SmartFusion2, IGLOO2, and RTG4 SmartTime, I/O Editor, and ChipPlanner for Libero SoC v11.8

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.**

5-02-00427-5/01.17

Table of Contents

Timing Constraints

Timing Constraints - SmartFusion2, IGLOO2, RTG4

With SmartTime, you can 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.

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 timing constraints.

Use the SmartTime Constraint Editor to enter and edit Timing Constraints.

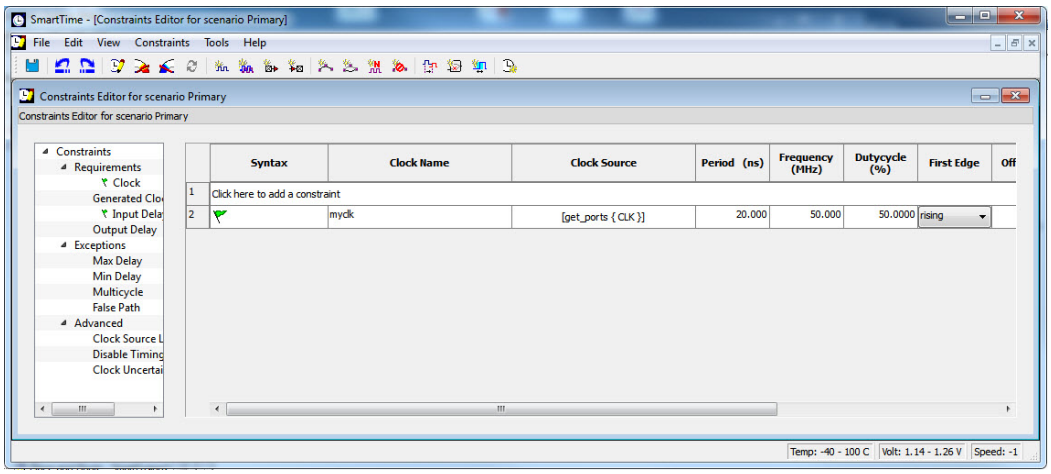


Figure 1 · SmartTime Constraints Editor

Timing Constraints 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

SmartTime Constraint Scenario - SmartFusion2, IGLOO2, RTG4

Constraints Editor Components

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

Constraints Editor Components

SmartTime Constraints 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, which enables you to analyze the impact of constraint changes.

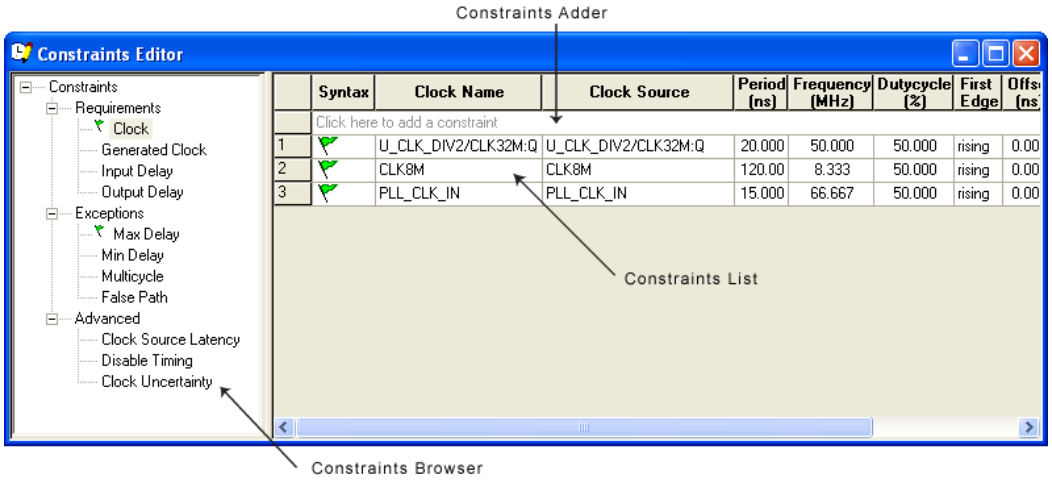


Figure 2 · SmartTime Constraints Editor View

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 and exceptions, while the Constraint List provides details about each constraint and enables you 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.
- Click or double-click a constraint in the Constraint List to edit and check the syntax of the selected constraint.
- Select a row and right-click to edit, delete, or copy the selected constraint to a spreadsheet.

- Select the entire spreadsheet and copy it to another spreadsheet.

See Also

Editable Constraints Grid

SmartTime Constraint Scenario - SmartFusion2

Editable Constraints Grid

The Constraints Editor enables you to add, edit and delete.

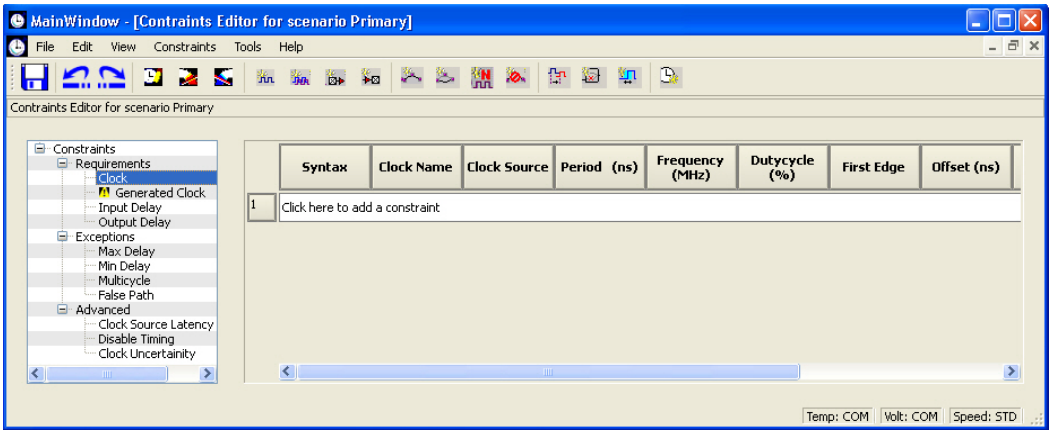


Figure 3 · Constraints Editor

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 the green check mark to apply your changes. To cancel the changes press the red cancel mark.
3. The new constraint is added to the Constraint List. The green syntax flag indicates that the constraint was successfully checked.

To edit a constraint:

1. Select a constraint type from the constraint browser.
2. Select the constraint, edit the values and click the green check mark to apply your changes. To cancel the changes press the red cancel mark. The green syntax flag indicates that the constraint was successfully checked.

To delete a constraint:

1. Select a constraint type from the constraint browser.
2. Right-click the constraint you want to delete and choose **Delete Constraint**.

