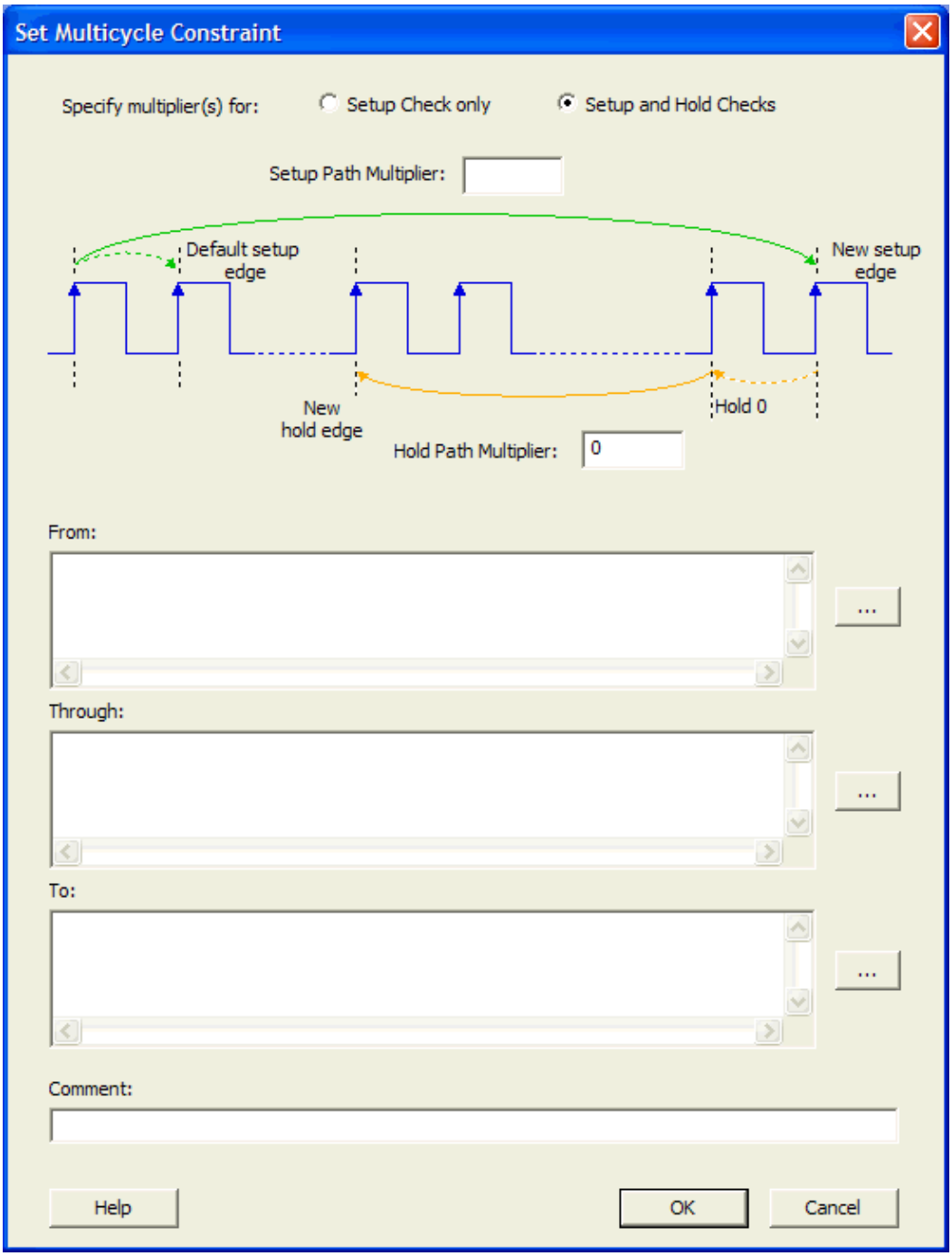


Figure 160 · Set Multicycle Constraint Dialog Box with Setup and Hold Checks Selected

Hold Path Multiplier

Specifies an integer value that represents a number of cycles the data path must have for a hold check, starting from one cycle before the setup check edge.

See Also

Specifying a Multicycle Constraint

Set Output Delay Constraint Dialog Box

Use this dialog box to apply output delay constraints. This constraint defines the output delay of an output relative to a clock.

To open the Set Output Delay Constraint dialog box (shown below) from the SmartTime Constraints Editor, choose **Constraints > Output Delay**.

Clock-to-Output

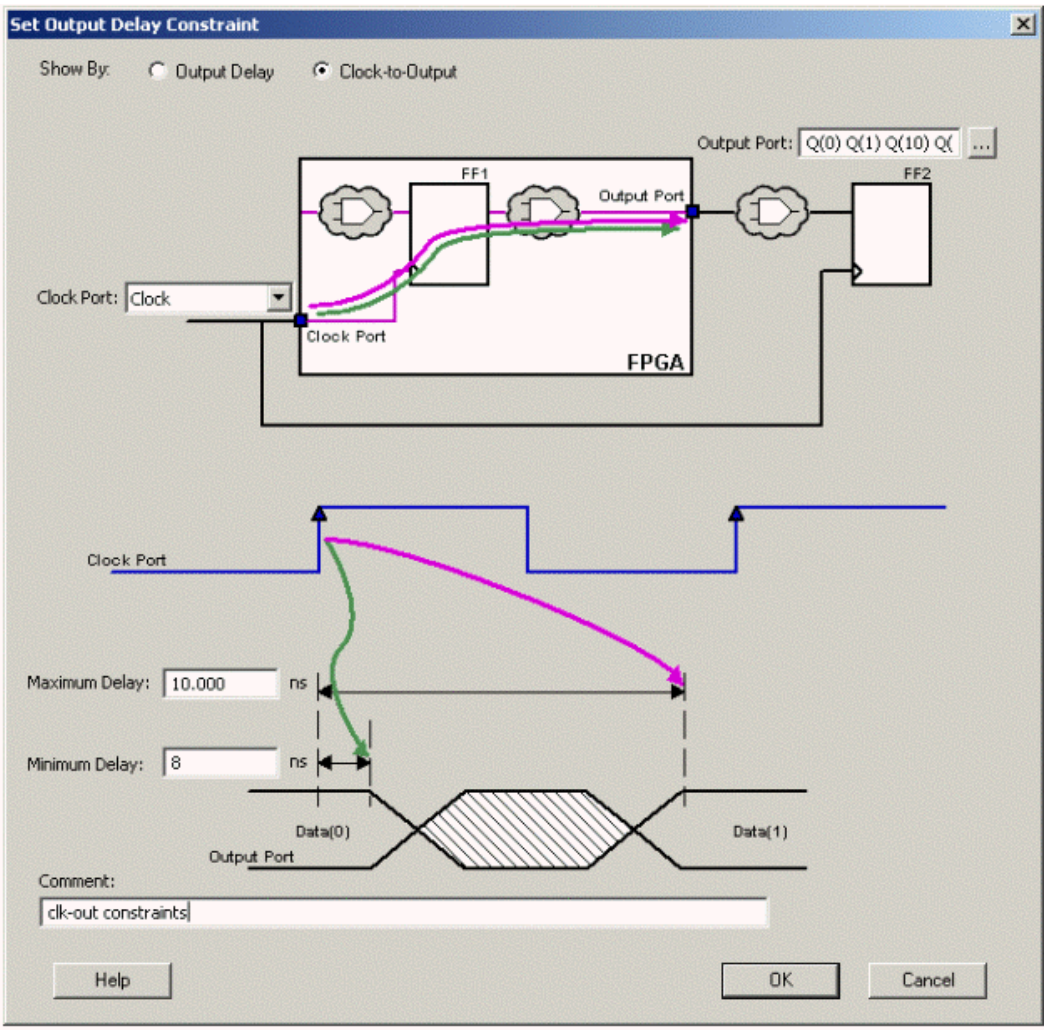


Figure 161 · Set Output Delay (Show By: Clock-to-Output) Dialog Box

Output Port

Specifies a list of output ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Port

Specifies the clock reference to which the specified **Clock-to-Output** is related.

Clock edge

Indicates the clock edge of the launched edge.

Maximum Delay

Specifies the delay for the longest path from the clock port to the output port. This constraint includes the combinational path delay from output of the launched edge to the output port.

Minimum Delay

Specifies the delay for the shortest path from the clock port to the output port. This constraint includes the combinational path delay from output of the launched edge to the output port.

Comment

Enables you to provide comments for this constraint.

Output Delay

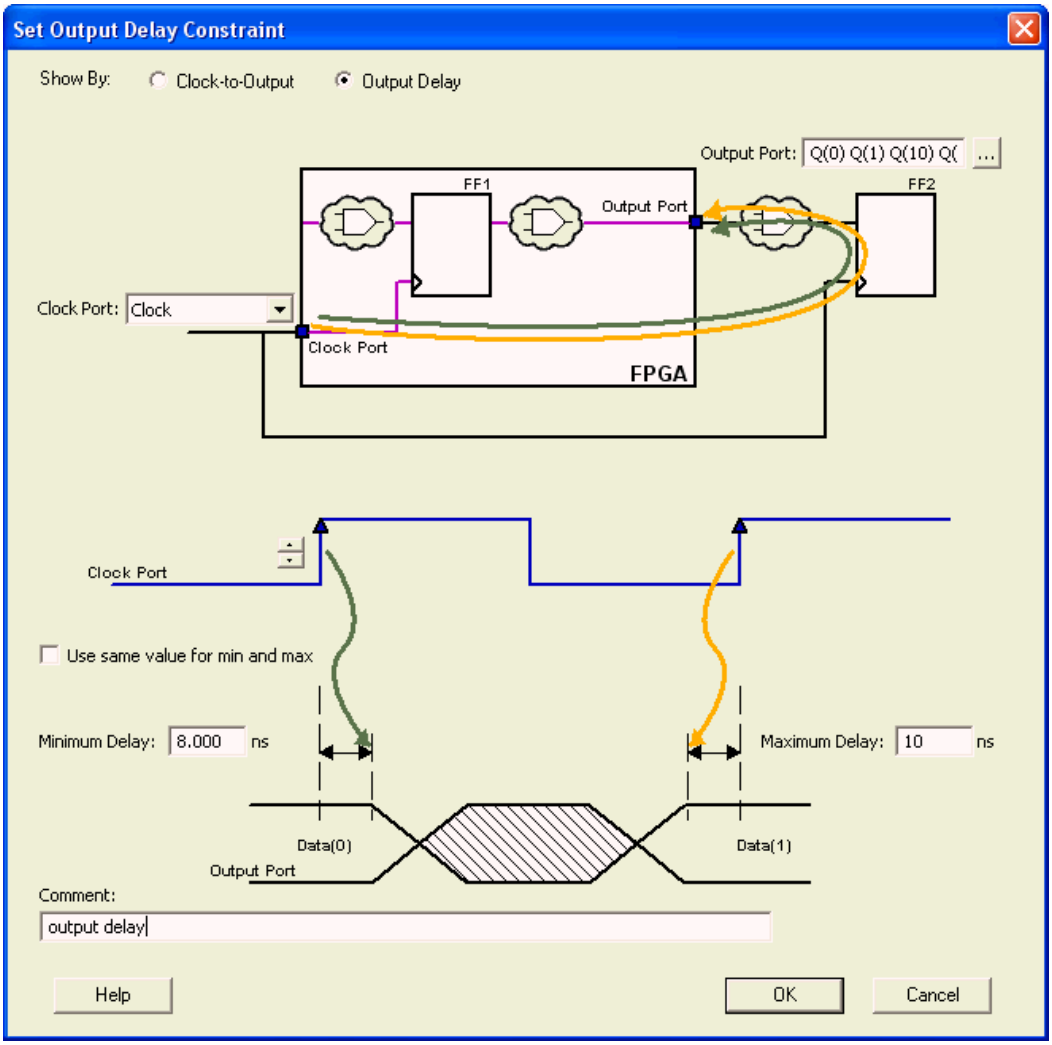


Figure 162 · Set Output Delay (Show By: Output Delay) Dialog Box

Output Port

Specifies a list of output ports in the current design to which the constraint is assigned. You can apply more than one port.

Clock Port

Specifies the clock reference to which the specified output delay is related.

Clock edge

Indicates the launching edge of the clock.

Maximum Delay

Specifies the delay for the longest path from the specified output to the captured edge. This represents a combinational path delay to a register outside the current design plus the library setup time.

Minimum Delay

Specifies the delay for the shortest path from the specified output to the captured edge. This represents a combinational path delay to a register outside the current design plus the library hold time.

Comment

Enables you to provide comments for this constraint.

See Also

Specifying an Output Delay Constraint

SmartTime Options Dialog Box - SmartFusion2, IGLOO2, RTG4

Use this dialog box to specify the SmartTime options to perform timing analysis.

This interface includes the following categories:

- General
- Analysis
- Advanced

To open the SmartTime Options dialog box (shown below) from the SmartTime tool, choose **Tools > Options**.

General

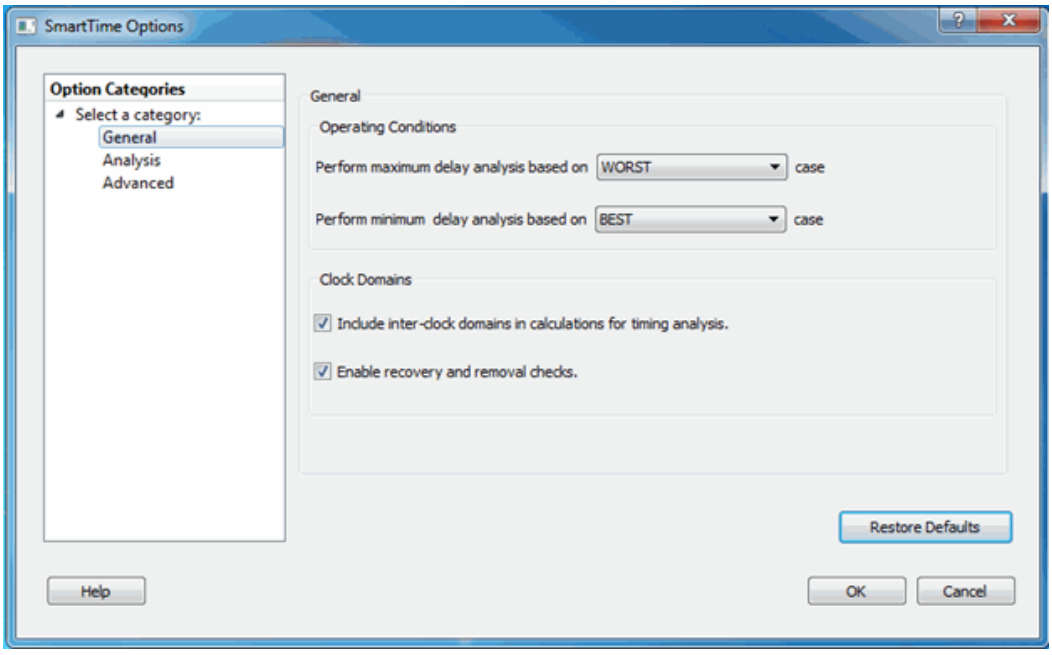


Figure 163 · SmartTime Options - General Dialog Box

Operating Conditions

Allows you to perform maximum or minimum delay analysis based on the Best, Typical, or Worst case. By default, maximum delay analysis is based on WORST case and minimum delay analysis is based on BEST case.

Clock Domains

- **Include inter-clock domains in calculations for timing analysis**: Enables you to specify if SmartTime must use inter-clock domains in calculations for timing analysis. By default, this option is unchecked.
- **Enable recovery and removal checks**: Enables SmartTime to check removal and recovery time on asynchronous signals. Additional sets are created in each clock domain in Analysis View to report the corresponding paths.

Restore Defaults

Resets all the options in the General panel to their default values.

Analysis

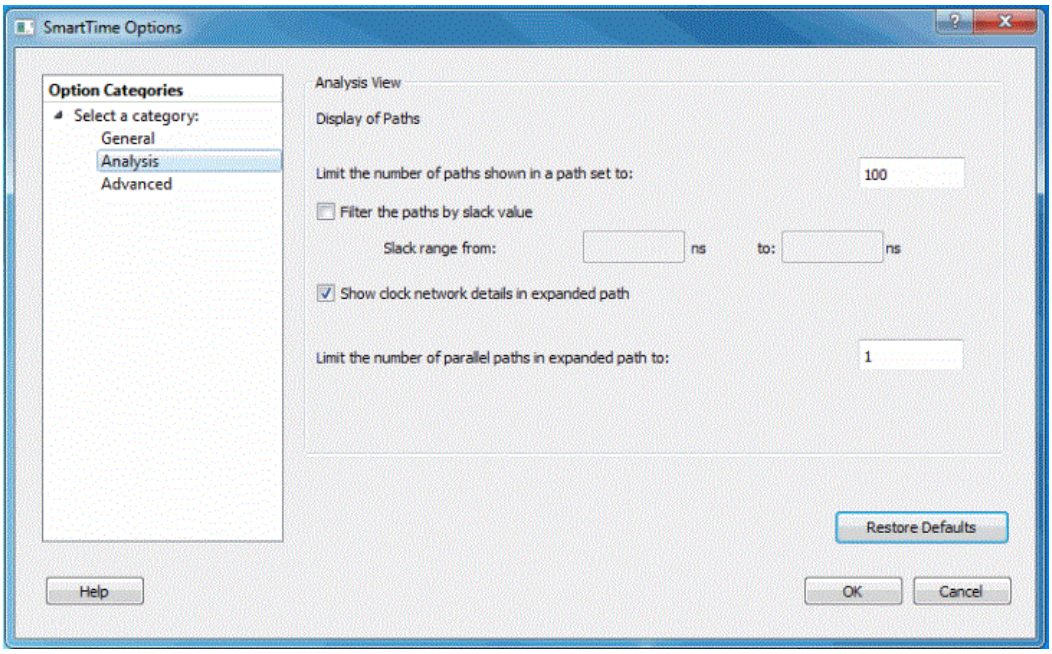


Figure 164 · SmartTime Options - Analysis View Dialog Box

Display of Paths

Limits the number of paths shown in a path set for timing analysis. The default value is 100. You must specify a number greater than 1.

Filter the paths by slack value

Specifies the slack range between minimum slack and maximum slack. This option is unchecked by default.

Show clock network details in expanded path

Displays the clock network details as well as the data path details in the Expanded Path views.

Limit the number of parallel paths in expanded path to: For each expanded path, specify the maximum number of parallel paths that SmartTime displays. The default number of parallel paths is 1.

Restore Defaults

Resets all the options in the Analysis View panel to their default values.