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

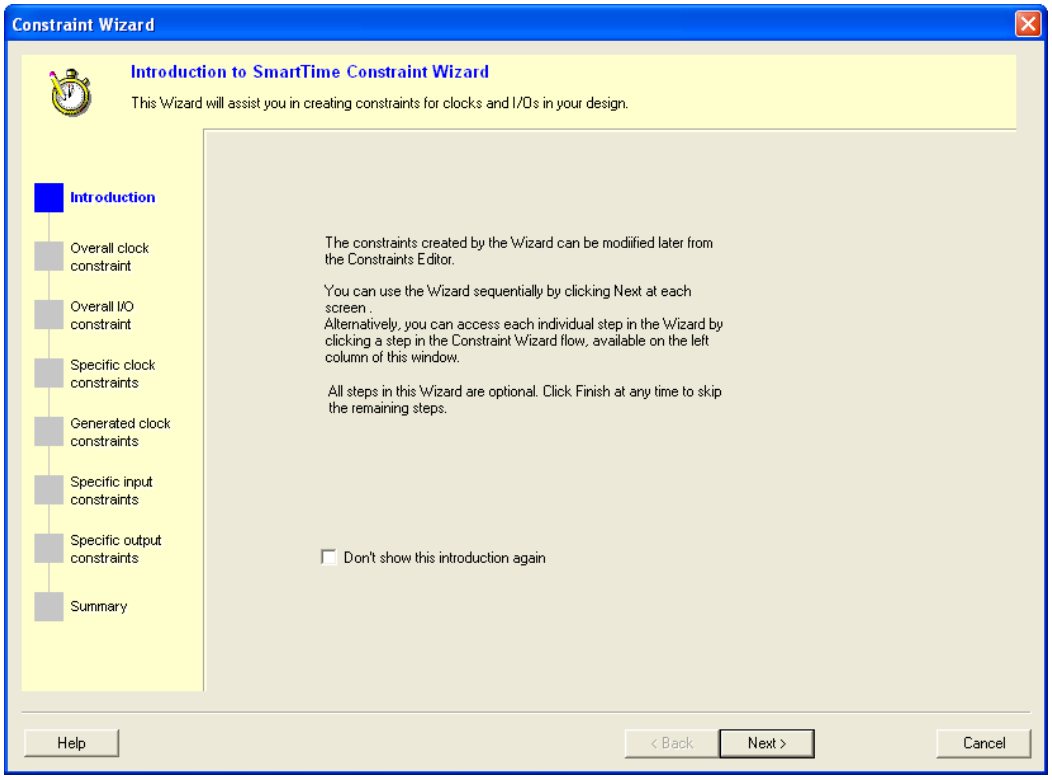


Figure 4 · Constraint Wizard

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

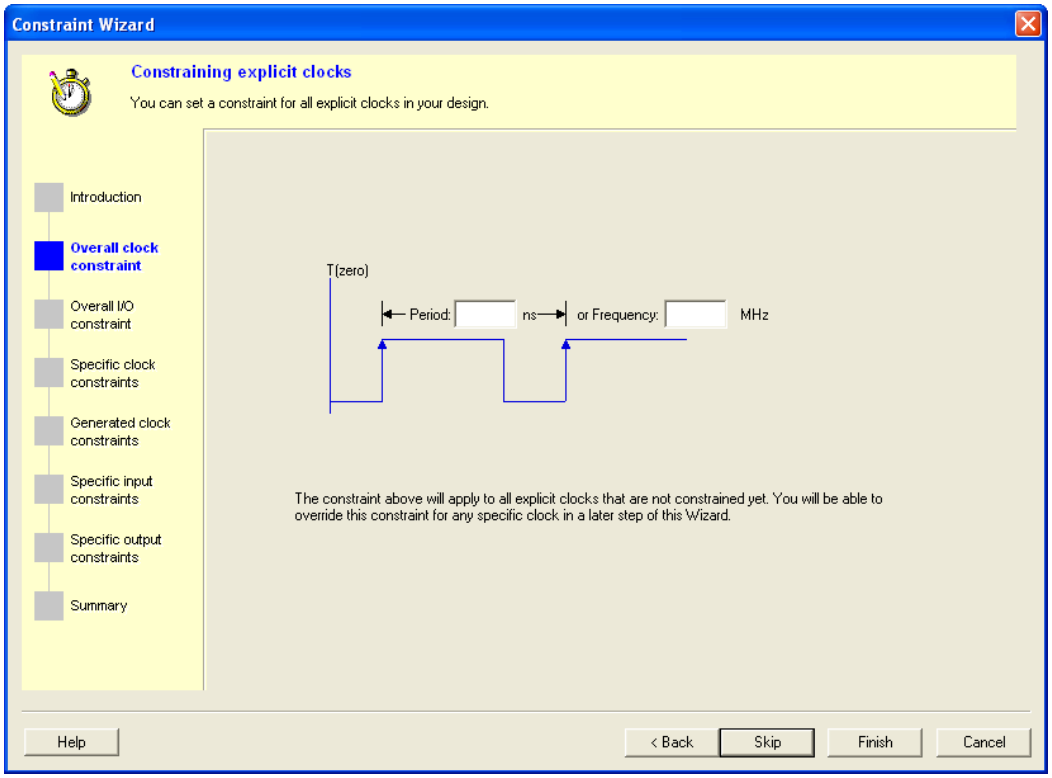


Figure 5 · Constraint Wizard – Overall Clock Requirements

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

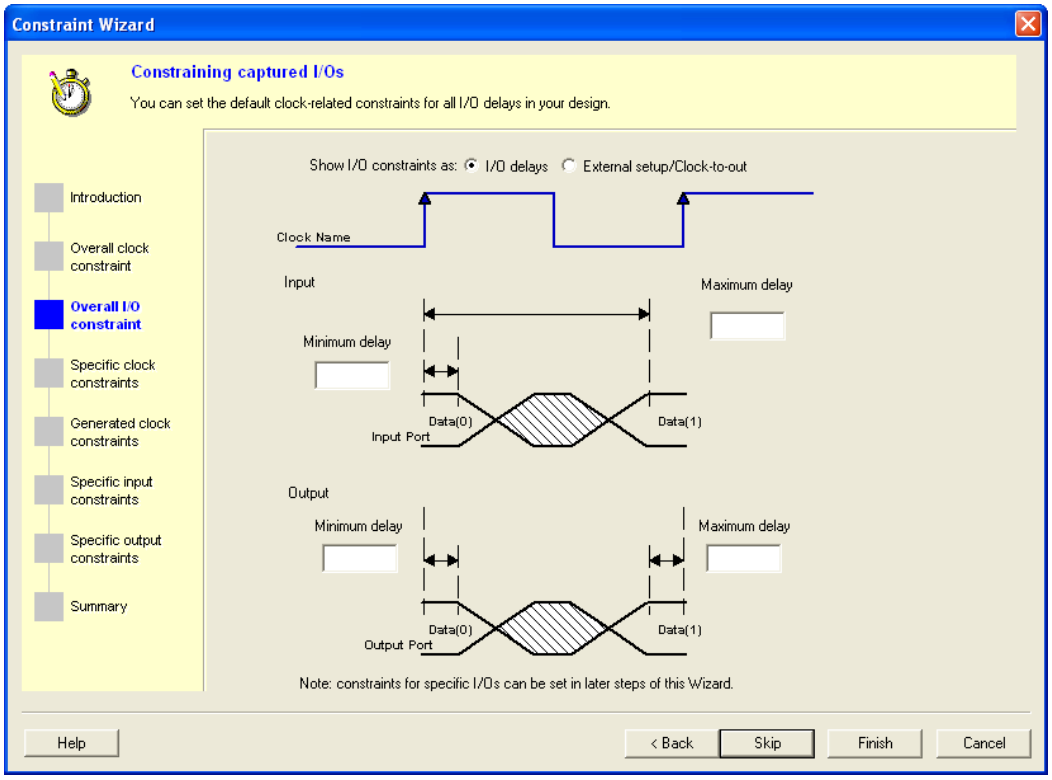


Figure 6 · Constraint Wizard – Overall I/O Constraint

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

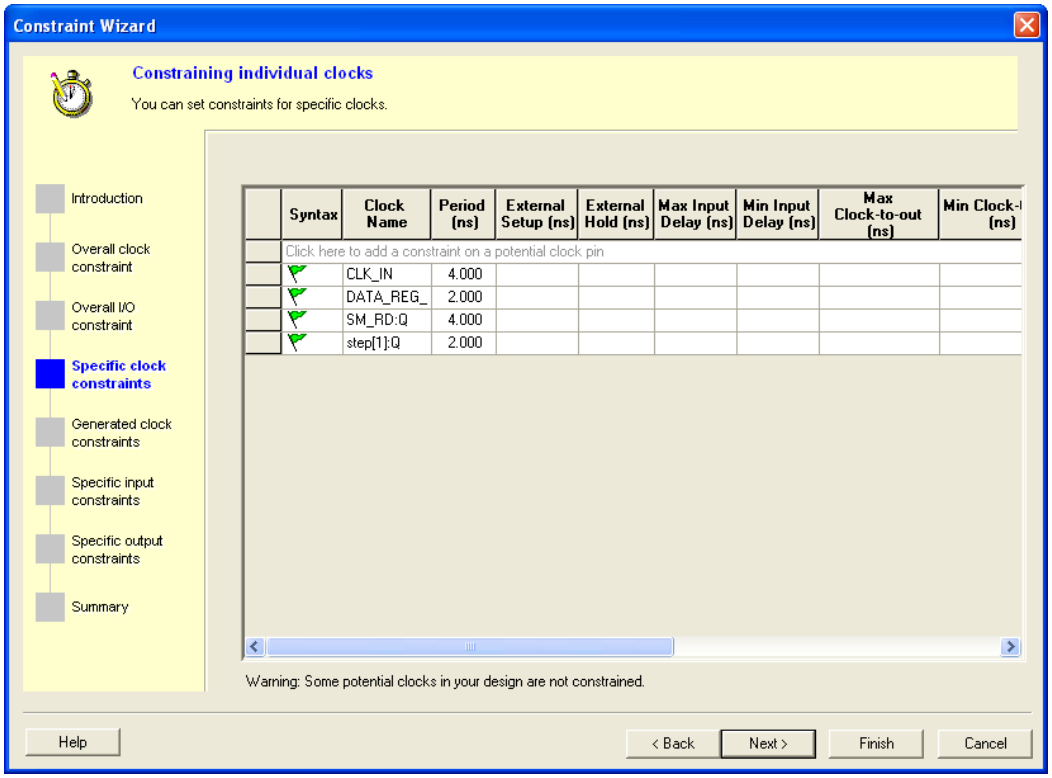


Figure 7 · Constraint Wizard – Specific Clock Constraints

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

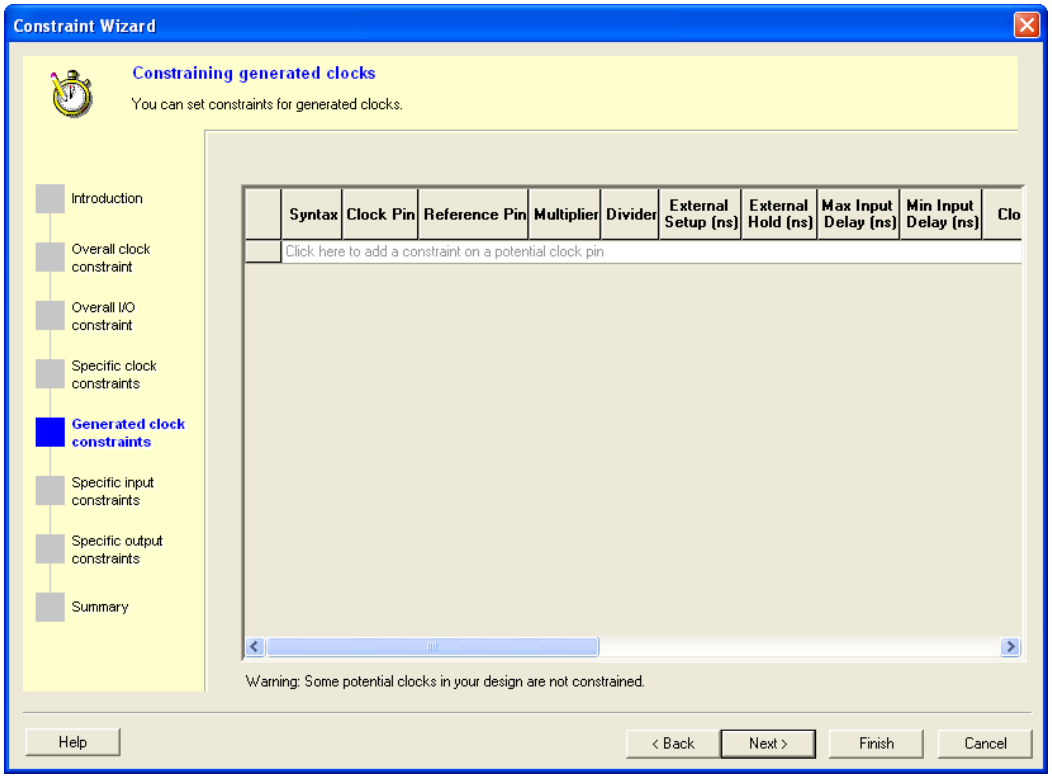


Figure 8 · Constraint Wizard – Generated Clock Constraints

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

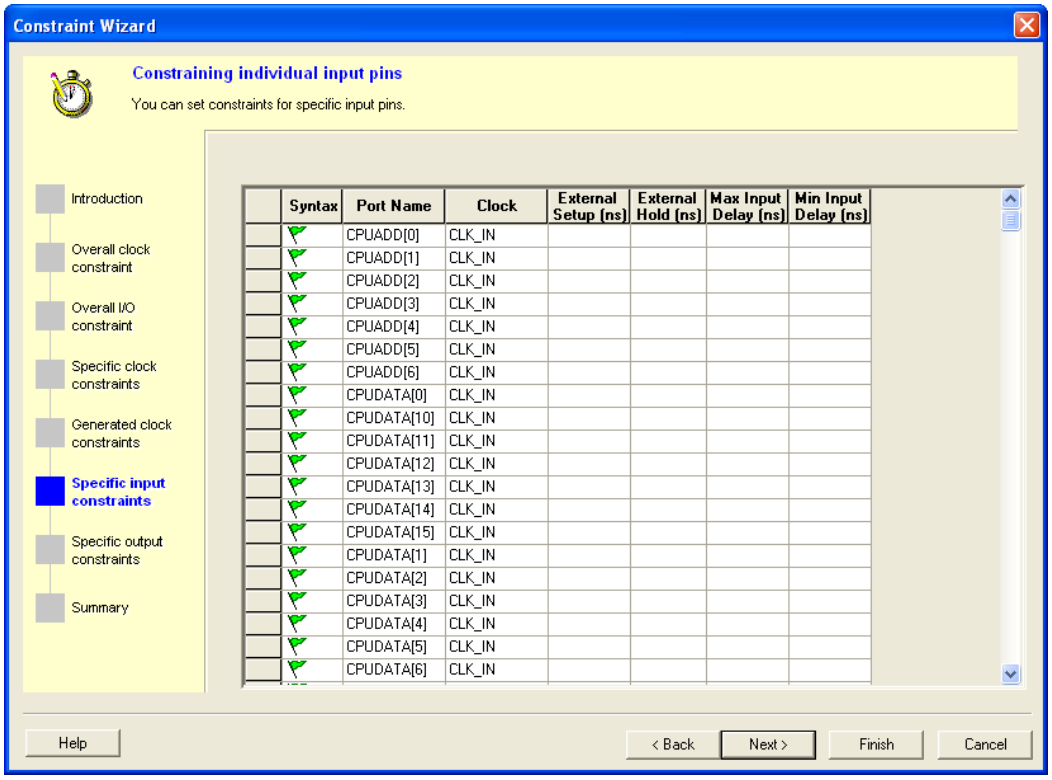


Figure 9 · Constraint Wizard – Specific Input Constraints

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

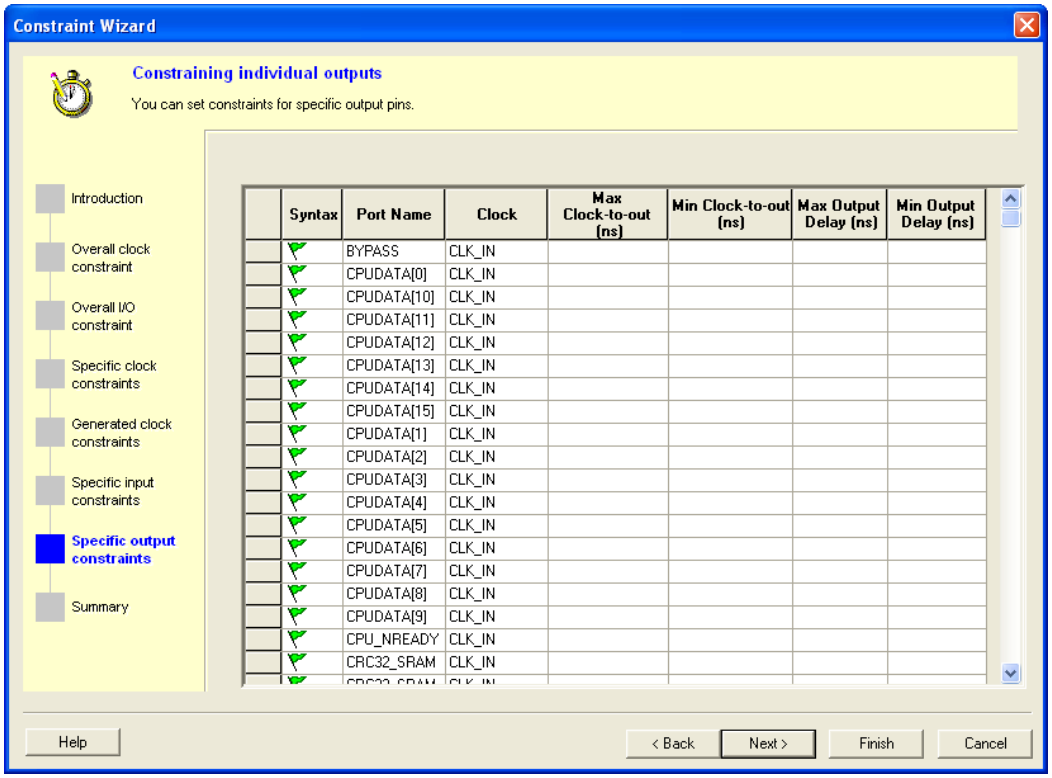


Figure 10 · Constraint Wizard – Specific Output Constraints

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

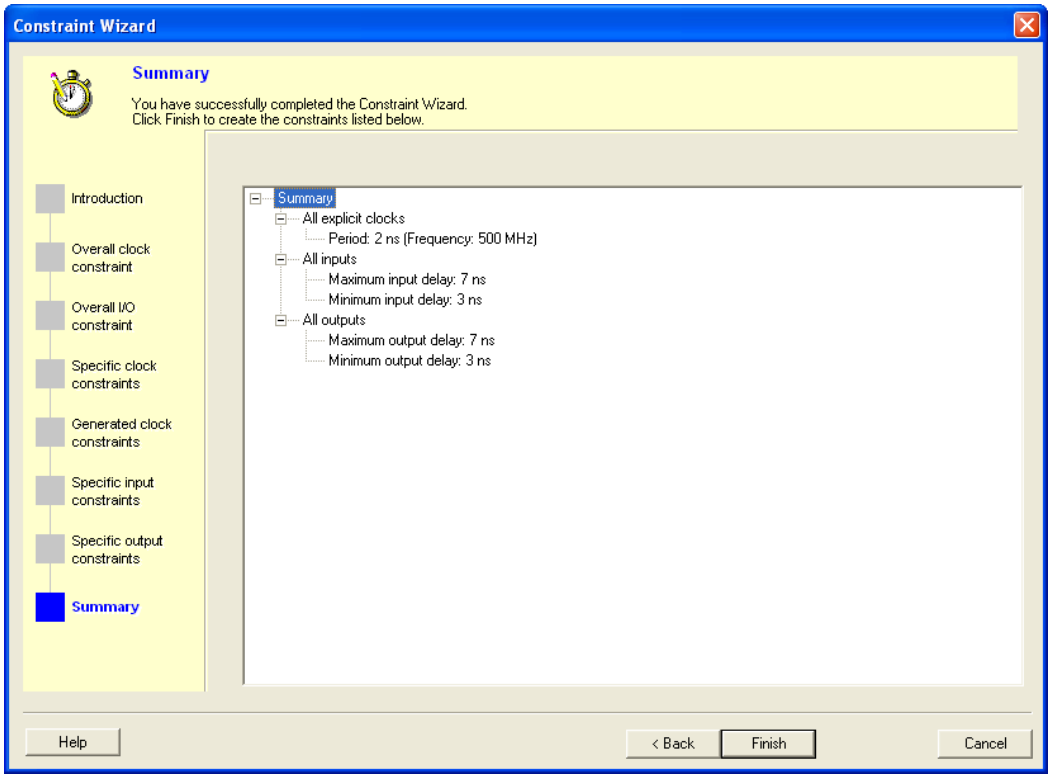


Figure 11 · Constraint Wizard – Summary