Advanced

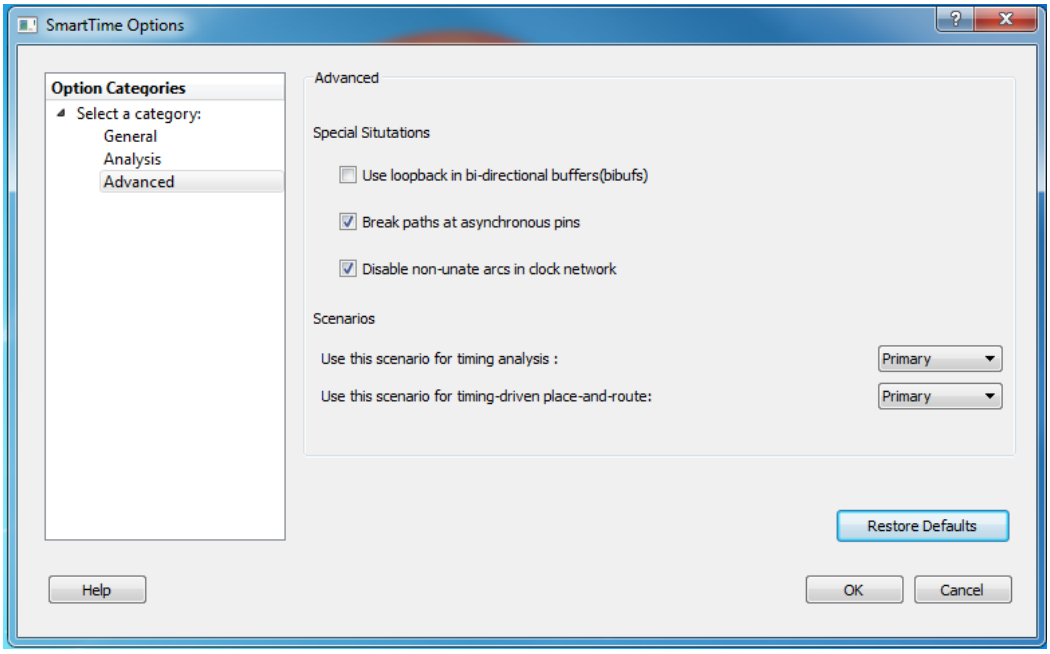


Figure 165 · SmartTime Options - Advanced Dialog Box

Special Situations

Enables you to specify if you need to use loopback in bi-directional buffers (bibufs) and/or break paths at asynchronous pins.

Scenarios

Enables you to select the scenario to use for timing analysis and for timing-driven place-and-route.

Restore Defaults

Resets all the options in the Analysis View panel to their default values.

Store Filter as Analysis Set Dialog Box

Use this dialog box to specify a filter.

To open the **Store Filter as Analysis Set** dialog box (shown below) from the SmartTime Timing Analyzer, select a path and click the **Store Filter** button in the Analysis View Filter.

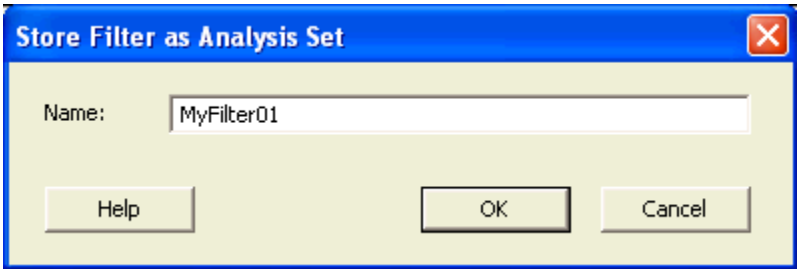


Figure 166 · Store Filter as Analysis Set Dialog Box

Name

Specifies the name of the filtered set.

See Also

Using filters

Timing Bottleneck Analysis Options Dialog Box

Use this dialog box to customize the timing bottleneck report.

You can set report bottleneck options for the following categories:

- General pane
- Bottleneck pane
- Sets pane

To open the Timing Bottleneck Analysis Options dialog box (shown below) from the SmartTime tool, choose **Tools > Bottleneck Analysis**.

General Pane

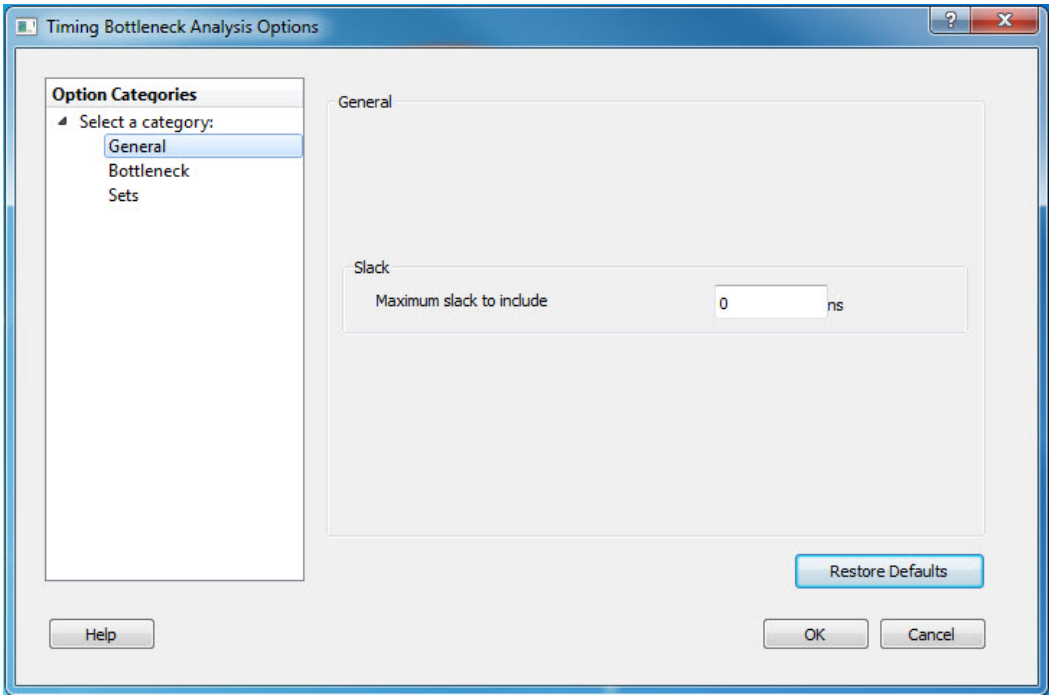


Figure 167 · Timing Bottleneck Report - General Pane Dialog Box

Slack

Lets you specify whether the reported paths will be filtered by threshold, and if so what will be the maximum slack to report. By default the paths are filtered by slack, and the slack threshold is 0.

Restore Defaults

Resets all the options in the General pane to their default values.

Bottleneck Pane

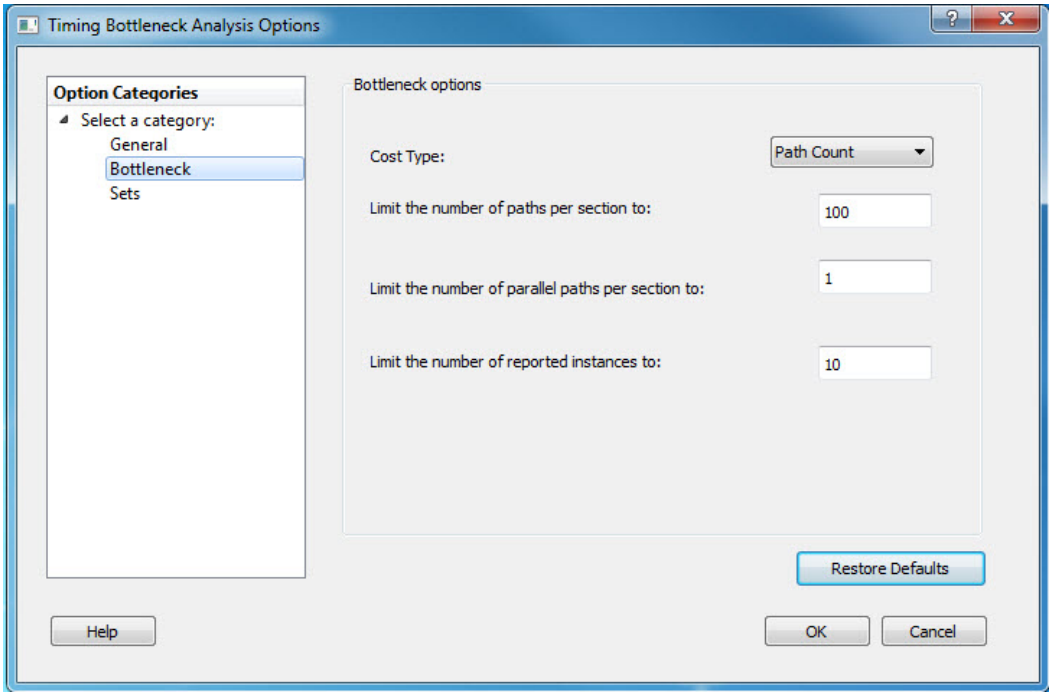


Figure 168 · Timing Bottleneck Report - Bottleneck Pane Dialog Box

Bottleneck Options

Cost Type: Select the cost type that SmartTime will include in the bottleneck report. By default, path count is selected. You may select one of the following two items from the drop-down list:

- **Path count**: This cost type associates the severity of the bottleneck to the count of violating/critical paths that traverse the instance. This is the default.
- **Path cost:** This cost type associates the severity of the bottleneck to the sum of the timing violations for the violating/critical paths that traverse the instance.

Limit the number of paths per section to: Specify the maximum number of paths per set type that SmartTime will include per section in the report. The default maximum number of paths reported is 100.

Limit the number of parallel paths per section to: For each expanded path, specify the maximum number of parallel paths that SmartTime will include in the
report. Only cells that lie on these violating paths are reported. The default number of parallel paths is 1.

Limit the number of reported instances: Specify the maximum number of cells that SmartTime will include per section in the report. The default number of cells is 10.

Restore Defaults

Resets all the options in the Bottleneck panel to their default values.

Sets Pane

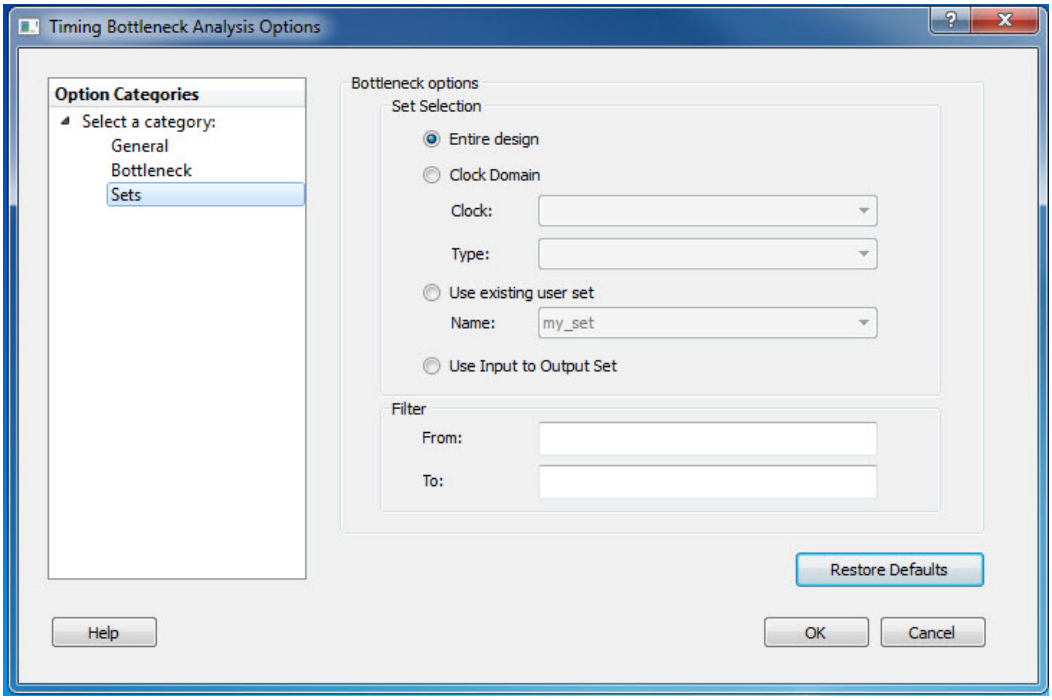


Figure 169 · Timing Bottleneck Report - Sets Pane Dialog Box

This pane has four mutually exclusive options:

- Entire Design
- Clock Domain
- Use existing user set
- Use Input to Output Set

Entire Design: Select this option to display the bottleneck information for the entire design.

Clock Domain: Select this option to display the bottleneck information for the selected clock domain. You can specify the following options:

- Clock: Allows pruning based on a given clock domains. Only cells that lie on these violating paths are reported.
- Type: This option can only be used in conjunction with -clock. The acceptable values are:

Value	Description
Register to Register	Paths between registers in the design
Asynchronous to Register	Paths from asynchronous pins to registers
Register to Asynchronous	Paths from registers to asynchronous pins
External Recovery	The set of paths from inputs to asynchronous pins
External Setup	Paths from input ports to register
Clock to Output	Paths from registers to output ports

Use existing user set: Displays the bottleneck information for the existing user set selected. Only paths that lie within the name set are will be considered towards the bottleneck report.

Filter: Allows you to filter the bottleneck report by the following options:

- From: Reports only cells that lie on violating paths that start at locations specified by this option.
- To: Reports only cells that lie on violating paths that end at locations specified by this option.

Filter defaults to all outputs.

Restore Defaults

Resets all the options in the Paths panel to their default values.

See Also

Bottleneck Analysis

Timing Datasheet Report Options Dialog Box

Use this dialog box to select the output format for your timing datasheet report.

To open the **Timing Datasheet Report Options** dialog box (shown below) from the SmartTime Max/Min Delay Analysis view, choose **Tools > Reports > Datasheet**.

You can generate your report in one of two formats:

Plain Text

Select this option to save your report to disk in plain ASCII text format.

Comma Separated Values

Select this option to save your report to disk in comma-separated value format (.CSV) format, which you can import into a spreadsheet.

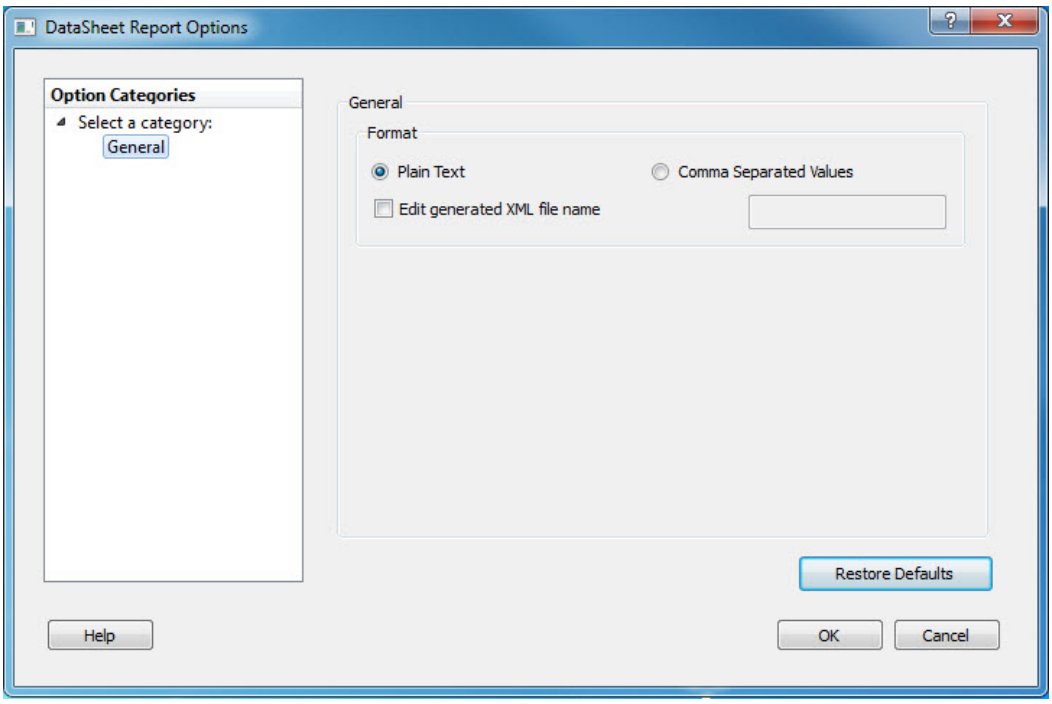


Figure 170 · Datasheet Report Options Dialog Box

Restore Defaults

Resets the option to its default value, which is Plain Text.

See Also

Generating a datasheet report

Understanding datasheet reports
report (Datasheet) using SmartTime

Timing Report Options Dialog Box

Use this dialog box to customize the timing report.

You can set report options for the following categories:

- General
- Paths
- Sets
- Clock Domains

To open the Timing Report Options dialog box (shown below) from the SmartTime Max/Min Delay Analysis View, choose **Tools > Reports> Timer**.

General

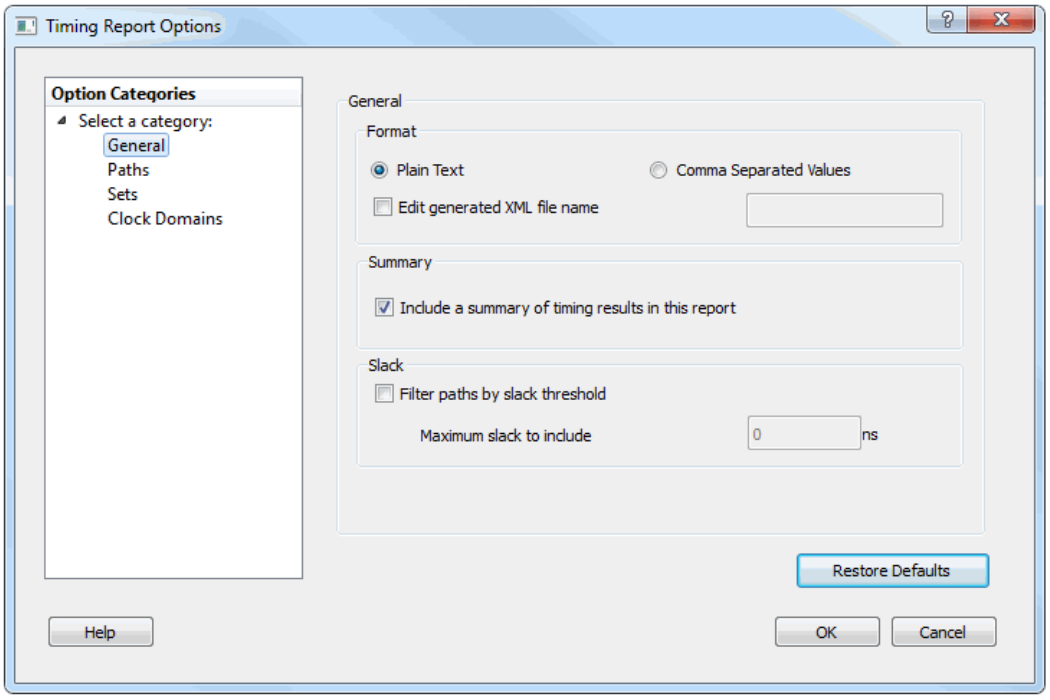


Figure 171 · Timing Report Options - General Dialog Box

Format

Specifies whether or not the report will be exported as a Comma Separated Value (CSV) file or a plain text file. By default, the **Plain Text** option is selected.

Summary

Specifies whether or not the summary section will be included in the report. By default, this option is selected.

Analysis

Specifies the type of analysis to be included in the timing report. It can be either a Maximum Delay Analysis report or Minimum Delay Analysis report. By default, the Maximum Delay Analysis report is included in the timing report.

Slack

Specifies whether the reported paths will be filtered by threshold, and if so what will be the maximum slack to report. By default, the paths are not filtered by slack.