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).

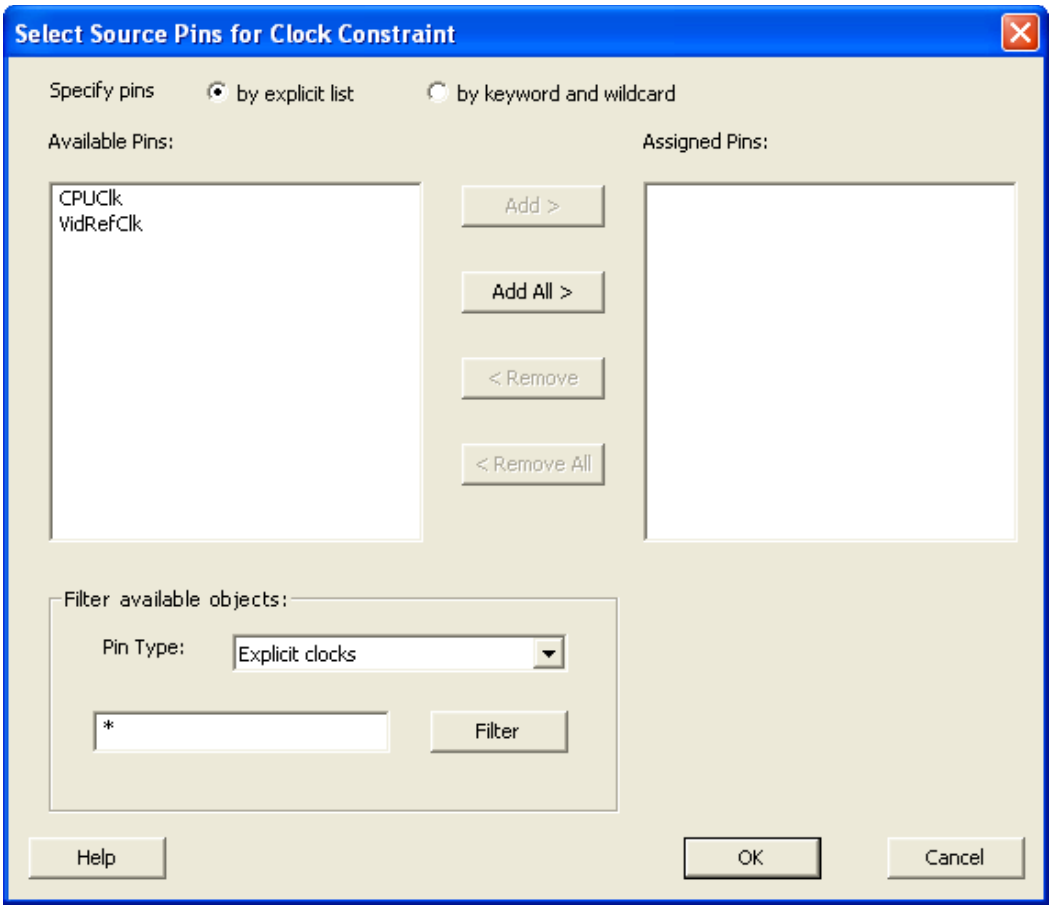


Figure 12 · Select Source Pins for Clock Constraint Dialog Box

See Also

Select Source Pins for Clock Constraint Dialog Box

Understanding Explicit Clocks

Understanding Potential Clocks

Understanding Clock Networks

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.

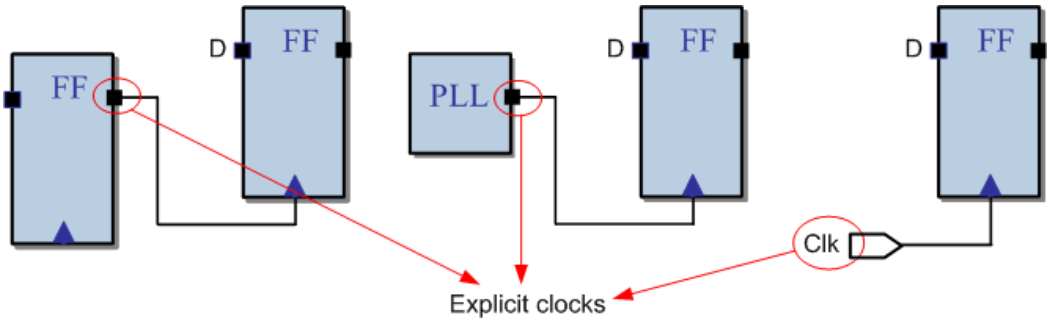


Figure 13 · Explicit Clocks

See Also

Select Source Pins for Clock Constraint Dialog Box

Using Clock Types

Potential Clocks

Clock Networks

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.

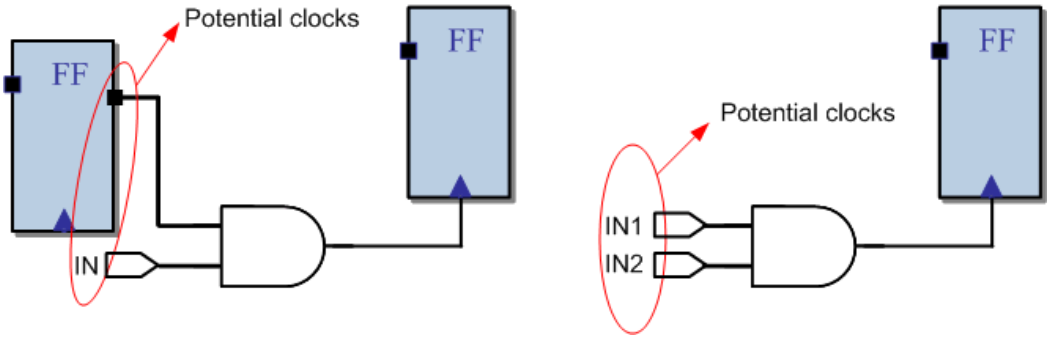


Figure 14 · Potential Clocks

See Also

Select Source Pins for Clock Constraint Dialog Box

Using Clock Types

Clock Networks

Explicit Clocks

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.

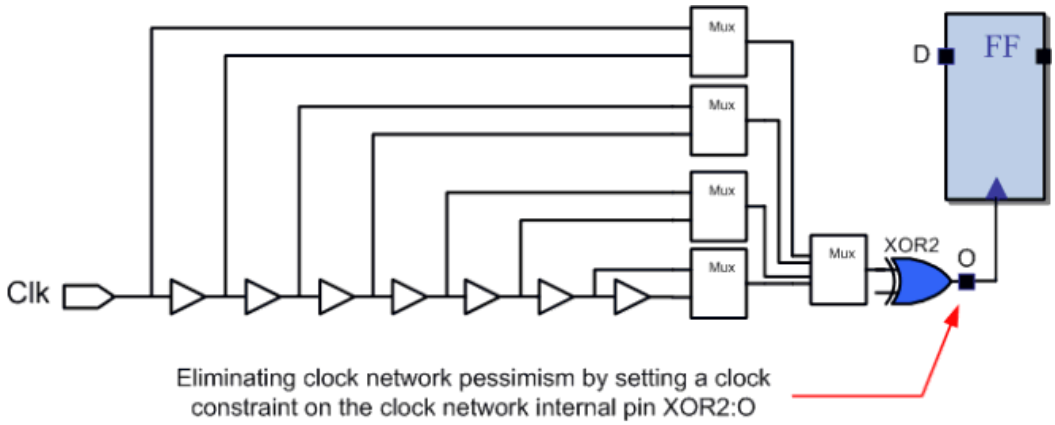


Figure 15 · Source Being Set Within a Clock Network

See Also

Select Source Pins for Clock Constraint Dialog Box

Using Clock Types

Potential Clocks

Explicit Clocks

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).

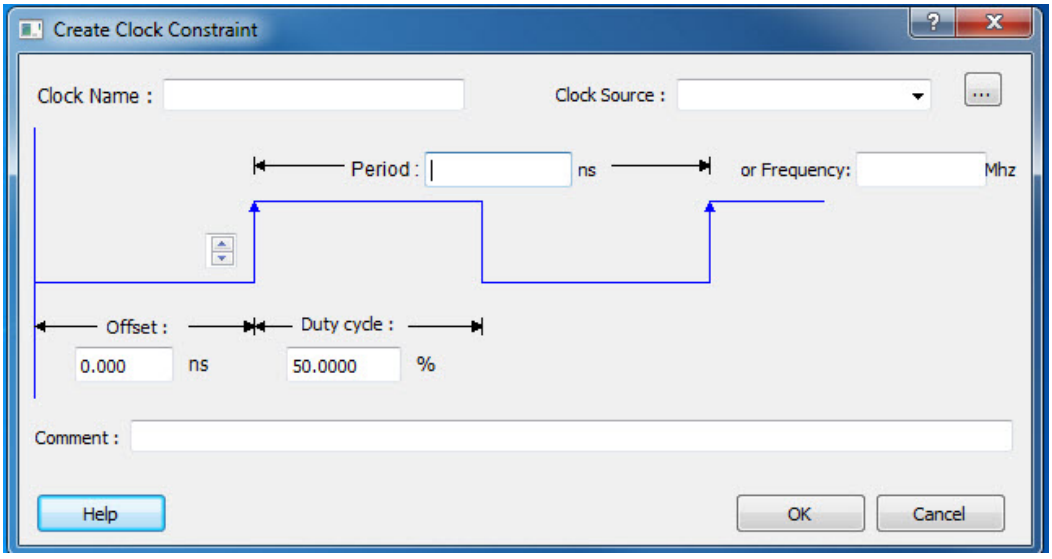


Figure 16 · Create Clock Constraint Dialog Box

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.

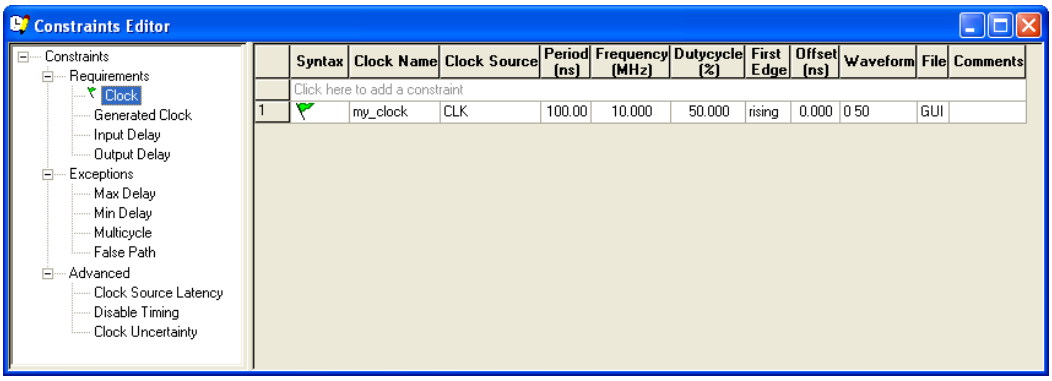


Figure 17 · SmartTime Timing Constraint View

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).

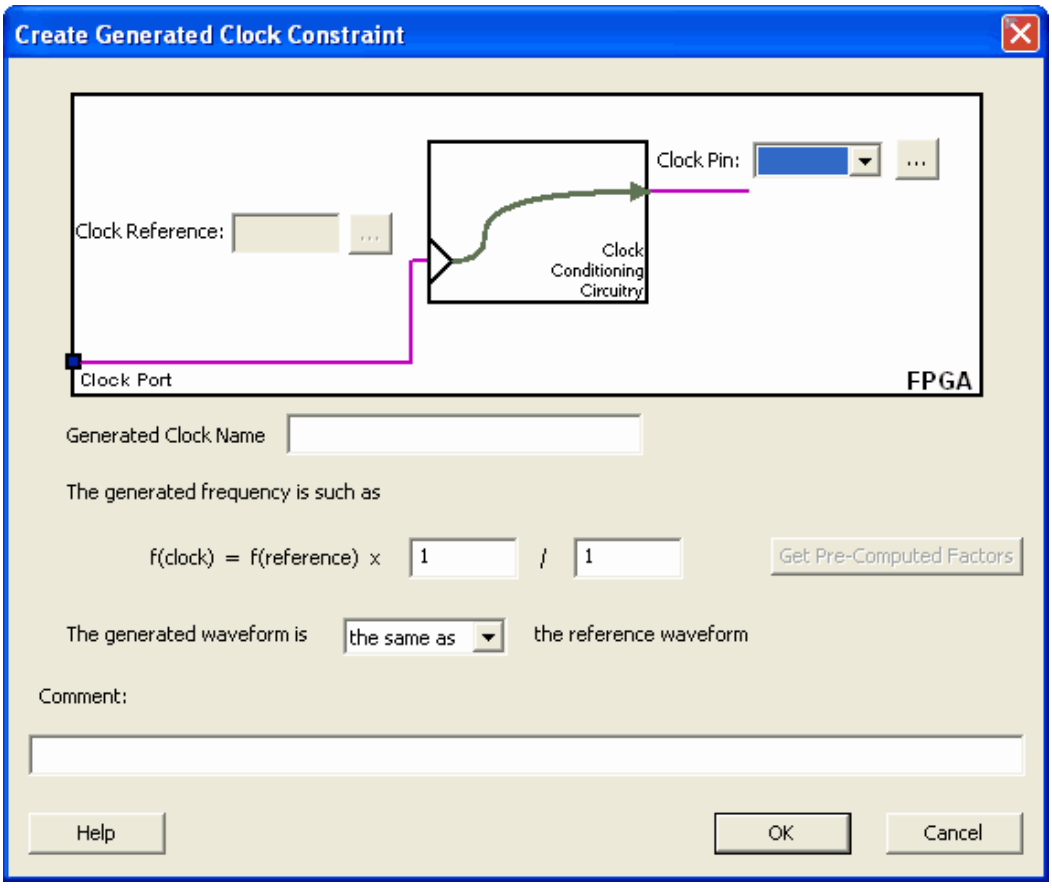


Figure 18 · Create Generated Clock Constraint

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).