Restore Defaults

Resets all the options in the General panel to their default values.

Paths

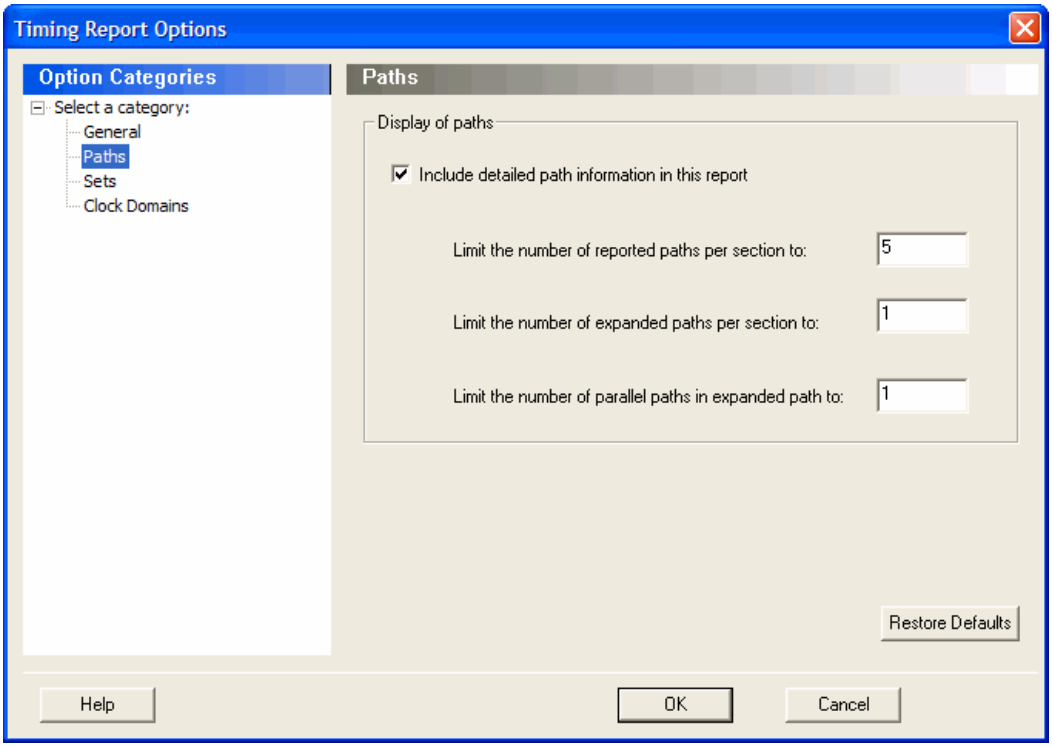


Figure 172 · Timing Report Options - Paths Dialog Box

Display of Paths

Include detailed path information in this report: Check this box to include the detailed path information in the timing report.

Limit the number of reported paths per section to: Specify the maximum number of paths that SmartTime will include per section in the report.

Limit the number of expanded paths per section to: Specify the maximum number of expanded paths that SmartTime will include per section in the report.

Limit the number of parallel paths in expanded path to: For each expanded path, specify the maximum number of parallel paths that SmartTime will include in the report. The default number of parallel paths is 1.

Restore Defaults

Resets all the options in the Paths panel to their default values.

Sets

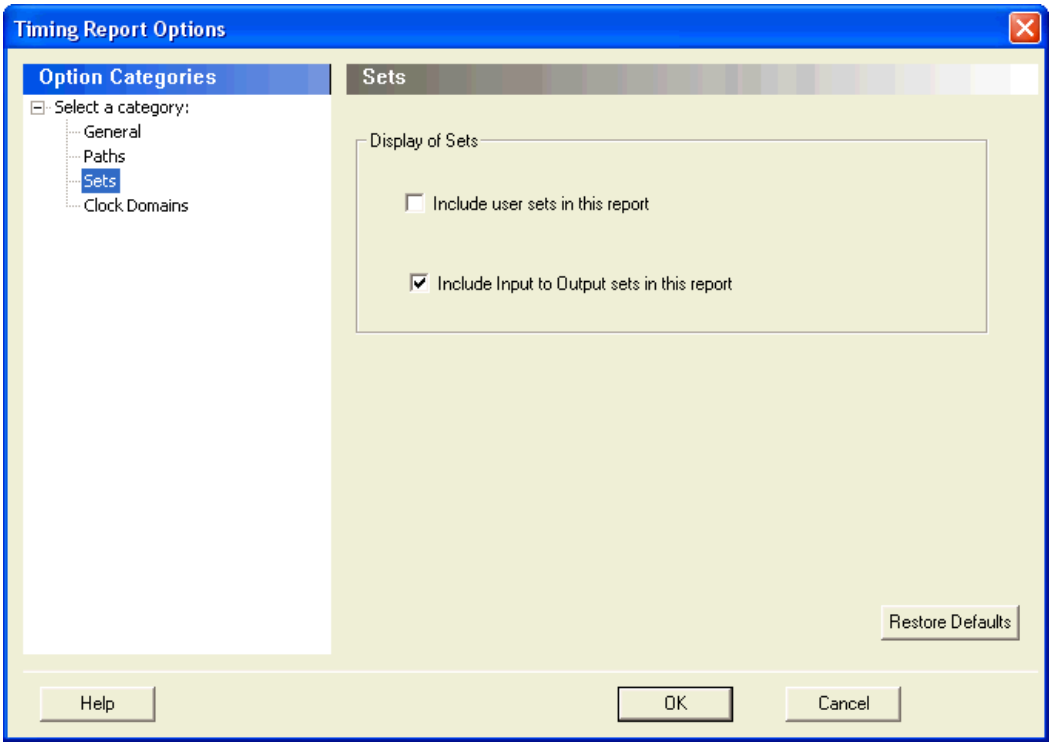


Figure 173 · Timing Report Options - Sets Dialog Box

Display of Sets

Specifies whether or not the user sets will be included in the timing report.

User sets are either filters that you have created and stored on the default paths sets (Register to Register, Inputs to Register, etc.) or Pin to Pin user sets. By default, the paths for these sets are not reported.

In addition, specify whether the Inputs to Output sets will be included in the report. By default, the Input to Output sets are reported.

Restore Defaults

Resets both options in the Sets panel to their default values.

Clock Domains

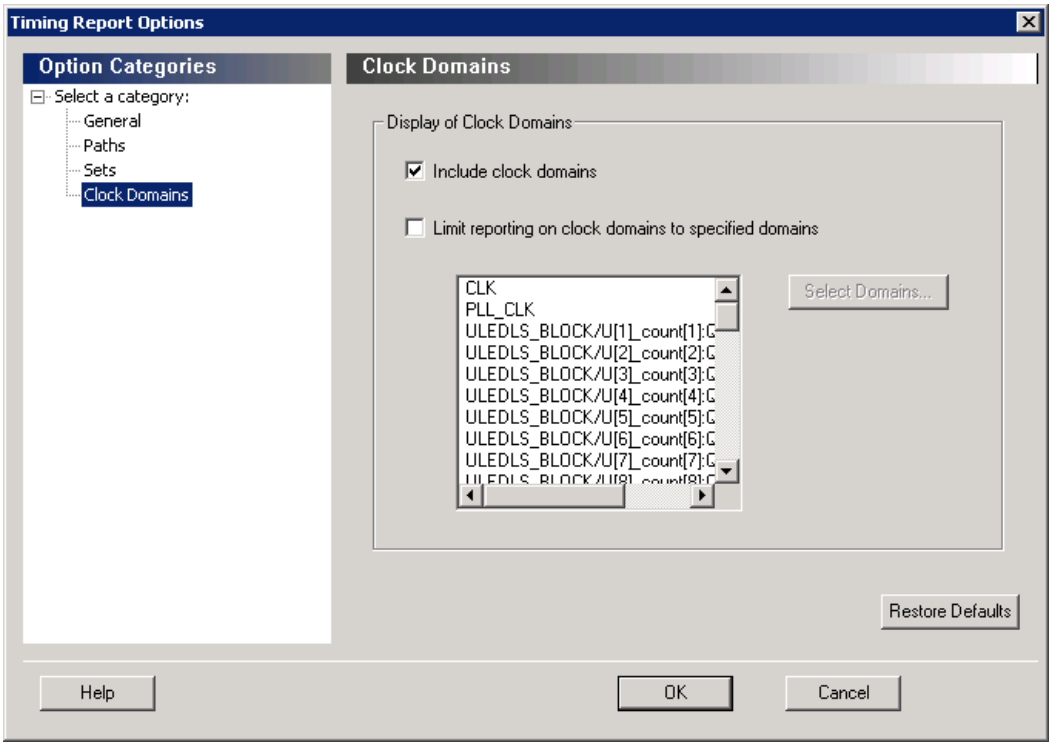


Figure 174 · Timing Report Options - Clock Domains Dialog Box

Display of Clock Domains

Lets you specify what clock domains will be included in the report. By default, the current clock domains used by the timing engine will be reported.

Include Clock Domains

Enables you to include or exclude clock domains in the report. Click the checkbox to include clock domains.

Limit reporting on clock domains to specified domains

Lets you include one or more of the clock domain names in the box, or include additional clock domain names using **Select Domains.**

Restore Defaults

Resets all options in the Clock Domains panel to their default values.

See Also

Generating a datasheet report

Understanding datasheet reports

report (Datasheet) using SmartTime

Timing Violations Report Options Dialog Box

Use this dialog box to customize the timing violation report.

You can set report violation options for the following categories:

- General
- Paths

To open the Timing Report Options dialog box (shown below) from the SmartTime tool, choose **Tools > Reports > Timing Violations**.

General

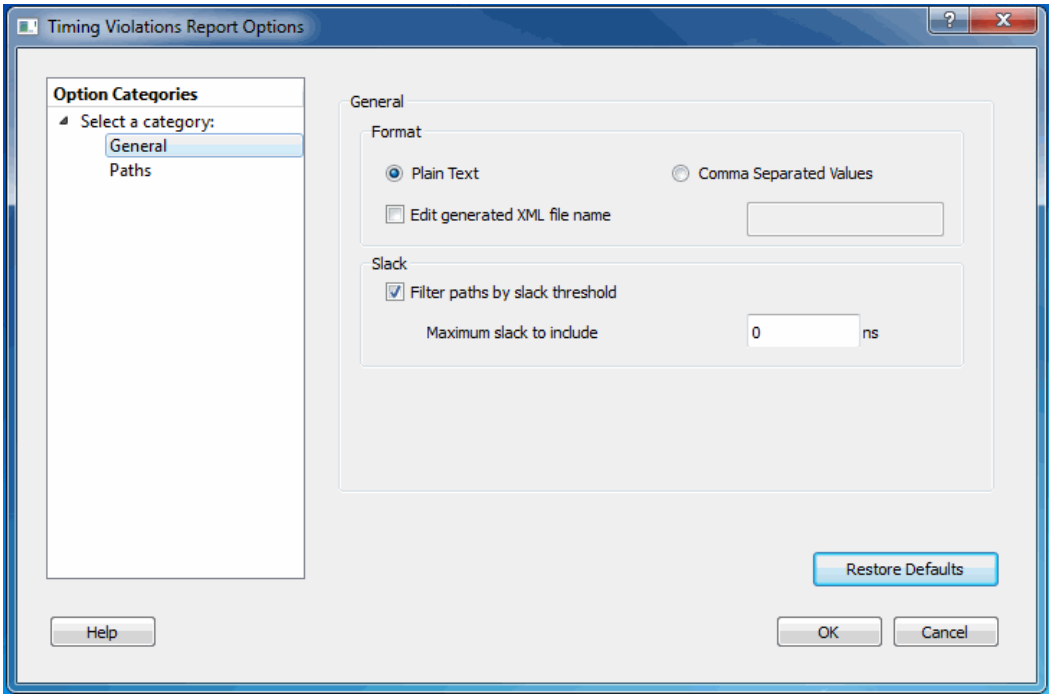


Figure 175 · Timing Violations Report - General Dialog Box

Format

Specifies whether or not the report will be exported as a Comma Separated Value (CSV) file or a plain text file. By default, the **Plain Text** option is selected.

Analysis

Lets you specify what type of analysis will be reported in the report. By default, the report includes Maximum Delay Analysis.

Slack

Lets you specify whether the reported paths will be filtered by threshold, and if so what will be the maximum slack to report. By default the paths are filtered by slack, and the slack threshold is 0.

Restore Defaults

Resets all the options in the General panel to their default values.

Paths

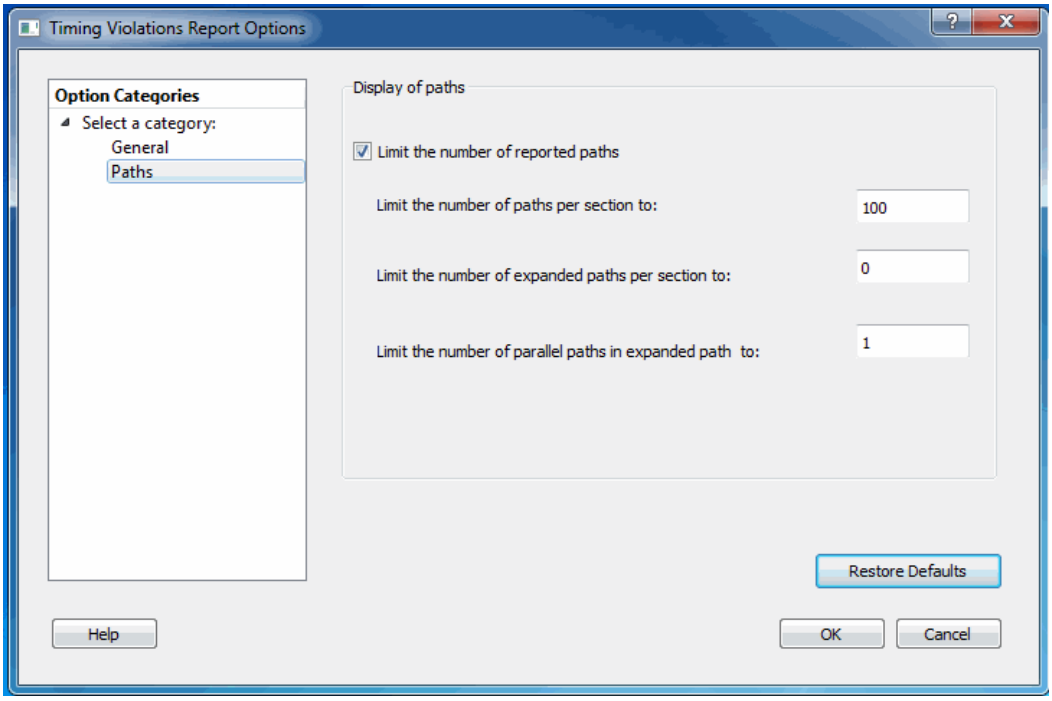


Figure 176 · Timing Violations Report - Paths Dialog Box

Display of paths

Limit the number of reported paths: Check this box to limit the number of paths in the report. By default, the number of paths is limited.

Limit the number of paths per section to: Specify the maximum number of paths that SmartTime will include per section in the report. The default maximum number of paths reported is 100.

Limit the number of expanded paths per section to: Specify the maximum number of expanded paths that SmartTime will include per section in the report. The default number of expanded paths is 0.

Limit the number of parallel paths in expanded path to: For each expanded path, specify the maximum number of parallel paths that SmartTime will include in the report. The default number of parallel paths is 1.

Restore Defaults

Resets all the options in the Paths panel to their default values.

See Also

Generating timing violation report

Understanding timing violation report

report (Timing violations) using SmartTime

Menus, Tools, and Shortcut Keys

File Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Icon	Shortcut	Function
Commit		CTRL + S	Saves changes to the working design for this Designer session only. **Note: To save changes to disk, you must also save your file in Designer.**
Print Preview			Displays the active design in a Preview window
Print		CTRL + P	Displays the Print dialog box from which you can print your active design
Exit			Closes SmartTime

Edit Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Icon	Shortcut	Function
Undo		CTRL + Z	Reverses your last action
Redo		CTRL + Y	Reverses the action of your last Undo command
Cut		CTRL + X	Removes the selection from your design
Copy		CTRL + C	Copies the selection to the Clipboard
Paste		CTRL + V	Pastes the selection from the Clipboard
Modify			Displays appropriate dialog box to edit the current constraint
Delete		Del	Deletes the selected constraint
Select All		CTRL + A	Selects all logic in your design

View Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Icon	Shortcut	Function
Recalculate All		F9	Recalculates all the generated values
Table: Name >			Provides option for customizing the table in the Timing Analysis View
Toolbars >			Hides or displays groups of toolbar buttons
Status Bar			Shows or hides the status bar at the bottom of the window
Scenarios			Shows or hides the scenarios panel

View > Table

Command	Icon	Shortcut	Function
Customize Current Table			Enables you to select columns and the order of the columns for the Paths List in the Timing Analysis View

View > Toolbars

Command	Icon	Shortcut	Function
Standard			Shows or hides the standard toolbar
Constraints			Show or hides the constraints toolbar
Analysis			Shows or hides the analysis toolbar

Actions Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Icon	Shortcut	Function
Constraints >			Provides options to create new constraints
Analysis >			Provides options to perform timing analysis

Actions > Constraints

Command	Icon	Shortcut	Function
Clock			Displays the Create Clock Constraint dialog box
Generated Clock			Displays the Create Generated Clock Constraint dialog box
Input Delay			Displays the Set Input Delay Constraint dialog box
Output Delay			Displays the Set Output Delay Constraint dialog box
Max Delay			Displays the Set Maximum Delay Constraint dialog box
Min Delay			Displays the Set Minimum Delay Constraint dialog box
False Path			Displays the Set False Path Constraint dialog box
Multicycle			Displays the Set Mulitcycle Constraint dialog box
Latency			Displays the Set Clock Source Latency dialog box
Disable Timing			Displays the Set Constraint to Disable Timing Arcs dialog box
Clock to Clock Uncertainty			Displays the Set Clock-to-Clock Uncertainty dialog box

Actions > Analysis

Command	Icon	Shortcut	Function
Clock Domain			Displays Manage Clock Domain dialog box
Path Set			Displays Add Path Analysis Set dialog box

Tools Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Icon	Shortcut	Function
Constraints Editor >			Provides options for constraints scenarios

Command	Icon	Shortcut	Function
Constraint Wizard >			Opens the Constraint Wizard for creating clock and I/O constraints
Timing Analyzer >			Provides options for timing analysis
Constraint Checker			Verifies if all timing constraints are valid
Reports >			Provides options to generate reports
Options			Displays the SmartTime **Options** dialog box

Tools > Constraints Editor

Command	Icon	Shortcut	Function
1. Primary Scenario (and all other available scenarios)			Displays the primary set of timing constraints for the selected scenario
Scenarios			Opens the scenario panel, which lists all available scenarios
New scenario			Creates a new scenario

Tools > Timing Analyzer

Command	Icon	Shortcut	Function
Maximum Delay Analysis			Displays the Maximum Delay Analysis View
Minimum Delay Analysis			Displays the Minimum Delay Analysis View
Bottleneck Analysis			Displays the Bottleneck Analysis View

Tools > Reports

Command	Icon	Shortcut	Function
Report Paths			Displays the **Timing Report Options** dialog box
Report Violations			Displays the **Timing Violations Report Options** dialog box
Report Datasheet			Displays the **Datasheet Report Options** dialog box

Command	Icon	Shortcut	Function
Report Constraints Coverage			Displays the **Constraints Coverage Report Options** dialog box
Report Combinational Loop			Displays the Combinational Loop report

Window Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Function
New Window	Opens another window for the currently active tool **Note: Use these windows to view different parts of the design at the same time.**
Cascade	Arranges windows so you can see the title bar of each window
Tile Horizontally	Arranges windows side-by-side in a horizontal pattern
Tile Vertically	Arranges windows side-by-side in a vertical pattern
Minimize All Windows	Minimizes all active windows
Arrange Icons	Arranges minimized windows left-to-right across the bottom of the Tool window
Close All Windows	Closes all tool views

Help Menu

This menu is available in both Timing Constraint Editor View and Timing Analysis View.

Command	Function
Help Topics	Displays the first Help topic for the SmartTime tool
SmartTime User's Guide	Displays the SmartTime User's Guide
About SmartTime	Displays the current version number and copyright information for the SmartTime tool
Data Change History	Displays features and enhancements, bug fixes and known issues for the current software release that may impact timing data of the current design

SmartTime Toolbar

The SmartTime toolbar contains commands for constraining or analyzing designs. Tool tips are available for each button.

Table 1 · SmartTime Toolbar

Icon	Description
	Commits the changes
	Prints the contents of the constraints editor
	Copies data to the clipboard
	Pastes data from the clipboard
	Modifies the selected object from the constraints editor
	Deletes the selected object from the constraints editor
	Undoes previous changes
	Redoes previous changes
	Opens the maximum delay analysis view
	Opens the minimum delay analysis view
	Opens the manage clock domains manager
	Opens the path set manager
	Recalculates all
	Opens the constraints editor
	Opens the add clock constraint dialog box
	Opens the add generated clock constraint dialog box
	Opens the set input delay clock constraint dialog box
	Opens the set output delay clock constraint dialog box

Icon	Description
	Opens the set false path constraint dialog box
	Opens the set maximum delay constraint dialog box
	Opens the set minimum delay constraint dialog box
	Opens the set multicycle constraint dialog box
	Opens the set clock source latency dialog box
	Opens the set constraint to disable timing arcs dialog box
	Opens the set clock-to-clock uncertainty constraint dialog box
	Checks timing constraints
	Opens the constraint wizard

Data Change History - SmartTime

The data change history lists features, enhancements and bug fixes for the current software release that may impact timing data of the current design.

To generate a data change history, from the **Help** menu, choose **Data Change History**. This opens a data change history in text format.

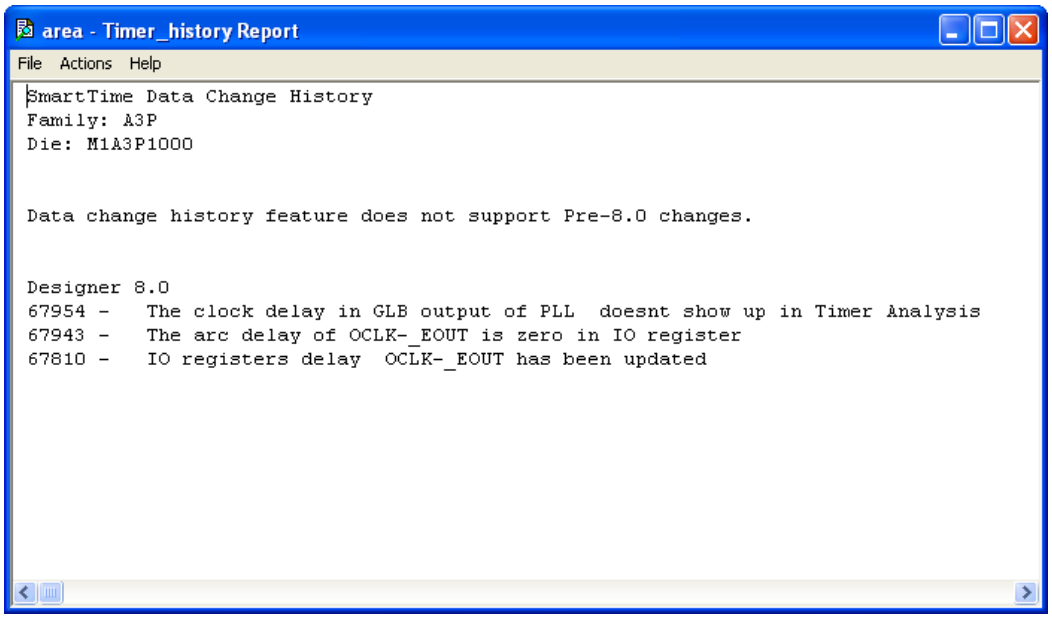


Figure 177 · SmartTime Data Change History Report

Tcl Commands

all_inputs

Tcl command; returns an object representing all input and inout pins in the current design.

```
all_inputs
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Exceptions

You can only use this command as part of a –from, -to, or –through argument in the following Tcl commands: set_min_delay, set_max_delay, set_multicycle_path, and set_false_path.

Examples

```
set_max_delay -from [all_inputs] -to [get_clocks ck1]
```

See Also

Tcl documentation conventions

Designer Tcl Command Reference

all_outputs

Tcl command; returns an object representing all output and inout pins in the current design.

```
all_outputs
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Exceptions

You can only use this command as part of a –from, -to, or –through argument in the following Tcl commands: set_min_delay, set_max_delay, set_multicycle_path, and set_false_path.

Examples

```
set_max_delay -from [all_inputs] -to [all_outputs]
```

See Also

Tcl documentation conventions

Designer Tcl Command Reference

all_registers

Tcl command; returns an object representing register pins or cells in the current scenario based on the given parameters.

```
all_registers [-clock clock_name]
[-async_pins][-output_pins][-data_pins][-clock_pins]
```

Arguments

-clock *clock_name*

Specifies the name of the clock domain to which the registers belong. If no clock is specified, all registers in the design will be targeted.

-async_pins

Lists all register pins that are async pins for the specified clock (or all registers asynchronous pins in the design).

-output_pins

Lists all register pins that are output pins for the specified clock (or all registers output pins in the design).

-data_pins

Lists all register pins that are data pins for the specified clock (or all registers data pins in the design).

-clock_pins

Lists all register pins that are data pins for the specified clock (or all registers clock pins in the design).

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Exceptions

You can only use this command as part of a –from, -to, or –through argument in the following Tcl commands: set_min_delay, set_max_delay, set_multicycle_path, and set_false_path.

Examples

```
set_max_delay 2.000 -from { ff_m:CLK ff_s2:CLK } -to [all_registers -clock_pins -clock {
ff_m:Q }]
```

See Also

Tcl documentation conventions

Designer Tcl Command Reference

check_constraints

Tcl command; checks all timing constraints in the current scenario for validity. This command performs the same checks as when the constraint is entered through SDC or Tcl.

```
check_constraints
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
check_constraints
```

clone_scenario (SmartFusion2, IGLOO2, and RTG4)

Tcl command; creates a new timing scenario by duplicating an existing one. You must provide a unique name (that is, it cannot already be used by another timing scenario).

```
clone_scenario original  new_scenario_name
```

Arguments

original

Specifies the name of the source timing scenario to clone (copy). The source must be a valid, existing timing scenario.

new_scenario_name

Specifies the name of the new scenario to be created.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command creates a timing scenario with the *new_scenario_name*, which includes a copy of all constraints in the original scenario. The new scenario is then added to the list of scenarios.

Example

```
clone_scenario primary my_new_scenario
```

See Also

create_scenario

delete_scenario

Tcl documentation conventions

Designer Tcl Command Reference

create_clock

Tcl command; creates a clock constraint on the specified ports/pins, or a virtual clock if no source other than a name is specified.

```
create_clock -period period_value [-name clock_name]
[-waveform> edge_list][source_objects]
```

Arguments

-period *period_value*

Specifies the clock period in nanoseconds. The value you specify is the minimum time over which the clock waveform repeats. The period_value must be greater than zero.

-name *clock_name*

Specifies the name of the clock constraint. You must specify either a clock name or a source.

`-waveform` *`edge_list`*

Specifies the rise and fall times of the clock waveform in ns over a complete clock period. There must be exactly two transitions in the list, a rising transition followed by a falling transition. You can define a clock starting with a falling edge by providing an edge list where fall time is less than rise time. If you do not specify -waveform option, the tool creates a default waveform, with a rising edge at instant 0.0 ns and a falling edge at instant (period_value/2)ns.

`source_objects`

Specifies the source of the clock constraint. The source can be ports, pins, or nets in the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing one. You must specify either a source or a clock name.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Creates a clock in the current design at the declared source and defines its period and waveform. The static timing analysis tool uses this information to propagate the waveform across the clock network to the clock pins of all sequential elements driven by this clock source.

The clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

Examples

The following example creates two clocks on ports CK1 and CK2 with a period of 6, a rising edge at 0, and a falling edge at 3:

```
create_clock -name {my_user_clock} -period 6 CK1
create_clock -name {my_other_user_clock} –period 6 –waveform {0 3} {CK2}
```

The following example creates a clock on port CK3 with a period of 7, a rising edge at 2, and a falling edge at 4:

```
create_clock –period 7 –waveform {2 4} [get_ports {CK3}]
```

See Also

`create_generated_clock`

Tcl Command Documentation Conventions

Designer Tcl Command Reference

create_generated_clock

Tcl command; creates an internally generated clock constraint on the ports/pins and defines its characteristics.

```
create_generated_clock [-name name] -source reference_pin [-divide_by divide_factor] [-
multiply_by multiply_factor] [-invert] source
```

Arguments

`-name` *`name`*

Specifies the name of the clock constraint.

`-source` *`reference_pin`*

Specifies the reference pin in the design from which the clock waveform is to be derived.

`-divide_by` *`divide_factor`*

Specifies the frequency division factor. For instance if the *divide_factor* is equal to 2, the generated clock period is twice the reference clock period.

`-multiply_by` *`multiply_factor`*

Specifies the frequency multiplication factor. For instance if the *multiply_factor* is equal to 2, the generated clock period is half the reference clock period.

`-invert`

Specifies that the generated clock waveform is inverted with respect to the reference clock.

`source`

Specifies the source of the clock constraint on internal pins of the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing clock. Only one source is accepted. Wildcards are accepted as long as the resolution shows one pin.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Creates a generated clock in the current design at a declared source by defining its frequency with respect to the frequency at the reference pin. The static timing analysis tool uses this information to compute and propagate its waveform across the clock network to the clock pins of all sequential elements driven by this source.

The generated clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

Examples

The following example creates a generated clock on pin U1/reg1:Q with a period twice as long as the period at the reference port CLK.

```
create_generated_clock -name {my_user_clock} –divide_by 2 –source [get_ports {CLK}]  U1/reg1:Q
```

The following example creates a generated clock at the primary output of myPLL with a period ¾ of the period at the reference pin clk.

```
create_generated_clock –divide_by 3 –multiply_by 4  -source clk [get_pins {myPLL:CLK1}]
```

See Also

`create_clock`

Tcl Command Documentation Conventions

Designer Tcl Command Reference

create_scenario

Tcl command; creates a new timing scenario with the specified name. You must provide a unique name (that is, it cannot already be used by another timing scenario).

```
create_scenario name
```

Arguments

`name`

Specifies the name of the new timing scenario.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

A timing scenario is a set of timing constraints used with a design. Scenarios enable you to easily refine the set of timing constraints used for Timing-Driven Place-and-Route, so as to achieve timing closure more rapidly.

This command creates an empty timing scenario with the specified name and adds it to the list of scenarios.

Example

```
create_scenario scenario_A
```

See Also

clone_scenario

Tcl Command Documentation Conventions

Designer Tcl Command Reference

expand_path

Tcl command; displays expanded path information (path details) for paths. The paths to be expanded are identified by the parameters required to display these paths with list_paths. For example, to expand the first path listed with list_paths -clock {MYCLOCK} -type {register_to_register}, use the command expand_path -clock {MYCLOCK} -type {register_to_register}. Path details contain the pin name, type, net name, cell name, operation, delay, total delay, and edge as well as the arrival time, required time, and slack. These details are the same as details available in the SmartTime Expanded Path window.

```
expand_path
-index value
-set name
-clock clock name
-type set_type
-analysis {max| min}
-format {csv | text}
-from_clock clock name
-to_clock clock name
```

Arguments

`-index value`

Specify the index of the path to be expanded in the list of paths. Default is 1.

`-analysis {max | min}`

Specify whether the timing analysis is done is max-delay (setup check) or min-delay (hold check). Valid values: max or min.

`-format {csv | text}`

Specify the list format. It can be either text (default) or csv (comma separated values). The former is suited for display the latter for parsing.

`-set name`

Displays a list of paths from the named set. You can either use the -set option to specify a user set by its name or use both -clock and -type to specify a set.

`-clock clock name`

Displays the set of paths belonging to the specified clock domain. You can either use this option along with -type to specify a set or use the -set option to specify the name of the set to display.

`-type set_type`

Specifies the type of paths in the clock domain to display in a list. You can only use this option with the -clock option. You can either use this option along with -clock to specify a set or use the -set option to specify a set name.

Value	Description
reg_to_reg	Paths between registers in the design
external_setup	Path from input ports to registers
external_hold	Path from input ports to registers
clock_to_out	Path from registers to output ports
reg_to_async	Path from registers to asynchronous pins
external_recovery	Set of paths from inputs to asynchronous pins
external_removal	Set of paths from inputs to asynchronous pins
async_to_reg	Path from asynchronous pins to registers

`-from_clock` *`clock_name`*

Displays a list of timing paths for an inter-clock domain set belonging to the source clock specified. You can only use this option with the -to_clock option, not by itself.

`-to_clock` *`clock_name`*

Displays a list of timing paths for an inter-clock domain set belonging to the sink clock specified. You can only use this option with the -from_clock option, not by itself.

`-analysis` *`name`*

Specifies the analysis for the paths to be listed. The following table shows the acceptable values for this argument.

Value	Description
maxdelay	Maximum delay analysis
mindelay	Minimum delay analysis

`-index` *`list_of_indices`*

Specifies which paths to display. The index starts at 1 and defaults to 1. Only values lower than the max_paths option will be expanded.

`-format` *`value`*

Specifies the file format of the output. The following table shows the acceptable values for this argument:

Value	Description
text	ASCII text format
csv	Comma separated value file format

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

Note: The following example returns a list of five paths:

```
puts [expand_path –clock { myclock } –type {reg_to_reg }]
puts [expand_path –clock {myclock} –type {reg_to_reg} –index { 1 2 3 } –format text]
```

See Also

list_paths

get_cells

Tcl command; returns an object representing the cells (instances) that match those specified in the pattern argument.

```
get_cells pattern
```

Arguments

`pattern`

Specifies the pattern to match the instances to return. For example, "get_cells U18*" returns all instances starting with the characters "U18", where “*” is a wildcard that represents any character string.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command returns a collection of instances matching the pattern you specify. You can only use this command as part of a –from, -to, or –through argument in the following Tcl commands: set_max delay, set_multicycle_path, and set_false_path.

Examples

```
set_max_delay 2 -from [get_cells {reg*}] -to [get_ports {out}]
set_false_path –through [get_cells {Rblock/muxA}]
```

See Also

get_clocks
get_nets
get_pins
get_ports
Tcl Command Documentation Conventions
Designer Tcl Command Reference

get_clocks

Tcl command; returns an object representing the clock(s) that match those specified in the pattern argument in the current timing scenario.

```
get_clocks pattern
```

Arguments

pattern

Specifies the pattern to use to match the clocks set in SmartTime.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

- If this command is used as a –from argument in either the set maximum (set_max_delay), or set minimum delay (set_min_delay), false path (set_false_path), and multicycle constraints (set_multicycle_path), the clock pins of all the registers related to this clock are used as path start points.
- If this command is used as a –to argument in either the set maximum (set_max_delay), or set minimum delay (set_min_delay), false path (set_false_path), and multicycle constraints (set_multicycle_path), the synchronous pins of all the registers related to this clock are used as path endpoints.

Example

```
set_max_delay -from [get_ports datal] -to \
[get_clocks ck1]
```

See Also

create_clock

create_generated_clock

Tcl Command Documentation Conventions

Designer Tcl Command Reference

get_current_scenario

Tcl command; returns the name of the current timing scenario.

```
get_current_scenario
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
get_current_scenario
```

See Also

set_current_scenario

Tcl documentation conventions

Designer Tcl Command Reference

get_nets

Tcl command; returns an object representing the nets that match those specified in the pattern argument.

```
get_nets pattern
```

Arguments

pattern

Specifies the pattern to match the names of the nets to return. For example, "get_nets N_255*" returns all nets starting with the characters "N_255", where “*” is a wildcard that represents any character string.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command returns a collection of nets matching the pattern you specify. You can only use this command as source objects in create clock (create_clock) or create generated clock (create_generated_clock) constraints and as `-through` arguments in the set false path, set minimum delay, set maximum delay, and set multicycle path constraints.

Examples

```
set_max_delay 2 -from [get_ports RDATA1] -through [get_nets {net_chkp1 net_chkqi}]
set_false_path –through [get_nets {Tblk/rm/n*}]
create_clock -name mainCLK -period 2.5 [get_nets {cknet}]
```

See Also

create_clock
create_generated_clock
set_false_path
set_min_delay
set_max_delay
set_multicycle_path
Tcl documentation conventions
Designer Tcl Command Reference

get_pins

Tcl command; returns an object representing the pin(s) that match those specified in the pattern argument.

```
get_pins pattern
```

Arguments

pattern

Specifies the pattern to match the pins to return. For example, "get_pins clock_gen*" returns all pins starting with the characters "clock_gen", where “*” is a wildcard that represents any character string.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
create_clock -period 10 [get_pins clock_gen/reg2:Q]
```

See Also

create_clock
create_generated_clock
set_clock_latency
set_false_path
set_min_delay

set_max_delay
set_multicycle_path
Tcl documentation conventions
Designer Tcl Command Reference

get_ports

Tcl command; returns an object representing the port(s) that match those specified in the pattern argument.