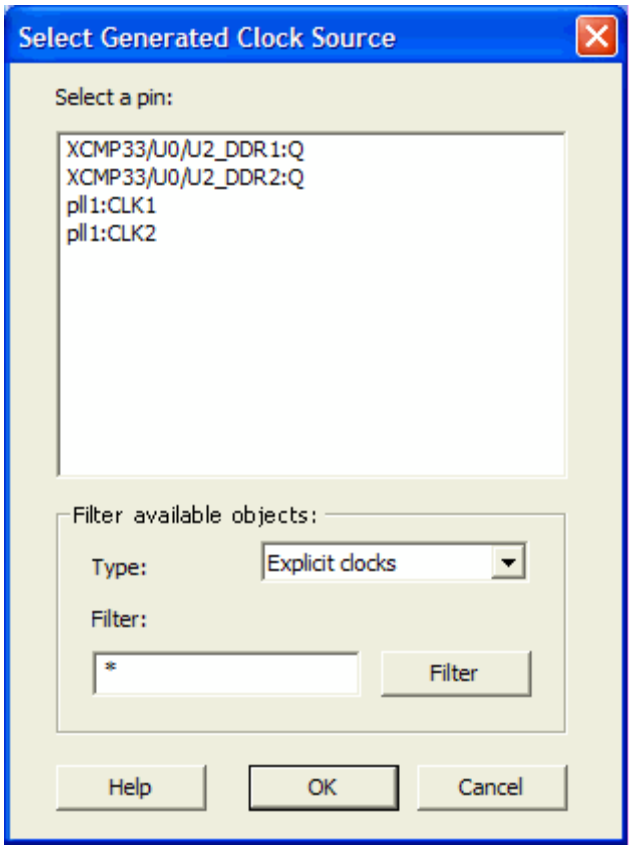


Figure 19 · Select Generated Clock Source Dialog Box

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).

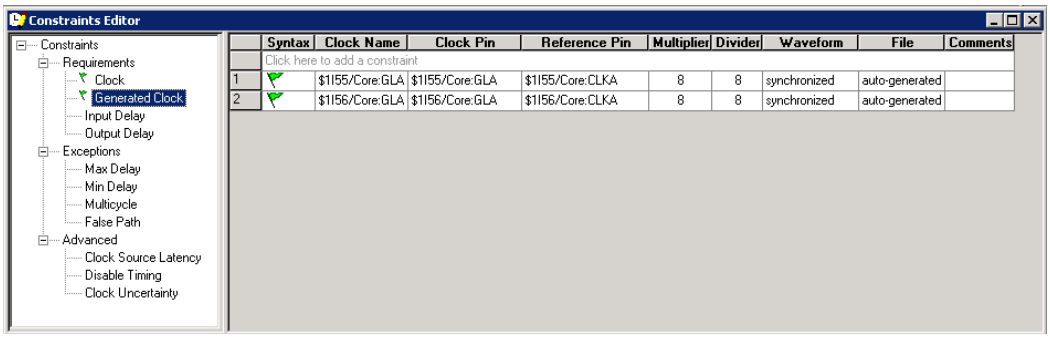


Figure 20 · Constraints Editor

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).

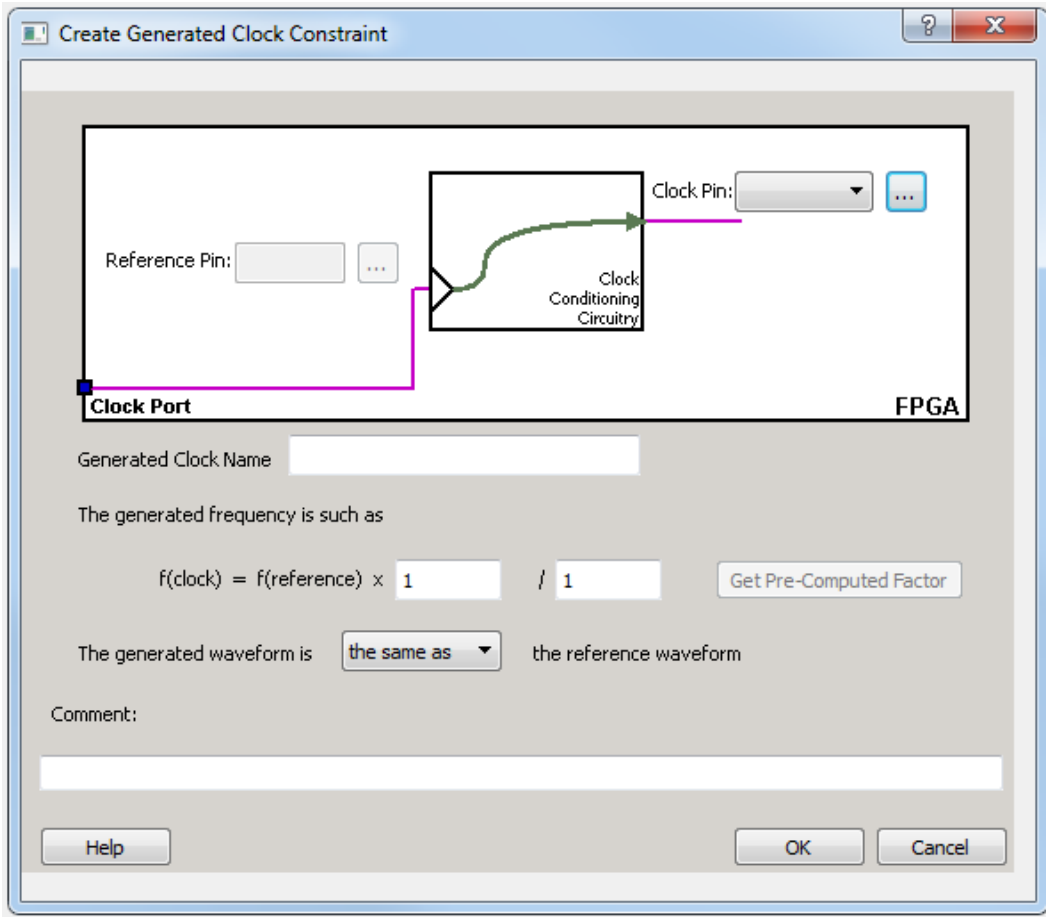


Figure 21 · Create Generated Clock Constraint

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 **Constraints** menu, choose **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).