```
get_ports pattern
```

Argument

pattern
Specifies the pattern to match the ports.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
create_clock -period 10 [get_ports CK1]
```

See Also

create_clock
set_clock_latency
set_input_delay
set_output_delay
set_min_delay
set_max_delay
set_false_path
set_multicycle_path
Tcl documentation conventions
Designer Tcl Command Reference

list_clock_latencies

Tcl command; returns details about all of the clock latencies in the current timing constraint scenario.

```
list_clock_latencies
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_clock_latencies]
```

See Also

set_clock_latency
remove_clock_latency
Tcl documentation conventions
Designer Tcl Command Reference

list_clock_uncertainties

Tcl command; returns details about all of the clock uncertainties in the current timing constraint scenario.

```
list_clock_uncertainties
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
list_clock_uncertainties
```

See Also

set_clock_uncertainty
remove_clock_uncertainty
Designer Tcl Command Reference

list_clocks

Tcl command; returns details about all of the clock constraints in the current timing constraint scenario.

```
list_clocks
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_clocks]
```

See Also

create_clock
remove_clock
Tcl documentation conventions
Designer Tcl Command Reference

list_disable_timings

Tcl command; returns the list of disable timing constraints for the current scenario.

```
list_disable_timings
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
list_disable_timings
```

See Also

Designer Tcl Command Reference

list_false_paths

Tcl command; returns details about all of the false paths in the current timing constraint scenario.

```
list_false_paths
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_false_paths]
```

See Also

set_false_path
remove_false_path
Tcl documentation conventions
Designer Tcl Command Reference

list_generated_clocks

Tcl command; returns details about all of the generated clock constraints in the current timing constraint scenario.

```
list_generated_clocks
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_generated_clocks]
```

See Also

create_generated_clock
remove_generated_clock
Tcl documentation conventions
Designer Tcl Command Reference

list_input_delays

Tcl command; returns details about all of the input delay constraints in the current timing constraint scenario.

```
list_input_delays
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_input_delays]
```

See Also

set_input_delay
remove_input_delay
Tcl documentation conventions
Designer Tcl Command Reference

list_max_delays

Tcl command; returns details about all of the maximum delay constraints in the current timing constraint scenario.

```
list_max_delays
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_max_delays]
```

See Also

set_max_delay
remove_max_delay
Tcl documentation conventions
Designer Tcl Command Reference

list_min_delays

Tcl command; returns details about all of the minimum delay constraints in the current timing constraint scenario.

```
list_min_delays
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_min_delays]
```

See Also

set_min_delay
remove_min_delay
Tcl documentation conventions
Designer Tcl Command Reference

list_multicycle_paths

Tcl command; returns details about all of the multicycle paths in the current timing constraint scenario.

```
list_multicycle_paths
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_multicycle_paths]
```

See Also

set_multicycle_path
remove_multicycle_path
Tcl documentation conventions
Designer Tcl Command Reference

list_objects

Tcl command; returns a list of object matching the parameter. Objects can be nets, pins, ports, clocks or instances.

```
list_objects <object>
```

Arguments

Any timing constraint parameter.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

The following example lists all the inputs in your design:

```
list_objects [all_inputs]
```

You can also use wildcards to filter your list, as in the following command:

```
list_objects [get_ports a*]
```

See Also

Tcl documentation conventions

Designer Tcl Command Reference

list_output_delays

Tcl command; returns details about all of the output delay constraints in the current timing constraint scenario.

```
list_output_delays
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
puts [list_output_delays]
```

See Also

set_output_delay

remove_output_delay

Tcl documentation conventions

Designer Tcl Command Reference

list_paths

Tcl command; returns a list of the *n* worst paths matching the arguments. The number of paths returned can be changed using the set_options -limit_max_paths <*value*> command.

```
list_paths
-analysis <max | min>
-format <csv | text>
-set <name>
-clock <clock name>
-type <set_type>
-from_clock <clock name>
-to_clock <clock name>
-in_to_out
-from <port/pin pattern>
-to <port/pin pattern>
```

Arguments

`-analysis <max | min>`

Specifies whether the timing analysis is done for max-delay (setup check) or min-delay (hold check). Valid values are: max or min.

`-format < text | csv >`

Specifies the list format. It can be either text (default) or csv (comma separated values). Text format is better for display and csv format is better for parsing.

`-set <name>`

Returns a list of paths from the named set. You can either use the -set option to specify a user set by its name or use both -clock and -type to specify a set.

`-clock <clock name>`

Returns a list of paths from the specified clock domain. This option requires the -type option.

`-type <set_type>`

Specifies the type of paths to be included. It can only be used along with -clock. Valid values are:

reg_to_reg -- Paths between registers

external_setup -- Path from input ports to data pins of registers

external_hold -- Path from input ports to data pins of registers

clock_to_out -- Path from registers to output ports

reg_to_async -- Path from registers to asynchronous pins of registers

external_recovery -- Path from input ports to asynchronous pins of registers

external_removal -- Path from input ports to asynchronous pins of registers

async_to_reg -- Path from asynchronous pins to registers

`-from_clock <clock name>`

Used along with -to_clock to get the list of paths of the inter-clock domain between the two clocks.

`-to_clock <clock name>`

Used along with -from_clock to get the list of paths of the inter-clock domain between the two clocks.

`-in_to_out`

Used to get the list of path between input and output ports.

`-from <port/pin pattern>`

Filter the list of paths to those starting from ports or pins matching the pattern.

`-to <port/pin pattern>`

Filter the list of paths to those ending at ports or pins matching the pattern.

Example

The following command displays the list of register to register paths of clock domain clk1:

```
puts [ list_paths -clock clk1 -type reg_to_reg ]
```

See Also

create_set
expand_path
set_options

list_scenarios

Tcl command; returns a list of names of all of the available timing scenarios.

```
list_scenarios
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

```
list_scenarios
```

See Also

get_current_scenario
Tcl documentation conventions
Designer Tcl Command Reference

read_sdc

The read_sdc Tcl command evaluate an SDC file, adding all constraints to the specified scenario (or the current/default one if none is specified). Existing constraints are removed if -add is not specified.

```
read_sdc
-add
-scenario scenario_name
-netlist (user | optimized)
-pin_separator (: | /)
file name
```

Arguments

`-add`

Specifies that the constraints from the SDC file will be added on top of the existing ones, overriding them in case of a conflict. If not used, the existing constraints are removed before the SDC file is read.

`-scenario` *scenario_name*

Specifies the scenario to add the constraints to. The scenario is created if none exists with this name.

`-netlist (`*user* | *optimized*`)`

Specifies whether the SDC file contains object defined at the post-synthesis netlist (user) level or physical (optimized) netlist (used for timing analysis).

`-pin_separator` *sep*

Specify the pin separator used in the SDC file. It can be either ':' or '/'.

file name

Specify the SDC file name.

Example

The following command removes all constraints from the current/default scenario and adds all constraints from design.sdc file to it:

```
read_sdc design.sdc
```

See Also

write_sdc

remove_all_constraints

Tcl command; removes all timing constraints from analysis.

```
remove_all_constraints
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
remove_all_constraints
```

See Also

remove_scenario

remove_clock

Tcl command; removes the specified clock constraint from the current timing scenario.

```
remove_clock -name clock_name| -id constraint_ID
```

Arguments

`-name clock_name`

Specifies the name of the clock constraint to remove from the current scenario. You must specify either a clock name or an ID.

`-id constraint_ID`

Specifies the ID of the clock constraint to remove from the current scenario. You must specify either an ID or a clock name that exists in the current scenario.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes the specified clock constraint from the current scenario. If the specified name does not match a clock constraint in the current scenario, or if the specified ID does not refer to a clock constraint, this command fails.

Do not specify both the name and the ID.

Exceptions

You cannot use wildcards when specifying a clock name.

Examples

The following example removes the clock constraint named "my_user_clock":

```
remove_clock -name my_user_clock
```

The following example removes the clock constraint using its ID:

```
set clockId [create_clock –name my_user_clock –period 2]
remove_clock –id $clockId
```

See Also

create_clock

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_clock_latency

Tcl command; removes a clock source latency from the specified clock and from all edges of the clock.

```
remove_clock_latency {-source clock_name_or_source |-id constraint_ID}
```

Arguments

`-source` *`clock_name_or_source`*

Specifies either the clock name or source name of the clock constraint from which to remove the clock source latency. You must specify either a clock or source name or its constraint ID.

`-id` *`constraint_ID`*

Specifies the ID of the clock constraint to remove from the current scenario. You must specify either a clock or source name or its constraint ID.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a clock source latency from the specified clock in the current scenario. If the specified source does not match a clock with a latency constraint in the current scenario, or if the specified ID does not refer to a clock with a latency constraint, this command fails.

Do not specify both the source and the ID.

Exceptions

You cannot use wildcards when specifying a clock name.

Examples

The following example removes the clock source latency from the specified clock.

```
remove_clock_latency -source my_clock
```

See Also

set_clock_latency

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_clock_uncertainty

Tcl command; removes a clock-to-clock uncertainty from the current timing scenario by specifying either its exact arguments or its ID.

```
remove_clock_uncertainty -from | -rise_from | -fall_from from_clock_list -to | -rise_to| -fall_to to_clock_list -setup {value} -hold {value}
remove_clock_uncertainty -id constraint_ID
```

Arguments

`-from`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

`-rise_from`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

`-fall_from`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

`from_clock_list`

Specifies the list of clock names as the uncertainty source.

`-to`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`-rise_to`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`-fall_to`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`to_clock_list`

Specifies the list of clock names as the uncertainty destination.

`-setup`

Specifies that the uncertainty applies only to setup checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

`-hold`

Specifies that the uncertainty applies only to hold checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

`-id` *`constraint_ID`*

Specifies the ID of the clock constraint to remove from the current scenario. You must specify either the exact parameters to set the constraint or its constraint ID.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a clock-to-clock uncertainty from the specified clock in the current scenario. If the specified arguments do not match clocks with an uncertainty constraint in the current scenario, or if the specified ID does not refer to a clock-to-clock uncertainty constraint, this command fails.

Do not specify both the exact arguments and the ID.

Examples

```
remove_clock_uncertainty -from Clk1 -to Clk2
remove_clock_uncertainty -from Clk1 -fall_to { Clk2 Clk3 } -setup
remove_clock_uncertainty 4.3 -fall_from { Clk1 Clk2 } -rise_to *
remove_clock_uncertainty 0.1 -rise_from [ get_clocks { Clk1 Clk2 } ] -fall_to { Clk3 Clk4 } -setup
remove_clock_uncertainty 5 -rise_from Clk1 -to [ get_clocks {*} ]
remove_clock_uncertainty -id $clockId
```

See Also

remove_clock

remove_generated_clock

set_clock_uncertainty

Designer Tcl Command Reference

remove_disable_timing

Tcl command; removes a disable timing constraint by specifying its arguments, or its ID. If the arguments do not match a disable timing constraint, or if the ID does not refer to a disable timing constraint, the command fails.

```
remove_disable_timing -from value -to value name -id name
```

Arguments

`-from` *`from_port`*

Specifies the starting port. The –from and –to arguments must either both be present or both omitted for the constraint to be valid.

`-to` *`to_port`*

Specifies the ending port. The –from and –to arguments must either both be present or both omitted for the constraint to be valid.

`name`

Specifies the cell name where the disable timing constraint will be removed. It is an error to supply both a cell name and a constraint ID, as they are mutually exclusive. No wildcards are allowed when specifying a clock name, either alone or in an accessor command1.

`-id` *`name`*

Specifies the constraint name where the disable timing constraint will be removed. It is an error to supply both a cell name and a constraint ID, as they are mutually exclusive. No wildcards are allowed when specifying a clock name, either alone or in an accessor command1.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
remove_disable_timing -from port1 -to port2 -id new_constraint
```

Designer Tcl Command Reference

remove_false_path

Tcl command; removes a false path from the current timing scenario by specifying either its exact arguments or its ID.

```
remove_false_path [-from from_list] [-to to_list] [-through through_list] [-id constraint_ID]
remove_false_path -id constraint_ID
```

Arguments

`-from from_list`

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through through_list`

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

`-to to_list`

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

`-id constraint_ID`

Specifies the ID of the false path constraint to remove from the current scenario. You must specify either the exact false path to remove or the constraint ID that refers to the false path constraint to remove.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a false path from the specified clock in the current scenario. If the arguments do not match a false path constraint in the current scenario, or if the specified ID does not refer to a false path constraint, this command fails.

Do not specify both the false path arguments and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an Accessor command such as get_pins or get_ports.

Examples

The following example specifies all false paths to remove:

```
remove_false_path -through U0/U1:Y
```

The following example removes the false path constraint using its id:

```
set fpId [set_false_path –from [get_clocks c*] –through {topx/reg/*} –to [get_ports out15] ]
remove_false_path –id $fpId
```

See Also

set_false_path

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_generated_clock

Tcl command; removes the specified generated clock constraint from the current scenario.

```
remove_generated_clock {-name clock_name | -id constraint_ID }
```

Arguments

`-name clock_name`

Specifies the name of the generated clock constraint to remove from the current scenario. You must specify either a clock name or an ID.

`-id constraint_ID`

Specifies the ID of the generated clock constraint to remove from the current scenario. You must specify either an ID or a clock name that exists in the current scenario.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes the specified generated clock constraint from the current scenario. If the specified name does not match a generated clock constraint in the current scenario, or if the specified ID does not refer to a generated clock constraint, this command fails.

Do not specify both the name and the ID.

Exceptions

You cannot use wildcards when specifying a generated clock name.

Examples

The following example removes the generated clock constraint named "my_user_clock":

```
remove_generated_clock -name my_user_clock
```

See Also

create_generated_clock

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_input_delay

Tcl command; removes an input delay a clock on a port by specifying both the clocks and port names or the ID of the input_delay constraint to remove.

```
remove_input_delay -clock clock_name port_pin_list
remove_input_delay -id constraint_ID
```

Arguments

`-clock clock_name`

Specifies the clock name to which the specified input delay value is assigned.

`port_pin_list`

Specifies the port names to which the specified input delay value is assigned.

`-id constraint_ID`

Specifies the ID of the clock with the input_delay value to remove from the current scenario. You must specify either both a clock name and list of port names or the input_delay constraint ID .

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes an input delay from the specified clocks and port in the current scenario. If the clocks and port names do not match an input delay constraint in the current scenario, or if the specified ID does not refer to an input delay constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command.

Examples

The following example removes the input delay from CLK1 on port data1:

```
remove_input_delay -clock [get_clocks CLK1] [get_ports data1]
```

See Also

set_input_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_max_delay

Tcl command; removes a maximum delay constraint from the current timing scenario by specifying either its exact arguments or its ID.

```
remove_max_delay [-from from_list] [-to to_list] [-through through_list]
remove_max_delay -id constraint_ID
```

Arguments

-from *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

-through *through_list*

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

-to to_list

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

-id *constraint_ID*

Specifies the ID of the maximum delay constraint to remove from the current scenario. You must specify either the exact maximum delay arguments to remove or the constraint ID that refers to the maximum delay constraint to remove.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a maximum delay value from the specified clock in the current scenario. If the arguments do not match a maximum delay constraint in the current scenario, or if the specified ID does not refer to a maximum delay constraint, this command fails.

Do not specify both the maximum delay arguments and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an Accessor command.

Examples

The following example specifies a range of maximum delay constraints to remove:

```
remove_max_delay -through U0/U1:Y
```

See Also

set_max_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_min_delay

Tcl command; removes a minimum delay constraint in the current timing scenario by specifying either its exact arguments or its ID.

```
remove_min_delay [-from from_list] [-to to_list] [-through through_list]
remove_min_delay -id constraint_ID
```

Arguments

`-from from_list`

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through through_list`

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

`-toto_list`

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

`-id constraint_ID`

Specifies the ID of the minimum delay constraint to remove from the current scenario. You must specify either the exact minimum delay arguments to remove or the constraint ID that refers to the minimum delay constraint to remove.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a minimum delay value from the specified clock in the current scenario. If the arguments do not match a minimum delay constraint in the current scenario, or if the specified ID does not refer to a minimum delay constraint, this command fails.

Do not specify both the minimum delay arguments and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command.

Examples

The following example specifies a range of minimum delay constraints to remove:

```
remove_min_delay -through U0/U1:Y
```

See Also

set_min_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_multicycle_path

Tcl command; removes a multicycle path constraint in the current timing scenario by specifying either its exact arguments or its ID.

```
remove_multicycle_path [-from from_list] [-to to_list] [-through through_list]
remove multicycle_path -id constraint_ID
```

Arguments

`-from from_list`

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through through_list`

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

`-toto_list`

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

`-id constraint_ID`

Specifies the ID of the multicycle path constraint to remove from the current scenario. You must specify either the exact multicycle path arguments to remove or the constraint ID that refers to the multicycle path constraint to remove.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes a multicycle path from the specified clock in the current scenario. If the arguments do not match a multicycle path constraint in the current scenario, or if the specified ID does not refer to a multicycle path constraint, this command fails.

Do not specify both the multicycle path arguments and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command.

Examples

The following example removes all paths between reg1 and reg2 to 3 cycles for setup check.

```
remove_multicycle_path -from [get_pins {reg1}] –to [get_pins {reg2}]
```

See Also

set_multicycle_path

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_output_delay

Tcl command; removes an ouput delay by specifying both the clocks and port names or the ID of the output_delay constraint to remove.

```
remove_output_delay -clock clock_name port_pin_list
remove_output_delay -id constraint_ID
```

Arguments

`-clock clock_name`

Specifies the clock name to which the specified output delay value is assigned.

port_pin_list

Specifies the port names to which the specified output delay value is assigned.

`-id constraint_ID`

Specifies the ID of the clock with the output_delay value to remove from the current scenario. You must specify either both a clock name and list of port names or the output_delay constraint ID .

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Removes an output delay from the specified clocks and port in the current scenario. If the clocks and port names do not match an output delay constraint in the current scenario, or if the specified ID does not refer to an output delay constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command.

Examples

The following example removes the output delay from CLK1 on port out1:

```
remove_output_delay -clock [get_clocks CLK1] [get_ports out1]
```

See Also

set_output_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

remove_scenario

Tcl command; removes a scenario from the constraint database.

```
remove_scenario <name>
```

Arguments

name

Specifies the name of the scenario to delete.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

The following command removes the scenario named my_scenario:

```
remove_scenario my_scenario
```

See Also

create_scenario

remove_set

Tcl command; removes a set of paths from analysis. Only user-created sets can be deleted.

```
remove_set -name name
```

Parameters

-name *name*

Specifies the name of the set to delete.

Example

The following command removes the set named my_set:

```
remove_set -name my_set
```

See Also

create_set

rename_scenario (SmartFusion2, IGLOO2, and RTG4)

Tcl command; renames an existing timing scenario to a new name. You must provide a unique name (that is, it cannot already be used by another timing scenario) for the new name.

```
rename_scenario old_name  new_name
```

Arguments

old_name

Specifies the name of the existing timing scenario to be renamed.

new_name

Specifies the new name for the scenario.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command renames an existing scenario name to a new name..

Example

```
rename_scenario my_old_scenario my_new_scenario
```

See Also

create_scenario
delete_scenario
Tcl documentation conventions
Designer Tcl Command Reference

save

Tcl command; saves all changes made prior to this command. This includes changes made on constraints, options and sets.

```
save
```

Arguments

None

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

The following script sets the maximum number of paths reported by list_paths to 10, reads an SDC file, and save both the option and the constraints into the design project:

```
set_options -limit_max_paths 10
read_sdc somefile.sdc
save
```

See Also

set_options

set_clock_latency

Tcl command; defines the delay between an external clock source and the definition pin of a clock within SmartTime.

```
set_clock_latency -source [-rise][-fall][-early][-late] delay clock
```

Arguments

-source

Specifies the source latency on a clock pin, potentially only on certain edges of the clock.

-rise

Specifies the edge for which this constraint will apply. If neither or both rise are passed, the same latency is applied to both edges.

`-fall`

Specifies the edge for which this constraint will apply. If neither or both rise are passed, the same latency is applied to both edges.

`-invert`

Specifies that the generated clock waveform is inverted with respect to the reference clock.

`-late`

Optional. Specifies that the latency is late bound on the latency. The appropriate bound is used to provide the most pessimistic timing scenario. However, if the value of "-late" is less than the value of "-early", optimistic timing takes place which could result in incorrect analysis. If neither or both "-early" and "-late" are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.