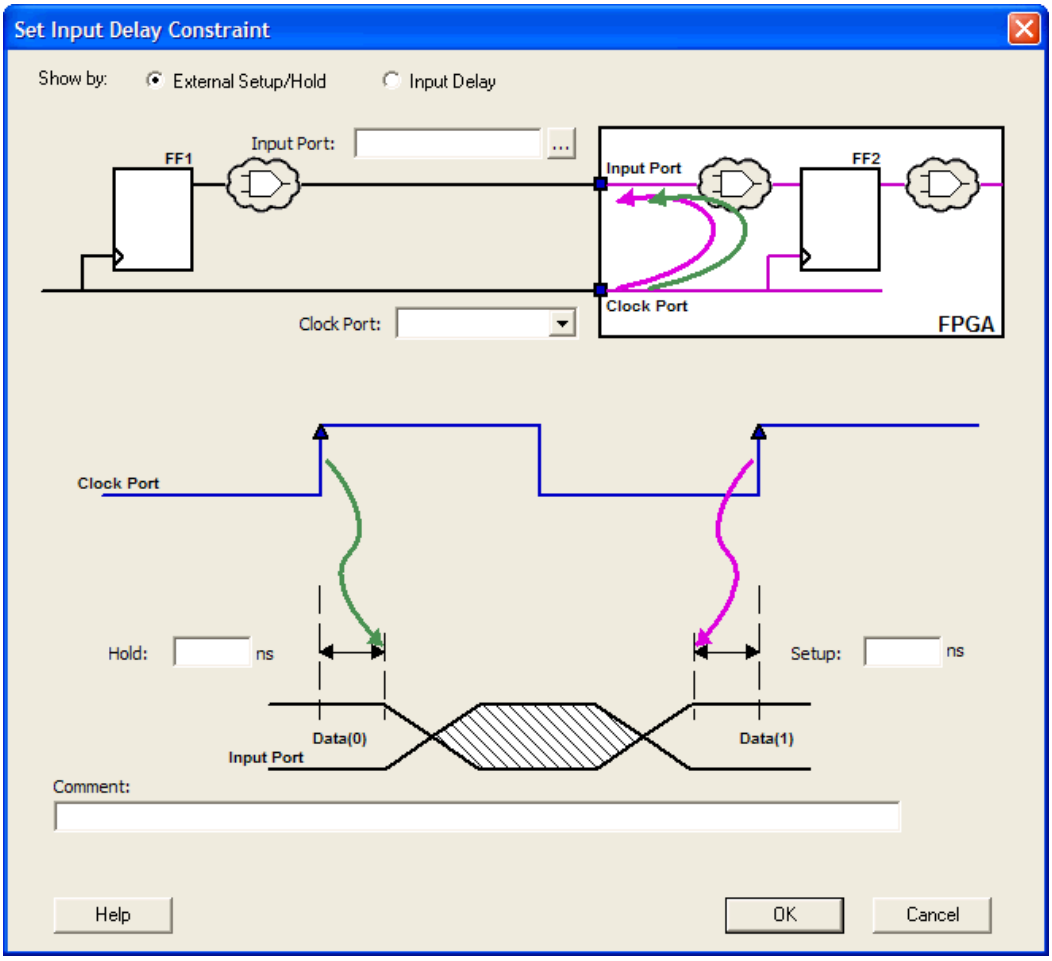


Figure 22 · Set Input Delay Dialog Box

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.

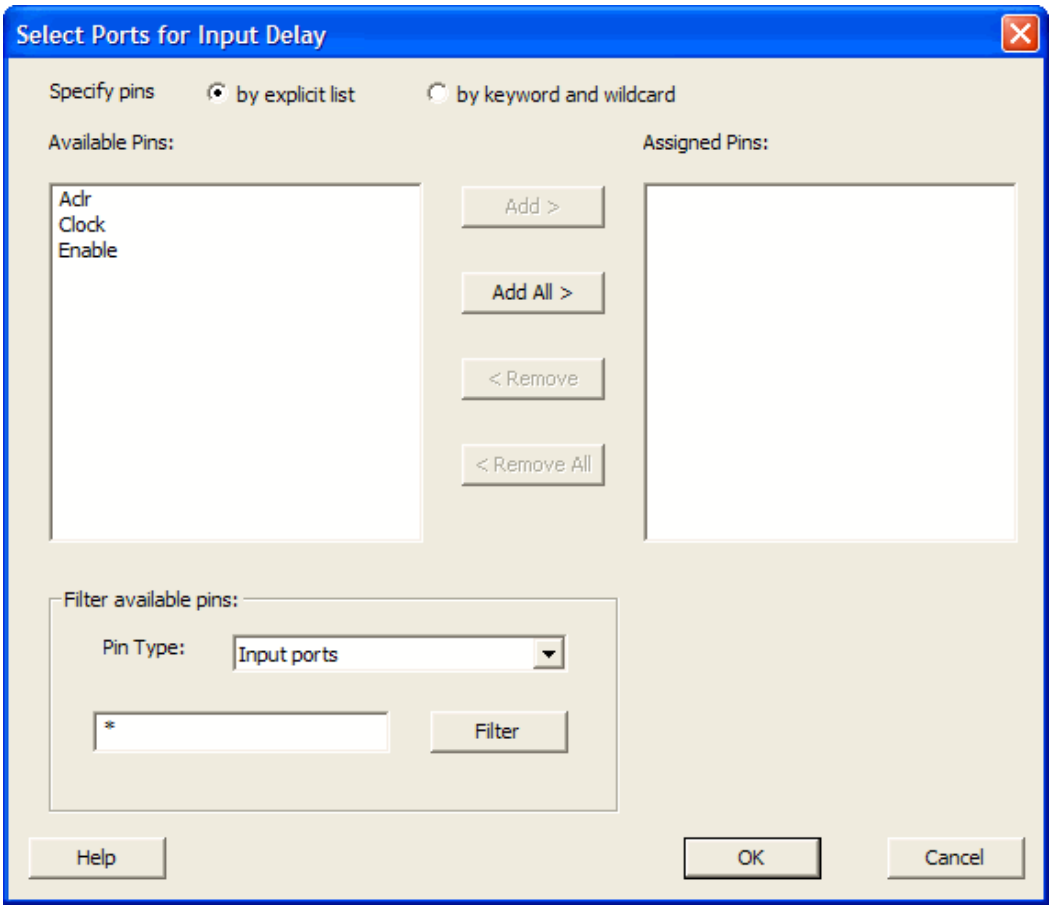


Figure 23 · Select Ports for Input Delay Dialog Box

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.

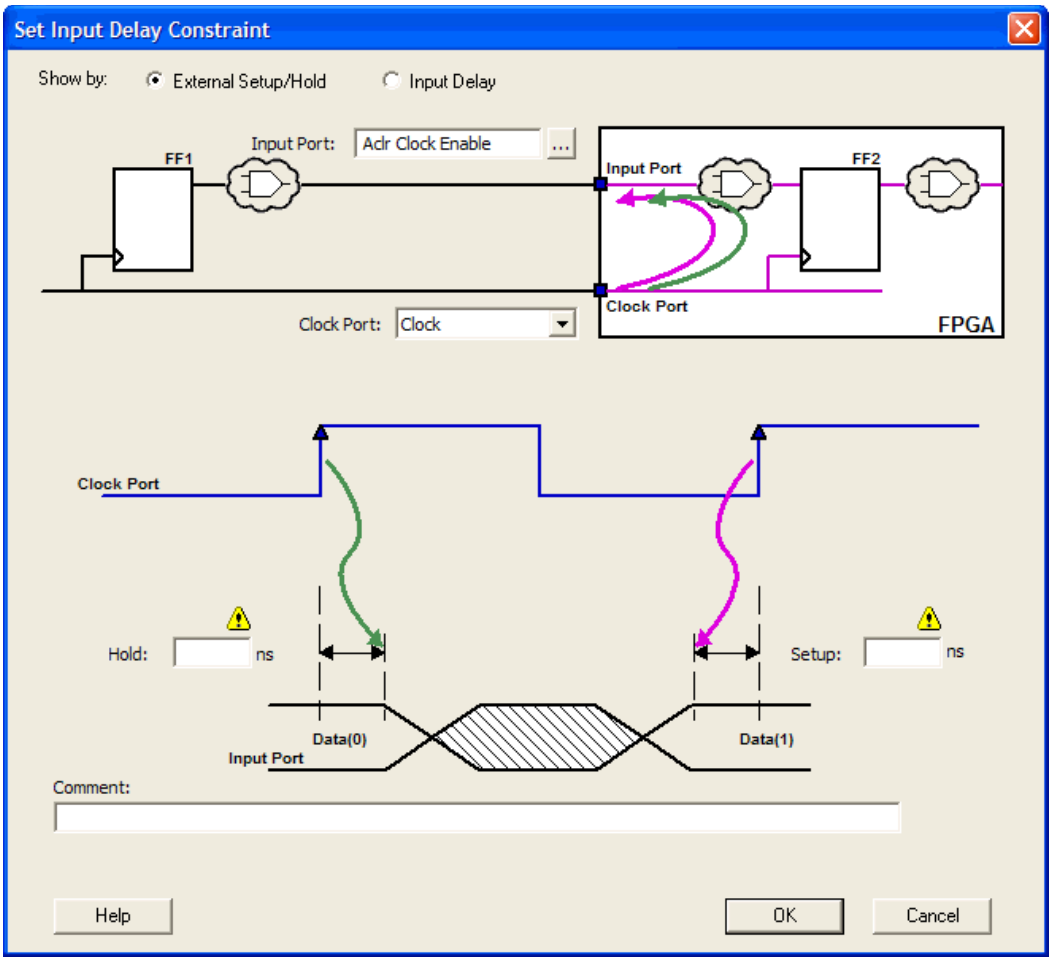


Figure 24 · Updated Set Input Delay Constraint Dialog Box

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 Output Delay Constraints** using one of the following methods:
   - Click the  icon.
   - Right-click **Output Delay** in the Constraint Browser and choose **Add Output Delay Constraints**.