`-early`

Optional. Specifies that the latency is early bound on the latency. The appropriate bound is used to provide the most pessimistic timing scenario. However, if the value of "-late" is less than the value of "-early", optimistic timing takes place which could result in incorrect analysis. If neither or both "-early" and "-late" are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.

`delay`

Specifies the latency value for the constraint.

`clock`

Specifies the clock to which the constraint is applied. This clock must be constrained.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Clock source latency defines the delay between an external clock source and the definition pin of a clock within SmartTime. It behaves much like an input delay constraint. You can specify both an "early" delay and a"late" delay for this latency, providing an uncertainty which SmartTime propagates through its calculations. Rising and falling edges of the same clock can have different latencies. If only one value is provided for the clock source latency, it is taken as the exact latency value, for both rising and falling edges.

Examples

The following example sets an early clock source latency of 0.4 on the rising edge of main_clock. It also sets a clock source latency of 1.2, for both the early and late values of the falling edge of main_clock. The late value for the clock source latency for the falling edge of main_clock remains undefined.

```
set_clock_latency –source –rise –early 0.4 { main_clock  }
set_clock_latency –source –fall    1.2 { main_clock }
```

See Also

`create_clock`

`create_generated_clock`

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_clock_to_output

SDC command; defines the timing budget available inside the FPGA for an output relative to a clock.

```
set_clock_to_output delay_value -clock clock_ref [–max] [–min] output_list
```

Arguments

delay_value

Specifies the clock to output delay in nanoseconds. This time represents the amount of time available inside the FPGA between the active clock edge and the data change at the output port.

`-clock` *clock_ref*

Specifies the reference clock to which the specified clock to output is related. This is a mandatory argument.

`-max`

Specifies that *delay_value* refers to the maximum clock to output at the specified output. If you do not specify –max or –min options, the tool assumes maximum and minimum clock to output delays to be equal.

`-min`

Specifies that *delay_value* refers to the minimum clock to output at the specified output. If you do not specify –max or –min options, the tool assumes maximum and minimum clock to output delays to be equal.

output_list

Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

set_clock_uncertainty

Tcl command; specifies a clock-to-clock uncertainty between two clocks (from and to) and returns the ID of the created constraint if the command succeeded.

```
set_clock_uncertainty uncertainty -from | -rise_from | -fall_from from_clock_list -to | -
rise_to | -fall_to to_clock_list -setup {value} -hold {value}
```

Arguments

uncertainty

Specifies the time in nanoseconds that represents the amount of variation between two clock edges.

`-from`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

`-rise_from`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

`-fall_from`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.

from_clock_list

Specifies the list of clock names as the uncertainty source.

`-to`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`-rise_to`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`-fall_to`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. Only one of the `-to`, `-rise_to` , or `-fall_to` arguments can be specified for the constraint to be valid.

`to_clock_list`

Specifies the list of clock names as the uncertainty destination.

`-setup`

Specifies that the uncertainty applies only to setup checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

`-hold`

Specifies that the uncertainty applies only to hold checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

The set_clock_uncertainty command sets the timing uncertainty between two clock waveforms or maximum clock skew. Timing between clocks have no uncertainty unless you specify it.

Examples

```
set_clock_uncertainty 10 -from Clk1 -to Clk2
set_clock_uncertainty 0 -from Clk1 -fall_to { Clk2 Clk3 } -setup
set_clock_uncertainty 4.3 -fall_from { Clk1 Clk2 } -rise_to *
set_clock_uncertainty 0.1 -rise_from [ get_clocks { Clk1 Clk2 } ] -fall_to { Clk3 Clk4 }
-setup
set_clock_uncertainty 5 -rise_from Clk1 -to [ get_clocks {*} ]
```

See Also

`create_clock`

`create_generated_clock`

`remove_clock_uncertainty`

Designer Tcl Command Reference

set_current_scenario

Tcl command; specifies the timing scenario for the Timing Analyzer to use. All commands that follow this command will apply to the specified timing scenario.

```
set_current_scenario name
```

Arguments

`name`

Specifies the name of the timing scenario to which to apply all commands from this point on.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

A timing scenario is a set of timing constraints used with a design. If the specified scenario is already the current one, this command has no effect.

After setting the current scenario, constraints can be listed, added, or removed, the checker can be invoked on the set of constraints, and so on.

This command uses the specified timing scenario to compute timing analysis.

Example

```
set_current_scenario scenario_A
```

See Also

get_current_scenario

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_disable_timing

Tcl command; disables timing arcs within a cell and returns the ID of the created constraint if the command succeeded.

```
set_disable_timing -from value -to value name
```

Arguments

`-from` *`from_port`*

Specifies the starting port. The –from and –to arguments must either both be present or both omitted for the constraint to be valid.

`-to` *`to_port`*

Specifies the ending port. The –from and –to arguments must either both be present or both omitted for the constraint to be valid.

`name`

Specifies the cell name where the timing arcs will be disabled.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Example

```
set_disable_timing -from A -to Y a2
```

See Also

Tcl documentation conventions

Designer Tcl Command Reference

set_external_check

SDC command; defines the external setup and hold delays for an input relative to a clock.

```
set_external_check delay_value -clock clock_ref [–setup] [–hold] [–clock_fall] input_list
```

Arguments

delay_value

Specifies the external setup or external hold delay in nanoseconds. This time represents the amount of time available inside the FPGA for the specified input after a clock edge.

-clock *clock_ref*

Specifies the reference clock to which the specified external check is related. This is a mandatory argument.

-setup

Specifies that *delay_value* refers to the setup check at the specified input. This is a mandatory argument if –hold is not used. You must specify either the -setup or -hold option.

-clock_fall

Specifies that the delay is relative to the falling edge of the reference clock. The default is the rising edge.

input_list

Provides a list of input ports in the current design to which *delay_value* is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

set_false_path

Tcl command; identifies paths that are considered false and excluded from the timing analysis in the current timing scenario.

```
set_false_path [-from from_list] [-through through_list] [-to to_list]
```

Arguments

-from *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

-through *through_list*

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

-to *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

The set_false_path command identifies specific timing paths as being false. The false timing paths are paths that do not propagate logic level changes. This constraint removes timing requirements on these false paths so that they are not considered during the timing analysis. The path starting points are the input ports or register clock pins, and the path ending points are the register data pins or output ports. This constraint disables setup and hold checking for the specified paths.

The false path information always takes precedence over multiple cycle path information and overrides maximum delay constraints. If more than one object is specified within one -through option, the path can pass through any objects.

You must specify at least one of the `–from`, `-to`, or `–through` arguments for this constraint to be valid.

Examples

The following example specifies all paths from clock pins of the registers in clock domain clk1 to data pins of a specific register in clock domain clk2 as false paths:

```
set_false_path –from [get_clocks {clk1}] –to reg_2:D
```

The following example specifies all paths through the pin U0/U1:Y to be false:

```
set_false_path -through U0/U1:Y
```

See Also

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_input_delay

Tcl command; creates an input delay on a port list by defining the arrival time of an input relative to a clock in the current scenario.

```
set_input_delay delay_value -clock clock_ref [–max] [–min] [–clock_fall] input_list
```

Arguments

`delay_value`

Specifies the arrival time in nanoseconds that represents the amount of time for which the signal is available at the specified input after a clock edge.

`-clock` *`clock_ref`*

Specifies the clock reference to which the specified input delay is related. This is a mandatory argument. If you do not specify -max or -min options, the tool assumes the maximum and minimum input delays to be equal.

`-max`

Specifies that delay_value refers to the longest path arriving at the specified input. If you do not specify -max or -min options, the tool assumes maximum and minimum input delays to be equal.

`-min`

Specifies that delay_value refers to the shortest path arriving at the specified input. If you do not specify -max or -min options, the tool assumes maximum and minimum input delays to be equal.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.

`input_list`

Provides a list of input ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.and IGLOOe, except ProASIC3 nano and ProASIC3L

Description

The set_input_delay command sets input path delays on input ports relative to a clock edge. This usually represents a combinational path delay from the clock pin of a register external to the current design. For

in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds input delay to path delay for paths starting at primary inputs.

A clock is a singleton that represents the name of a defined clock constraint. This can be:

- a single port name used as source for a clock constraint
- a single pin name used as source for a clock constraint; for instance reg1:CLK. This name can be hierarchical (for instance toplevel/block1/reg2:CLK)
- an object accessor that will refer to one clock: [get_clocks {clk}]

Examples

The following example sets an input delay of 1.2ns for port data1 relative to the rising edge of CLK1:

```
set_input_delay 1.2 -clock [get_clocks CLK1] [get_ports data1]
```

The following example sets a different maximum and minimum input delay for port IN1 relative to the falling edge of CLK2:

```
set_input_delay 1.0 -clock_fall -clock CLK2 –min {IN1}
set_input_delay 1.4 -clock_fall -clock CLK2 –max {IN1}
```

See Also

set_output_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_max_delay

Tcl command; specifies the maximum delay for the timing paths in the current scenario.

```
set_max_delay delay_value [-from from_list] [-to to_list] [-through through_list]
```

Arguments

delay_value

Specifies a floating point number in nanoseconds that represents the required maximum delay value for specified paths.

- If the path starting point is on a sequential device, the tool includes clock skew in the computed delay.
- If the path starting point has an input delay specified, the tool adds that delay value to the path delay.
- If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay.
- If the ending point has an output delay specified, the tool adds that delay to the path delay.

`-from` *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-to` *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

`-through` *through_list*

Specifies a list of pins, ports, cells, or nets through which the timing paths must pass.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command specifies the required maximum delay for timing paths in the current design. The path length for any startpoint in from_list to any endpoint in to_list must be less than delay_value.

The timing engine automatically derives the individual maximum delay targets from clock waveforms and port input or output delays.

The maximum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multicycle path constraint.

You must specify at least one of the `–from`, `-to`, or `–through` arguments for this constraint to be valid.

Examples

The following example sets a maximum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay less than 5 ns:

```
set_max_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```

The following example sets a maximum delay by constraining all paths to output ports whose names start by “out” with a delay less than 3.8 ns:

```
set_max_delay 3.8 -to [get_ports out*]
```

See Also

set_min_delay

remove_max_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_min_delay

Tcl command; specifies the minimum delay for the timing paths in the current scenario.

```
set_min_delay delay_value [-from from_list] [-to to_list] [-through through_list]
```

Arguments

delay_value

Specifies a floating point number in nanoseconds that represents the required minimum delay value for specified paths.

- If the path starting point is on a sequential device, the tool includes clock skew in the computed delay.
- If the path starting point has an input delay specified, the tool adds that delay value to the path delay.
- If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay.
- If the ending point has an output delay specified, the tool adds that delay to the path delay.

`-from` *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-to` *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

`-through` *through_list*

Specifies a list of pins, ports, cells, or nets through which the timing paths must pass.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

This command specifies the required minimum delay for timing paths in the current design. The path length for any startpoint in from_list to any endpoint in to_list must be less than delay_value.

The timing engine automatically derives the individual minimum delay targets from clock waveforms and port input or output delays.

The minimum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multicycle path constraint.

You must specify at least one of the `–from`, `-to`, or `–through` arguments for this constraint to be valid.

Examples

The following example sets a minimum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay less than 5 ns:

```
set_min_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```

The following example sets a minimum delay by constraining all paths to output ports whose names start by “out” with a delay less than 3.8 ns:

```
set_min_delay 3.8 -to [get_ports out*]
```

See Also

`set_max_delay`

`remove_min_delay`

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_multicycle_path

Tcl command; defines a path that takes multiple clock cycles in the current scenario.

```
set_multicycle_path ncycles [-setup] [-hold] [-from from_list[–through through_list[-to
to_list
```

Arguments

`ncycles`

Specifies an integer value that represents a number of cycles the data path must have for setup or hold check. The value is relative to the starting point or ending point clock, before data is required at the ending point.

`-setup`

Optional. Applies the cycle value for the setup check only. This option does not affect the hold check. The default hold check will be applied unless you have specified another set_multicycle_path command for the hold value.

`-hold`

Optional. Applies the cycle value for the hold check only. This option does not affect the setup check.

Note: If you do not specify "-setup" or "-hold", the cycle value is applied to the setup check and the default hold check is performed (*ncycles* -1).

`-from` *`from_list`*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through` *`through_list`*

Specifies a list of pins or ports through which the multiple cycle paths must pass.

`-to` *`to_list`*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

Setting multiple cycle paths constraint overrides the single cycle timing relationships between sequential elements by specifying the number of cycles that the data path must have for setup or hold checks. If you change the multiplier, it affects both the setup and hold checks.

False path information always takes precedence over multiple cycle path information. A specific maximum delay constraint overrides a general multiple cycle path constraint.

If you specify more than one object within one -through option, the path passes through any of the objects.

You must specify at least one of the `–from`, `-to`, or `–through` arguments for this constraint to be valid.

Exceptions

Multiple priority management is not supported in Microsemi SoC designs. All multiple cycle path constraints are handled with the same priority.

Examples

The following example sets all paths between reg1 and reg2 to 3 cycles for setup check. Hold check is measured at the previous edge of the clock at reg2.

```
set_multicycle_path 3 -from [get_pins {reg1}] –to [get_pins {reg2}]
```

The following example specifies that four cycles are needed for setup check on all paths starting at the registers in the clock domain ck1. Hold check is further specified with two cycles instead of the three cycles that would have been applied otherwise.

```
set_multicycle_path 4 -setup -from [get_clocks {ck1}]
set_multicycle_path 2 -hold -from [get_clocks {ck1}]
```

See Also

`remove_multicycle_path`

Tcl Command Documentation Conventions

Designer Tcl Command Reference

set_options (SmartFusion2, IGLOO2, RTG4)

SmartTime-specific Tcl command; sets options for timing analysis. Some options will also affect timing-driven place-and-route. The same parameters can be changed in the SmartTime Options dialog box in the SmartTime GUI.

```
set_options
 [-max_opcond value ]
 [-min_opcond value]
 [-interclockdomain_analysis value]
 [-use_bibuf_loopbacks value]
```

```
[-enable_recovery_removal_checks value]
[-break_at_async value]
[-filter_when_slack_below value]
[-filter_when_slack_above value]
[-remove_slack_filters]
[-limit_max_paths value]
[-expand_clock_network value]
[-expand_parallel_paths value]
[-analysis_scenario value]
[-tdpr_scenario value]
[-reset]
```

Arguments

-max_opcond *value*

Sets the operating condition to use for Maximum Delay Analysis. The following table shows the acceptable values for this argument. Default is *worst*.

Value	Description
worst	Use Worst Case conditions for Maximum Delay Analysis
typical	Use Typical conditions for Maximum Delay Analysis
best	Use Best Case conditions for Maximum Delay Analysis

-min_opcond *value*

Sets the operating condition to use for Minimum Delay Analysis. The following table shows the acceptable values for this argument. Default is *best*.

Value	Description
best	Use Best Case conditions for Minimum Delay Analysis
typical	Use Typical conditions for Minimum Delay Analysis
worst	Use Worst Case conditions for Minimum Delay Analysis

-interclockdomain_analysis *value*

Enables or disables inter-clock domain analysis. Default is *yes*.

Value	Description
yes	Enables inter-clock domain analysis
no	Disables inter-clock domain analysis

-use_bibuf_loopbacks *value*

Instructs the timing analysis whether to consider loopback path in bidirectional buffers (D->Y, E->Y)as false-path {no}. Default is *yes*; i.e., loopback are false paths.

Value	Description
yes	Enables loopback in bibufs

Value	Description
no	Disables loopback in bibufs

-enable_recovery_removal_checks *value*

Enables recovery checks to be included in max-delay analysis and removal checks in min-delay analysis. Default is *yes*.

Value	Description
yes	Enables recovery and removal checks
no	Disables recovery and removal checks

-break_at_async *value*

Specifies whether or not timing analysis is allowed to cross asynchronous pins (clear, reset of sequential elements). Default is *no*.

Value	Description
yes	Enables breaking paths at asynchronous ports
no	Disables breaking paths at asynchronous ports

-filter_when_slack_below *value*

Specifies a minimum slack value for paths reported by list_paths. Not set by default.

-filter_when_slack_above *value*

Specifies a maximum slack value for paths reported by list_paths. Not set by default.

-remove_slack_filters

Removes the slack minimum and maximum set using -filter_when_slack_below and filter_when_slack_above.

-limit_max_paths *value*

Specifies the maximum number of paths reported by list_paths. Default is *100*.

-expand_clock_network *value*

Specify whether or not clock network details are reported in expand_path. Default is *yes*.

Value	Description
yes	Enables expanded clock network information in paths
no	Disables expanded clock network information in paths

-expand_parallel_paths *value*

Specify the number of parallel paths {paths with the same ends} to include in expand_path. Default is *1*.

-analysis_scenario *value*

Specify the constraint scenario to be used for timing analysis. Default is *Primary*, the default scenario.

-tdpr_scenario *value*

Specify the constraint scenario to be used for timing-driven place-and-route. Default is Primary, the default scenario.

-reset

Reset all options to the default values, except those for analysis and TDPR scenarios, which remain unchanged.

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Examples

The following script commands the timing engine to use best operating conditions for both max-delay analysis and min-delay analysis:

```
set_options -max_opcond {best} -min_opcond {best}
```

The following script changes the scenario used by timing-driven place-and-route and saves the change in the Libero project for place-and-route tools to see the change.

```
set_options -tdpr_scenario {My_TDPR_Scenario}
```

See Also

save

set_output_delay

Tcl command; defines the output delay of an output relative to a clock in the current scenario.

```
set_output_delay [-max] [-min] delay_value -clock clock_ref [-clock_fall] output_list
```

Arguments

`-max`

Specifies that delay_value refers to the longest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.

`-min`

Specifies that delay_value refers to the shortest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.

`delay_value`

Specifies the amount of time before a clock edge for which the signal is required. This represents a combinational path delay to a register outside the current design plus the library setup time (for maximum output delay) or hold time (for minimum output delay).

`-clock` *`clock_ref`*

Specifies the clock reference to which the specified output delay is related. This is a mandatory argument. If you do not specify -max or -min options, the tool assumes the maximum and minimum input delays to be equal.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.

`output_list`

Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

See the Tcl Commands and Supported Families table for the list of families that support this command.

Description

The set_output_delay command sets output path delays on output ports relative to a clock edge. Output ports have no output delay unless you specify it. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds output delay to path delay for paths ending at primary outputs.

Examples

The following example sets an output delay of 1.2ns for port OUT1 relative to the rising edge of CLK1:

```
set_output_delay 1.2 -clock [get_clocks CLK1] [get_ports OUT1]
```

The following example sets a different maximum and minimum output delay for port OUT1 relative to the falling edge of CLK2:

```
set_output_delay –min {OUT1} 1.0 -clock_fall -clock CLK2
set_output_delay –max {OUT1} 1.4 -clock_fall -clock CLK2
```

See Also

remove_output_delay

set_input_delay

Tcl Command Documentation Conventions

Designer Tcl Command Reference

write_sdc

Tcl command; writes timing constraints into an SDC file. If multiple constraint scenarios are defined, -scenario allows the user to specify which scenario to write. By default, the current scenario is written.

```
write_sdc
-scenario scenario name
-pin_separator (: | /)
file name
```

Arguments

`-scenario` *scenario name*

Specify the scenario to write. By default the current scenario is used.

`-pin_separator` *sep*

Specify the pin separator used in the SDC file. It can be either ':' or '/'.

file name

Specify the SDC file name.

Example

The following script merges two SDC files and writes the result into a third SDC file:

```
read_sdc first.sdc
read_sdc -add second.sdc
write_sdc merged.sdc
```

See Also

read_sdc, set_current_scenario

VERIFYTIMING (SmartFusion2 , IGLOO2, RTG4,)

Constraints by File Format - SDC Command Reference

About Synopsys Design Constraints (SDC) Files

Synopsys Design Constraints (SDC) is a Tcl-based format used by Synopsys tools to specify the design intent, including the timing and area constraints for a design. Microsemi tools use a subset of the SDC format to capture supported timing constraints. Any timing constraint that you can enter using Designer tools can also be specified in an SDC file.

Use the SDC-based flow to share timing constraint information between Microsemi tools and third-party EDA tools.

Command	Action
create_clock	Creates a clock and defines its characteristics
create_generated_clock	Creates an internally generated clock and defines its characteristics
remove_clock_uncertainty	Removes a clock-to-clock uncertainty from the current timing scenario.
set_clock_latency	Defines the delay between an external clock source and the definition pin of a clock within SmartTime
set_clock_uncertainty	Defines the timing uncertainty between two clock waveforms or maximum skew
set_false_path	Identifies paths that are to be considered false and excluded from the timing analysis
set_input_delay	Defines the arrival time of an input relative to a clock
set_load	Sets the load to a specified value on a specified port
set_max_delay	Specifies the maximum delay for the timing paths
set_min_delay	Specifies the minimum delay for the timing paths
set_multicycle_path	Defines a path that takes multiple clock cycles
set_output_delay	Defines the output delay of an output relative to a clock

See Also

Constraint Entry

SDC Syntax Conventions

Importing Constraint Files

SDC Syntax Conventions

The following table shows the typographical conventions that are used for the SDC command syntax.

Syntax Notation	Description
command -argument	Commands and arguments appear in `Courier New` typeface.
`variable`	Variables appear in blue, italic *`Courier New`* typeface. You must substitute an appropriate value for the variable.
[-argument *`value`*]	Optional arguments begin and end with a square bracket.

Note: SDC commands and arguments are case sensitive.

Example

The following example shows syntax for the create_clock command and a sample command:

```
create_clock -period period_value [-waveform edge_list] source
```

```
create_clock –period 7 –waveform {2 4}{CLK1}
```

Wildcard Characters

You can use the following wildcard characters in names used in the SDC commands:

Wildcard	What it does
\	Interprets the next character literally
*	Matches any string

Note: The matching function requires that you add a backslash (\) before each slash in the pin names in case the slash does not denote the hierarchy in your design.

Special Characters ([ ], { }, and \)

Square brackets ([ ]) are part of the command syntax to access ports, pins and clocks. In cases where these netlist objects names themselves contain square brackets (for example, buses), you must either enclose the names with curly brackets ({}) or precede the open and closed square brackets ([ ]) characters with a backslash (\). If you do not do this, the tool displays an error message.

For example:

```
create_clock -period 3 clk\[0\]
set_max_delay 1.5 -from [get_pins ff1\[5\]:CLK] -to [get_clocks {clk[0]}]
```

Although not necessary, Microsemi recommends the use of curly brackets around the names, as shown in the following example:

```
set_false_path –from {data1} –to [get_pins {reg1:D}]
```

In any case, the use of the curly bracket is mandatory when you have to provide more than one name.

For example:

```
set_false_path –from {data3 data4} –to [get_pins {reg2:D reg5:D}]
```

Entering Arguments on Separate Lines

If a command needs to be split on multiple lines, each line except the last must end with a backslash (\) character as shown in the following example:

```
set_multicycle_path 2 –from \
[get_pins {reg1*}] \
-to {reg2:D}
```

See Also

About SDC Files

Referenced Topics

create_clock

SDC command; creates a clock and defines its characteristics.

```
create_clock -name name -period period_value [-waveform edge_list]  source
```

Arguments

`-name name`

Specifies the name of the clock constraint. This parameter is required for virtual clocks when no clock source is provided.

`-period period_value`

Specifies the clock period in nanoseconds. The value you specify is the minimum time over which the clock waveform repeats. The period_value must be greater than zero.

`-waveform edge_list`

Specifies the rise and fall times of the clock waveform in ns over a complete clock period. There must be exactly two transitions in the list, a rising transition followed by a falling transition. You can define a clock starting with a falling edge by providing an edge list where fall time is less than rise time. If you do not specify -waveform option, the tool creates a default waveform, with a rising edge at instant 0.0 ns and ©a falling edge at instant (period_value/2)ns.

source

Specifies the source of the clock constraint. The source can be ports or pins in the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing one. Only one source is accepted. Wildcards are accepted as long as the resolution shows one port or pin.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Creates a clock in the current design at the declared source and defines its period and waveform. The static timing analysis tool uses this information to propagate the waveform across the clock network to the clock pins of all sequential elements driven by this clock source.

The clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

Exceptions

- None

Examples

The following example creates two clocks on ports CK1 and CK2 with a period of 6, a rising edge at 0, and a falling edge at 3:

```
create_clock -name {my_user_clock} -period 6 CK1
create_clock -name {my_other_user_clock} –period 6 –waveform {0 3} {CK2}
```

The following example creates a clock on port CK3 with a period of 7, a rising edge at 2, and a falling edge at 4:

```
create_clock –period 7 –waveform {2 4} [get_ports {CK3}]
```

Microsemi Implementation Specifics

- The -waveform in SDC accepts waveforms with multiple edges within a period. In Microsemi design implementation, only two waveforms are accepted.
- SDC accepts defining a clock on many sources using a single command. In Microsemi design implementation, only one source is accepted.
- The source argument in SDC create_clock command is optional. This is in conjunction with the -name argument in SDC to support the concept of virtual clocks. In Microsemi implementation, source is a mandatory argument as -name and virtual clocks concept is not supported.
- The -domain argument in the SDC create_clock command is not supported.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Clock Definition

Create Clock

Create a New Clock Constraint

create_generated_clock

SDC command; creates an internally generated clock and defines its characteristics.

```
create_generated_clock -name {name -source reference_pin [-divide_by divide_factor] [-
multiply_by multiply_factor] [-invert] source -pll_output pll_feedback_clock -pll_feedback
pll_feedback_input
```

Arguments

`-name` *name*

Specifies the name of the clock constraint. This parameter is required for virtual clocks when no clock source is provided.

`-source` *reference_pin*

Specifies the reference pin in the design from which the clock waveform is to be derived.

`-divide_by`*divide_factor*

Specifies the frequency division factor. For instance if the *divide_factor* is equal to 2, the generated clock period is twice the reference clock period.

`-multiply_by` *multiply_factor*

Specifies the frequency multiplication factor. For instance if the *multiply_factor* is equal to 2, the generated clock period is half the reference clock period.

`-invert`

Specifies that the generated clock waveform is inverted with respect to the reference clock.

`source`

Specifies the source of the clock constraint on internal pins of the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing clock. Only one source is accepted. Wildcards are accepted as long as the resolution shows one pin.

`-pll_output` *pll_feedback_clock*

Specifies the output pin of the PLL which is used as the external feedback clock. This pin must drive the feedback input pin of the PLL specified using the –pll_feedback option. The PLL will align the rising edge of the reference input clock to the feedback clock. This is a mandatory argument if the PLL is operating in external feedback mode.

`-pll_feedback` *pll_feedback_input*

Specifies the feedback input pin of the PLL. This pin must be driven by the output pin of the PLL specified using the –pll_output option. The PLL will align the rising edge of the reference input clock to the external feedback clock. This is a mandatory argument if the PLL is operating in external feedback mode.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Creates a generated clock in the current design at a declared source by defining its frequency with respect to the frequency at the reference pin. The static timing analysis tool uses this information to compute and propagate its waveform across the clock network to the clock pins of all sequential elements driven by this source.

The generated clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

Examples

The following example creates a generated clock on pin U1/reg1:Q with a period twice as long as the period at the reference port CLK.

```
create_generated_clock -name {my_user_clock} –divide_by 2 –source [get_ports {CLK}]
U1/reg1/Q
```

The following example creates a generated clock at the primary output of myPLL with a period ¾ of the period at the reference pin clk.

```
create_generated_clock –divide_by 3 –multiply_by 4 -source clk [get_pins {myPLL/CLK1}]
```

The following example creates a generated clock named system_clk on the GL2 output pin of FCCC_0 with a period equal to half the period of the source clock. The constraint also identifies GL2 output pin as the external feedback clock source and CLK2 as the feedback input pin for FCCC_0.

```
create_generated_clock -name { system_clk } \
-multiply_by 2 \
-source { FCCC_0/CCC_INST/CLK3_PAD } \
-pll_output { FCCC_0/CCC_INST/GL2 } \
-pll_feedback { FCCC_0/CCC_INST/CLK2 } \
{ FCCC_0/CCC_INST/GL2 }
```

Microsemi Implementation Specifics

- SDC accepts either –multiply_by or –divide_by option. In Microsemi design implementation, both are accepted to accurately model the PLL behavior.
- SDC accepts defining a generated clock on many sources using a single command. In Microsemi design implementation, only one source is accepted.
- The -duty_cycle ,-edges and –edge_shift options in the SDC create_generated_clock command are not supported in Microsemi design implementation.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Create Generated Clock Constraint (SDC)

remove_clock_uncertainty

SDC command; Removes a clock-to-clock uncertainty from the current timing scenario.

```
remove_clock_uncertainty -from | -rise_from | -fall_from from_clock_list -to | -rise_to| -
fall_to to_clock_list -setup {value} -hold {value}
remove_clock_uncertainty -id constraint_ID
```

Arguments

`-from`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid.

`-rise_from`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid.

`-fall_from`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid.

`from_clock_list`

Specifies the list of clock names as the uncertainty source.

`-to`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`-rise_to`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`-fall_to`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`to_clock_list`

Specifies the list of clock names as the uncertainty destination.

`-setup`

Specifies that the uncertainty applies only to setup checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

`-hold`

Specifies that the uncertainty applies only to hold checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

`-id` *`constraint_ID`*

Specifies the ID of the clock constraint to remove from the current scenario. You must specify either the exact parameters to set the constraint or its constraint ID.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Removes a clock-to-clock uncertainty from the specified clock in the current scenario. If the specified arguments do not match clocks with an uncertainty constraint in the current scenario, or if the specified ID does not refer to a clock-to-clock uncertainty constraint, this command fails.

Do not specify both the exact arguments and the ID.

Exceptions

None

Examples

```
remove_clock_uncertainty -from Clk1 -to Clk2
remove_clock_uncertainty -from Clk1 -fall_to { Clk2 Clk3 } -setup
remove_clock_uncertainty 4.3 -fall_from { Clk1 Clk2 } -rise_to *
remove_clock_uncertainty 0.1 -rise_from [ get_clocks { Clk1 Clk2 } ] -fall_to { Clk3 Clk4 } -setup
remove_clock_uncertainty 5 -rise_from Clk1 -to [ get_clocks {*} ]
remove_clock_uncertainty -id $clockId
```

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

set_clock_uncertainty

set_clock_latency

SDC command; defines the delay between an external clock source and the definition pin of a clock within SmartTime.

```
set_clock_latency -source [-rise][-fall][-early][-late] delay clock
```

Arguments

`-source`

Specifies a clock source latency on a clock pin.

`-rise`

Specifies the edge for which this constraint will apply. If neither or both rise are passed, the same latency is applied to both edges.

`-fall`

Specifies the edge for which this constraint will apply. If neither or both rise are passed, the same latency is applied to both edges.

`-invert`

Specifies that the generated clock waveform is inverted with respect to the reference clock.

`-late`

Optional. Specifies that the latency is late bound on the latency. The appropriate bound is used to provide the most pessimistic timing scenario. However, if the value of "-late" is less than the value of "-early", optimistic timing takes place which could result in incorrect analysis. If neither or both "-early" and "-late" are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.

`-early`

Optional. Specifies that the latency is early bound on the latency. The appropriate bound is used to provide the most pessimistic timing scenario. However, if the value of "-late" is less than the value of "-early", optimistic timing takes place which could result in incorrect analysis. If neither or both "-early" and "-late" are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.

`delay`

Specifies the latency value for the constraint.

`clock`

Specifies the clock to which the constraint is applied. This clock must be constrained.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Clock source latency defines the delay between an external clock source and the definition pin of a clock within SmartTime. It behaves much like an input delay constraint. You can specify both an "early" delay and a"late" delay for this latency, providing an uncertainty which SmartTime propagates through its calculations. Rising and falling edges of the same clock can have different latencies. If only one value is provided for the clock source latency, it is taken as the exact latency value, for both rising and falling edges.

Exceptions

None

Examples

The following example sets an early clock source latency of 0.4 on the rising edge of main_clock. It also sets a clock source latency of 1.2, for both the early and late values of the falling edge of main_clock. The late value for the clock source latency for the falling edge of main_clock remains undefined.

```
set_clock_latency –source –rise –early 0.4 { main_clock  }
set_clock_latency –source –fall    1.2 { main_clock }
```

Microsemi Implementation Specifics

SDC accepts a list of clocks to -set_clock_latency. In Microsemi design implementation, only one clock pin can have its source latency specified per command.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

set_clock_to_output

SDC command; defines the timing budget available inside the FPGA for an output relative to a clock.

```
set_clock_to_output delay_value -clock clock_ref [–max] [–min] [–clock_fall] output_list
```

Arguments

delay_value

Specifies the clock to output delay in nanoseconds. This time represents the amount of time available inside the FPGA between the active clock edge and the data change at the output port.

`-clock` *clock_ref*

Specifies the reference clock to which the specified clock to output is related. This is a mandatory argument.

`-max`

Specifies that delay_value refers to the maximum clock to output at the specified output. If you do not specify –max or –min options, the tool assumes maximum and minimum clock to output delays to be equal.

`-min`

Specifies that delay_value refers to the minimum clock to output at the specified output. If you do not specify –max or –min options, the tool assumes maximum and minimum clock to output delays to be equal.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the reference clock. The default is the rising edge.

`output_list`

Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

The set_clock_to_output command specifies the clock to output maximum and minimum delays on output ports relative to a clock edge. This usually represents a combinational path delay from a register internal to the current design to the output port. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool uses clock to output delays for paths ending at primary outputs.

A clock is a singleton that represents the name of a defined clock constraint. This can be an object accessor that will refer to one clock. For example:

```
[get_clocks {system_clk}]
[get_clocks {sys*_clk}]
```

Examples

The following example sets a maximum clock to output delay of 12 ns and a minimum clock to output delay of 6 ns for port data_out relative to the rising edge of CLK1:

```
set_clock_to_output 12 -clock [get_clocks CLK1] -max [get_ports data_out]
set_clock_to_output 6 -clock [get_clocks CLK1] -min [get_ports data_out]
```

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

set_clock_uncertainty

SDC command; defines the timing uncertainty between two clock waveforms or maximum skew.

```
set_clock_uncertainty uncertainty (-from | -rise_from | -fall_from) from_clock_list (-to | -
rise_to | -fall_to) to_clock_list [-setup | -hold]
```

Arguments

`uncertainty`

Specifies the time in nanoseconds that represents the amount of variation between two clock edges. The value must be a positive floating point number.

`-from`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid. This option is the default.

`-rise_from`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid.

`-fall_from`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the source clock list. You can specify only one of the `-from`, `-rise_from`, or `-fall_from` arguments for the constraint to be valid.

`from_clock_list`

Specifies the list of clock names as the uncertainty source.

`-to`

Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`-rise_to`

Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`-fall_to`

Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. You can specify only one of the `-to`, `-rise_to` , or `-fall_to` arguments for the constraint to be valid.

`to_clock_list`

Specifies the list of clock names as the uncertainty destination.

`-setup`

Specifies that the uncertainty applies only to setup checks. If you do not specify either option (`-setup` or `-hold`) or if you specify both options, the uncertainty applies to both setup and hold checks.

`-hold`

Specifies that the uncertainty applies only to hold checks. If you do not specify either option (`-setup` or `-hold`) or if you specify both options, the uncertainty applies to both setup and hold checks.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Clock uncertainty defines the timing between an two clock waveforms or maximum clock skew.

Both setup and hold checks must account for clock skew. However, for setup check, SmartTime looks for the smallest skew. This skew is computed by using the maximum insertion delay to the launching sequential component and the shortest insertion delay to the receiving component.

For hold check, SmartTime looks for the largest skew. This skew is computed by using the shortest insertion delay to the launching sequential component and the largest insertion delay to the receiving component. SmartTime makes this distinction automatically.

Exceptions

None

Examples

The following example defines two clocks and sets the uncertainty constraints, which analyzes the inter-clock domain between clk1 and clk2.

```
create_clock –period 10 clk1
create_generated_clock –name clk2 -source clk1 -multiply_by 2 sclk1
set_clock_uncertainty 0.4 -rise_from clk1 -rise_to clk2
```

Microsemi Implementation Specifics

- SDC accepts a list of clocks to -set_clock_uncertainty.

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions
create_clock (SDC)
create_generated_clock (SDC)
remove_clock_uncertainty

set_disable_timing

SDC command; disables timing arcs within the specified cell and returns the ID of the created constraint if the command succeeded.

```
set_disable_timing [-from from_port] [-to to_port] cell_name
```

Arguments

`-from from_port`

Specifies the starting port.

`-to to_port`

Specifies the ending port.

`cell_name`

Specifies the name of the cell in which timing arcs will be disabled.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command disables the timing arcs in the specified cell, and returns the ID of the created constraint if the command succeeded. The –from and –to arguments must either both be present or both omitted for the constraint to be valid.

Examples

The following example disables the arc between a2:A and a2:Y.

```
set_disable_timing -from port1 -to port2 cellname
```

This command ensures that the arc is disabled within a cell instead of between cells.

Microsemi Implementation Specifics

- None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

set_external_check

SDC command; defines the external setup and hold delays for an input relative to a clock.

```
set_external_check delay_value -clock clock_ref [–setup] [–hold] [–clock_fall] input_list
```

Arguments

delay_value

Specifies the external setup or external hold delay in nanoseconds. This time represents the amount of time available inside the FPGA for the specified input after a clock edge.

`-clock` *clock_ref*

Specifies the reference clock to which the specified external check is related. This is a mandatory argument.

`-setup`

Specifies that delay_value refers to the setup check at the specified input. This is a mandatory argument if –hold is not used. You must specify either -setup or -hold option.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the reference clock. The default is the rising edge.

input_list

Provides a list of input ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

The set_external_check command specifies the external setup and hold times on input ports relative to a clock edge. This usually represents a combinational path delay from the input port to the clock pin of a register internal to the current design. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool uses external setup and external hold times for paths starting at primary inputs.

A clock is a singleton that represents the name of a defined clock constraint. This can be an object accessor that will refer to one clock. For example:

```
[get_clocks {system_clk}]
[get_clocks {sys*_clk}]
```

Examples

The following example sets an external setup check of 12 ns and an external hold check of 6 ns for port data_in relative to the rising edge of CLK1:

```
set_external_check 12 -clock [get_clocks CLK1] -setup [get_ports data_in]
set_external_check 6 -clock [get_clocks CLK1] -hold [get_ports data_in]
```

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

set_false_path

SDC command; identifies paths that are considered false and excluded from the timing analysis.

```
set_false_path [-from from_list] [-through through_list] [-to to_list]
```

Arguments

`-from from_list`

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through through_list`

Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

`-to to_list`

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

The set_false_path command identifies specific timing paths as being false. The false timing paths are paths that do not propagate logic level changes. This constraint removes timing requirements on these false paths so that they are not considered during the timing analysis. The path starting points are the input ports or register clock pins, and the path ending points are the register data pins or output ports. This constraint disables setup and hold checking for the specified paths.

The false path information always takes precedence over multiple cycle path information and overrides maximum delay constraints. If more than one object is specified within one -through option, the path can pass through any objects.

Examples

The following example specifies all paths from clock pins of the registers in clock domain clk1 to data pins of a specific register in clock domain clk2 as false paths:

```
set_false_path –from [get_clocks {clk1}] –to reg_2:D
```

The following example specifies all paths through the pin U0/U1:Y to be false:

```
set_false_path -through U0/U1:Y
```

Microsemi Implementation Specifics

SDC accepts multiple -through options in a single constraint to specify paths that traverse multiple points in the design. In Microsemi design implementation, only one –through option is accepted.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Set False Path Constraint

set_input_delay

SDC command; defines the arrival time of an input relative to a clock.

```
set_input_delay delay_value -clock clock_ref [–max] [–min] [–clock_fall] input_list
```

Arguments

`delay_value`

Specifies the arrival time in nanoseconds that represents the amount of time for which the signal is available at the specified input after a clock edge.

`-clock clock_ref`

Specifies the clock reference to which the specified input delay is related. This is a mandatory argument. If you do not specify -max or -min options, the tool assumes the maximum and minimum input delays to be equal.

`-max`

Specifies that delay_value refers to the longest path arriving at the specified input. If you do not specify -max or -min options, the tool assumes maximum and minimum input delays to be equal.

`-min`

Specifies that delay_value refers to the shortest path arriving at the specified input. If you do not specify -max or -min options, the tool assumes maximum and minimum input delays to be equal.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.

`input_list`

Provides a list of input ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion and IGLOOe, except ProASIC3 nano and ProASIC3L

Description

The set_input_delay command sets input path delays on input ports relative to a clock edge. This usually represents a combinational path delay from the clock pin of a register external to the current design. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds input delay to path delay for paths starting at primary inputs.

A clock is a singleton that represents the name of a defined clock constraint. This can be:

- a single port name used as source for a clock constraint
- a single pin name used as source for a clock constraint; for instance reg1:CLK. This name can be hierarchical (for instance toplevel/block1/reg2:CLK)
- an object accessor that will refer to one clock: [get_clocks {clk}]

Examples

The following example sets an input delay of 1.2ns for port data1 relative to the rising edge of CLK1:

```
set_input_delay 1.2 -clock [get_clocks CLK1] [get_ports data1]
```

The following example sets a different maximum and minimum input delay for port IN1 relative to the falling edge of CLK2:

```
set_input_delay 1.0 -clock_fall -clock CLK2 –min {IN1}
set_input_delay 1.4 -clock_fall -clock CLK2 –max {IN1}
```

Microsemi Implementation Specifics

In SDC, the -clock is an optional argument that allows you to set input delay for combinational designs. Microsemi Implementation currently requires this argument.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Set Input Delay

set_load

SDC command; sets the load to a specified value on a specified port.

```
set_load capacitance port_list
```

Arguments

capacitance

Specifies the capacitance value that must be set on the specified ports.

port_list

Specifies a list of ports in the current design on which the capacitance is to be set.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

The load constraint enables the Designer software to account for external capacitance at a specified port. You cannot set load constraint on the nets. When you specify this constraint on the output ports, it impacts the delay calculation on the specified ports.

Examples

The following examples show how to set output capacitance on different output ports:

```
set_load  35   out_p
set_load  40   {O1 02}
set_load  25   [get_ports out]
```

Microsemi Implementation Specifics

- In SDC, you can use the set_load command to specify capacitance value on nets. Microsemi Implementation only supports output ports.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Set Load on Port

set_max_delay (SDC)

SDC command; specifies the maximum delay for the timing paths.

```
set_max_delay delay_value [-from from_list] [-to to_list]
```

Arguments

delay_value

Specifies a floating point number in nanoseconds that represents the required maximum delay value for specified paths.

- If the path starting point is on a sequential device, the tool includes clock skew in the computed delay.
- If the path starting point has an input delay specified, the tool adds that delay value to the path delay.
- If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay.
- If the ending point has an output delay specified, the tool adds that delay to the path delay.

`-from` *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-to` *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command specifies the required maximum delay for timing paths in the current design. The path length for any startpoint in from_list to any endpoint in to_list must be less than delay_value.

The tool automatically derives the individual maximum delay targets from clock waveforms and port input or output delays. For more information, refer to the create_clock, set_input_delay, and set_output_delay commands.

The maximum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multicycle path constraint.

Examples

The following example sets a maximum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay less than 5 ns:

```
set_max_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```

The following example sets a maximum delay by constraining all paths to output ports whose names start by “out” with a delay less than 3.8 ns:

```
set_max_delay 3.8 -to [get_ports out*]
```

Microsemi Implementation Specifics

The –through option in the set_max_delay SDC command is not supported.

See Also

Constraint Support by Family

Constraint Entry Table
SDC Syntax Conventions
Set Max Delay

set_min_delay

SDC command; specifies the minimum delay for the timing paths.

```
set_min_delay delay_value [-from from_list] [-to to_list]
```

Arguments

delay_value

Specifies a floating point number in nanoseconds that represents the required minimum delay value for specified paths.

- If the path starting point is on a sequential device, the tool includes clock skew in the computed delay.
- If the path starting point has an input delay specified, the tool adds that delay value to the path delay.
- If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay.
- If the ending point has an output delay specified, the tool adds that delay to the path delay.

`-from` *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-to` *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command specifies the required minimum delay for timing paths in the current design. The path length for any startpoint in from_list to any endpoint in to_list must be less than delay_value.
The tool automatically derives the individual minimum delay targets from clock waveforms and port input or output delays. For more information, refer to the create_clock, set_input_delay, and set_output_delay commands.
The minimum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multicycle path constraint.

Examples

The following example sets a minimum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay less than 5 ns:

```
set_min_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```

The following example sets a minimum delay by constraining all paths to output ports whose names start by “out” with a delay less than 3.8 ns:

```
set_min_delay 3.8 -to [get_ports out*]
```

Microsemi Implementation Specifics

The –through option in the set_min_delay SDC command is not supported.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

set_multicycle_path

SDC command; defines a path that takes multiple clock cycles.

```
set_multicycle_path ncycles [-setup] [-hold] [-from from_list] [–through through_list] [-to to_list]
```

Arguments

ncycles

Specifies an integer value that represents a number of cycles the data path must have for setup or hold check. The value is relative to the starting point or ending point clock, before data is required at the ending point.

`-setup`

Optional. Applies the cycle value for the setup check only. This option does not affect the hold check. The default hold check will be applied unless you have specified another set_multicycle_path command for the hold value.

`-hold`

Optional. Applies the cycle value for the hold check only. This option does not affect the setup check.

Note: If you do not specify "-setup" or "-hold", the cycle value is applied to the setup check and the default hold check is performed (*ncycles* -1).

`-from` *from_list*

Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.

`-through` *through_list*

Specifies a list of pins or ports through which the multiple cycle paths must pass.

`-to` *to_list*

Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

Setting multiple cycle paths constraint overrides the single cycle timing relationships between sequential elements by specifying the number of cycles that the data path must have for setup or hold checks. If you change the multiplier, it affects both the setup and hold checks.

False path information always takes precedence over multiple cycle path information. A specific maximum delay constraint overrides a general multiple cycle path constraint.

If you specify more than one object within one -through option, the path passes through any of the objects.

Examples

The following example sets all paths between reg1 and reg2 to 3 cycles for setup check. Hold check is measured at the previous edge of the clock at reg2.

```
set_multicycle_path 3 -from [get_pins {reg1}] –to [get_pins {reg2}]
```

The following example specifies that four cycles are needed for setup check on all paths starting at the registers in the clock domain ck1. Hold check is further specified with two cycles instead of the three cycles that would have been applied otherwise.

```
set_multicycle_path 4 -setup -from [get_clocks {ck1}]
set_multicycle_path 2 -hold -from [get_clocks {ck1}]
```

Microsemi Implementation Specifics

- SDC allows multiple priority management on the multiple cycle path constraint depending on the scope of the object accessors. In Microsemi design implementation, such priority management is not supported. All multiple cycle path constraints are handled with the same priority.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Set Multicycle Path

set_output_delay

SDC command; defines the output delay of an output relative to a clock.

```
set_output_delay delay_value -clock clock_ref [–max] [–min] [–clock_fall] output_list
```

Arguments

`delay_value`

Specifies the amount of time before a clock edge for which the signal is required. This represents a combinational path delay to a register outside the current design plus the library setup time (for maximum output delay) or hold time (for minimum output delay).

`-clock clock_ref`

Specifies the clock reference to which the specified output delay is related. This is a mandatory argument. If you do not specify -max or -min options, the tool assumes the maximum and minimum input delays to be equal.

`-max`

Specifies that delay_value refers to the longest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.

`-min`

Specifies that delay_value refers to the shortest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.

`-clock_fall`

Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.

`output_list`

Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

The set_output_delay command sets output path delays on output ports relative to a clock edge. Output ports have no output delay unless you specify it. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds output delay to path delay for paths ending at primary outputs.

Examples

The following example sets an output delay of 1.2ns for port OUT1 relative to the rising edge of CLK1:

```
set_output_delay 1.2 -clock [get_clocks CLK1] [get_ports OUT1]
```

The following example sets a different maximum and minimum output delay for port OUT1 relative to the falling edge of CLK2:

```
set_output_delay 1.0 -clock_fall -clock CLK2 –min {OUT1}
set_output_delay 1.4 -clock_fall -clock CLK2 –max {OUT1}
```

Microsemi Implementation Specifics

- In SDC, the -clock is an optional argument that allows you to set the output delay for combinational designs. Microsemi Implementation currently requires this option.

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

Set Output Delay

Design Object Access Commands

Design object access commands are SDC commands. Most SDC constraint commands require one of these commands as command arguments.

Microsemi software supports the following SDC access commands:

Design Object	Access Command
Cell	get_cells
Clock	get_clocks
Net	get_nets
Port	get_ports
Pin	get_pins
Input ports	all_inputs
Output ports	all_outputs
Registers	all_registers

See Also

About SDC Files

all_inputs

Design object access command; returns all the input or inout ports of the design.

```
all_inputs
```

Arguments

- None

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Exceptions

- None

Example

```
set_max_delay -from [all_inputs] -to [get_clocks ck1]
```

Microsemi Implementation Specifics

- None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

all_outputs

Design object access command; returns all the output or inout ports of the design.

```
all_outputs
```

Arguments

- None

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Exceptions

- None

Example

```
set_max_delay -from [all_inputs] -to [all_outputs]
```

Microsemi Implementation Specifics

None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

all_registers

Design object access command; returns either a collection of register cells or register pins, whichever you specify.

```
all_registers [-clock clock_name] [-cells] [-data_pins ]
 [-clock_pins] [-async_pins] [-output_pins]
```

Arguments

`-clock clock_name`

Creates a collection of register cells or register pins in the specified clock domain.

`-cells`

Creates a collection of register cells. This is the default. This option cannot be used in combination with any other option.

`-data_pins`

Creates a collection of register data pins.

`-clock_pins`

Creates a collection of register clock pins.

`-async_pins`

Creates a collection of register asynchronous pins.

`-output_pins`

Creates a collection of register output pins.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command creates either a collection of register cells (default) or register pins, whichever is specified. If you do not specify an option, this command creates a collection of register cells.

Exceptions

- None

Examples

```
set_max_delay 2 -from [all_registers] -to [get_ports {out}]
set_max_delay 3 –to [all_registers –async_pins]
set_false_path –from [all_registers –clock clk150]
set_multicycle_path –to [all_registers –clock c* -data_pins
 –clock_pins]
```

Microsemi Implementation Specifics

- None

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

get_cells

Design object access command; returns the cells (instances) specified by the pattern argument.

```
get_cells pattern
```

Arguments

pattern

Specifies the pattern to match the instances to return. For example, "get_cells U18*" returns all instances starting with the characters "U18", where “*” is a wildcard that represents any character string.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command returns a collection of instances matching the pattern you specify. You can only use this command as part of a –from, -to, or –through argument for the following constraint exceptions: set_max delay, set_multicycle_path, and set_false_path design constraints.

Exceptions

None

Examples

```
set_max_delay 2 -from [get_cells {reg*}] -to [get_ports {out}]
set_false_path –through [get_cells {Rblock/muxA}]
```

Microsemi Implementation Specifics

- None

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

get_clocks

Design object access command; returns the specified clock.

```
get_clocks pattern
```

Arguments

pattern

Specifies the pattern to match to the SmartTime on which a clock constraint has been set.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

- If this command is used as a –from argument in maximum delay (set_max_path_delay), false path (set_false_path), and multicycle constraints (set_multicycle_path), the clock pins of all the registers related to this clock are used as path start points.
- If this command is used as a –to argument in maximum delay (set_max_path_delay), false path (set_false_path), and multicycle constraints (set_multicycle_path), the synchronous pins of all the registers related to this clock are used as path endpoints.

Exceptions

- None

Example

```
set_max_delay -from [get_ports datal] -to \
[get_clocks ck1]
```

Microsemi Implementation Specifics

None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

get_pins

Design object access command; returns the specified pins.

```
get_pins pattern
```

Arguments

pattern
Specifies the pattern to match the pins.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Exceptions

None

Example

```
create_clock -period 10 [get_pins clock_gen/reg2:Q]
```

Microsemi Implementation Specifics

- None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

get_nets

Design object access command; returns the named nets specified by the pattern argument.

```
get_nets pattern
```

Arguments

pattern
Specifies the pattern to match the names of the nets to return. For example, "get_nets N_255*" returns all nets starting with the characters "N_255", where “*” is a wildcard that represents any character string.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Description

This command returns a collection of nets matching the pattern you specify. You can only use this command as source objects in create clock (create_clock) or create generated clock (create_generated_clock) constraints and as -through arguments in set false path (set_false_path), set minimum delay (set_min_delay), set maximum delay (set_max_delay), and set multicycle path (set_multicycle_path) constraints.

Exceptions

None

Examples

```
set_max_delay 2 -from [get_ports RDATA1] -through [get_nets {net_chkp1 net_chkqi}]
set_false_path –through [get_nets {Tblk/rm/n*}]
create_clcok -name mainCLK -per 2.5 [get_nets {cknet}]
```

Microsemi Implementation Specifics

None

See Also

Constraint Support by Family

Constraint Entry Table

SDC Syntax Conventions

get_ports

Design object access command; returns the specified ports.

```
get_ports pattern
```

Argument

pattern

Specifies the pattern to match the ports. This is equivalent to the macros $in()[<pattern>] when used as –from argument and $out()[<pattern>] when used as –to argument or $ports()[<pattern>] when used as a –through argument.

Supported Families

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion

Exceptions

None

Example

```
create_clock -period 10[get_ports CK1]
```

Microsemi Implementation Specifics

None

See Also

Constraint Support by Family
Constraint Entry Table
SDC Syntax Conventions

Glossary

arrival time

Actual time in nanoseconds at which the data arrives at a sink pin when considering the propagation delays across the path.

asynchronous

Two signals that are not related to each other. Signals not related to the clock are usually asynchronous.

capture edge

The clock edge that triggers the capture of data at the end point of a path.

clock

A periodic signal that captures data into sequential elements.

critical path

A path with the maximum delay between a starting point and an end point. In the presence of a clock constraint, the worst critical path between registers in this clock domain is the path with the worst slack.

dynamic timing analysis

The standard method for verifying design functionality and performance. Both pre-layout and post-layout timing analysis can be performed via the SDF interface.

exception

See timing exception.

explicit clock

Clock sources that can be traced back unambiguously from the clock pin of the registers they deserve, including the output of a DLL or PLL.

filter

A set of limitations applied to object names in timing analysis to generate target specific sets.

launch edge

The clock edge that triggers the release of data from a starting point to be captured by another clock edge at an end point.

minimum period

Timing characteristic of a path between two registers. It indicates how fast the clock will run when this path is the most critical one. The minimum period value takes into consideration both the skew and the setup on the receiving register.

parallel paths

Paths that run in parallel between a given source and sink pair.

path

A sequence of elements in the design that identifies a logical flow starting at a source pin and ending at a sink pin.

path details

An expansion of the path that shows all the nets and cells between the source pin and the sink pin.

path set

A collection of paths.

paths list

Same as path set.

post-layout

The state of the design after you run Layout. In post-layout, the placement and routing information are available for the whole design.

potential clock

Pins or ports connected to the clock pins of sequential elements that the Static Timing Analysis (STA) tool cannot determine whether they are is enabled sources or clock sources. This type of clock is generally associated with the use of gated clocks.

pre-layout

The state of the design before you run Layout. In pre-layout, the placement and routing information are not available.

recovery time

The amount of time before the active clock edge when the de-activation of asynchronous signals is not allowed.

removal time

The amount of time after the active clock edge when the de-activation of asynchronous signals is not allowed.

required time

The time at which the data must be at a sink pin to avoid being in violation.

requirement

See timing requirement.

scenario (timing constraints scenario)

Set of timing constraints defined by the user.

setup time

The time in nanoseconds relative to a clock edge during which the data at the input to a sequential element must remain stable.

sink pin

The pin located at the end of the timing path. This pin is usually the one where arrival time and required time are evaluated for path violation.

skew

The difference between the clock insertion delay to the clock pin of a sink register and the insertion delay to the clock pin of a source register.

slack

The difference between the arrival time and the required time at a specific pin, generally at the data pin of a sequential component.

slew rate

The time needed for a signal to transition from one logic level to another.

source pin

The pin located at the beginning of a timing path.

STA

See static timing analysis.

standard delay format (SDF)

Standard Delay Format, a standard file format used to store design data suited for back-annotation.

static timing analysis

An efficient technique to identify timing violations in a design and to ensure that all timing requirements are met. It is well suited for traditional synchronous designs. The main advantages are that it does not require input vectors, and it exclusively covers all possible paths in the design in a relatively short run-time.

synopsys design constraint (SDC)

A standard file format for timing constraints. Synopsys Design Constraints (SDC) is a Tcl-based format used by Synopsys tools to specify the design intent, including the timing and area constraints for a design. Microsemi SoC tools use a subset of the SDC format to capture supported timing constraints. You can import or export an SDC file from the Designer software. Any timing constraint that you can enter using Designer tools, can also be specified in an SDC file.

timing constraint

A requirement or limitation on the design to be satisfied during the design implementation.

timing exception

An exception to a general requirement usually applied on a subset of the objects on which the requirement is applied.

timing requirement

A constraint on the design usually determined by the specifications at the system level.

virtual clock

A virtual clock is a clock with no source associated to it. It is used to describe clocks outside the FPGA that have an impact on the timing analysis inside the FPGA. For example, if the I/Os are synchronous to an external clock.

WLM

Wire Load Model. A timing model used in pre-layout to estimate a net delay based on the fan-out.