- Double-click **Output Delay** in the Constraints Browser.

The Add Output Delay Constraint dialog box appears.

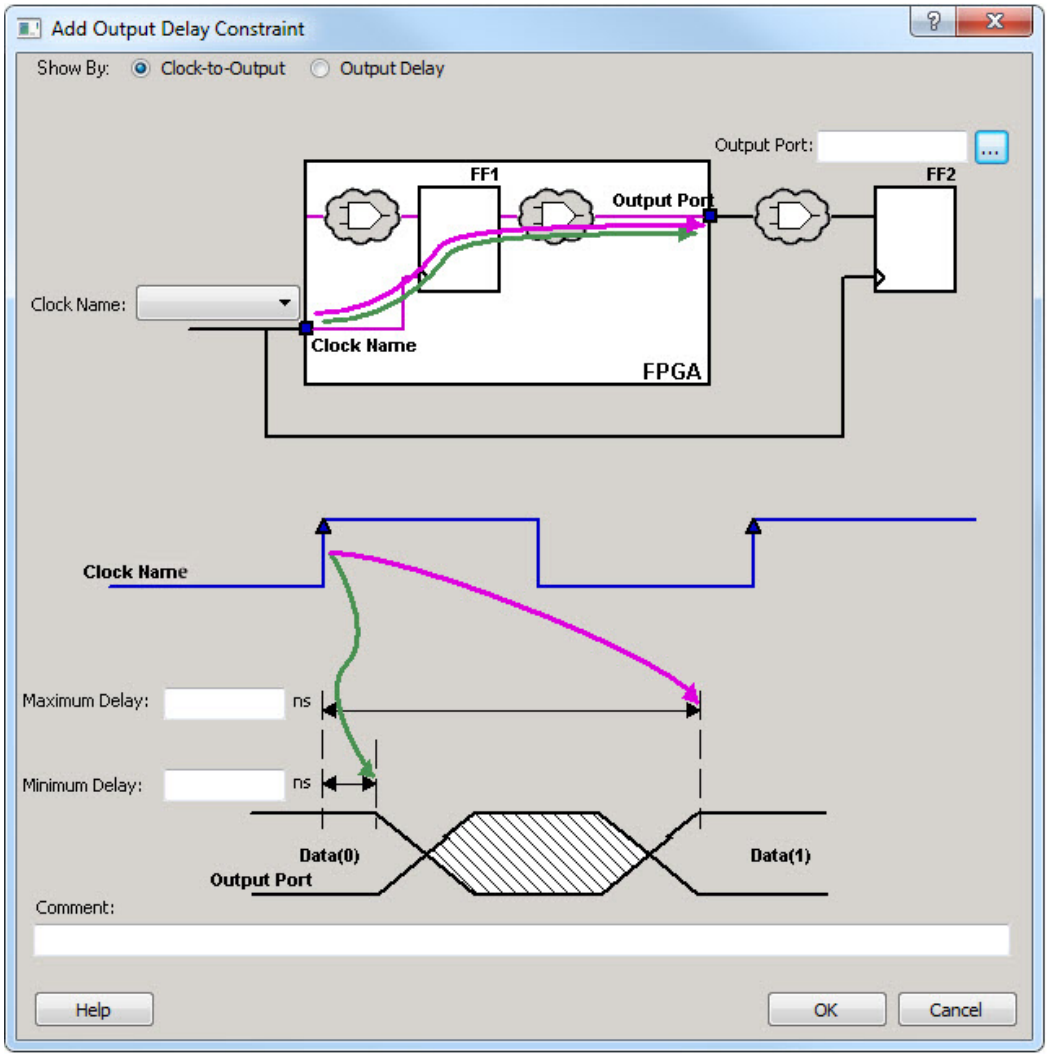


Figure 25 · Add Output Delay Constraint Dialog Box

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.

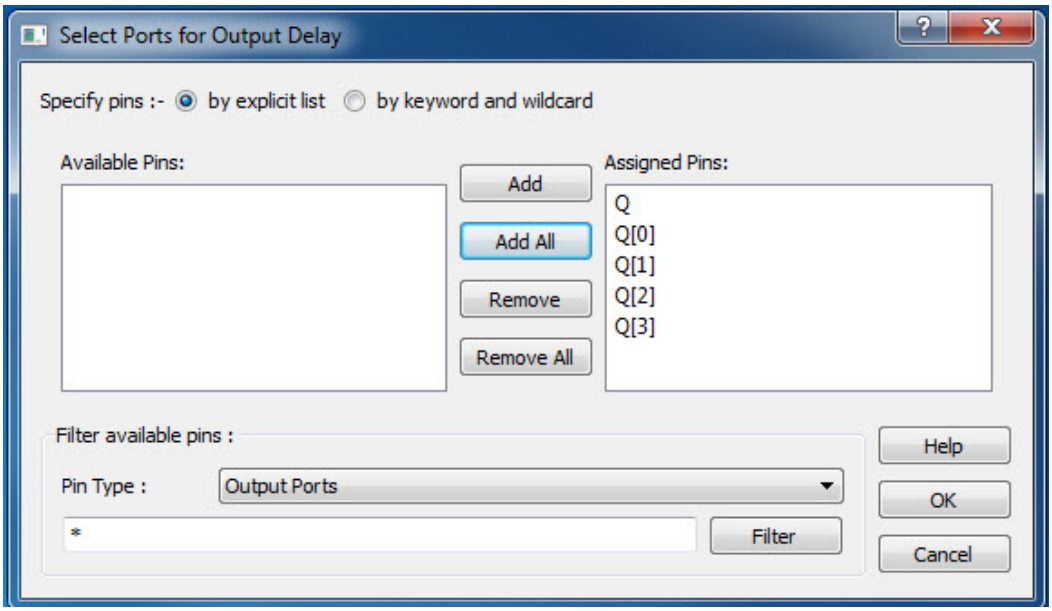


Figure 26 · Select Ports for Output Delay Dialog Box

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

I/O Editor - SmartFusion2 and IGLOO2, RTG4

For Classic Constraint Flow: Double-click **I/O Constraints** (in the Libero SoC Design Flow window, **Implement Design > Edit Constraints > I/O Constraints**) to start the I/O Editor.

For Enhanced Constraint Flow: Double-click **Manage Constraints** in the Libero SoC Design Flow window. Select the **I/O Attributes** tab and select **Edit > Edit with I/O Editor.**

The I/O Editor opens and displays all assigned and unassigned I/O macros and their attributes in a spreadsheet format.

Use I/O Attribute Editor to:

- View, sort, select, and edit common and device-specific I/O attributes
- Lock and unlock assigned attributes

You can view the I/O attributes by port or by package pin. Click the **Ports** tab to view I/O attributes by port name. Click the **Package Pins** tab to view I/O attributes by pin number.

Each row corresponds to an I/O macro (port) or a pin in the design, depending on the view displayed. The column headings specify the names of the I/O attributes in your design. The first four column headings are standard for all families so they will not change. However, the other column headings will change depending on the family you are designing for. For some I/O attributes, you will choose from a drop-down menu; for others, you might enter a value.

To edit I/O attributes:

1. Select an I/O standard for each I/O macro in your device.
2. Select I/O attributes that are available for your selected I/O standard.

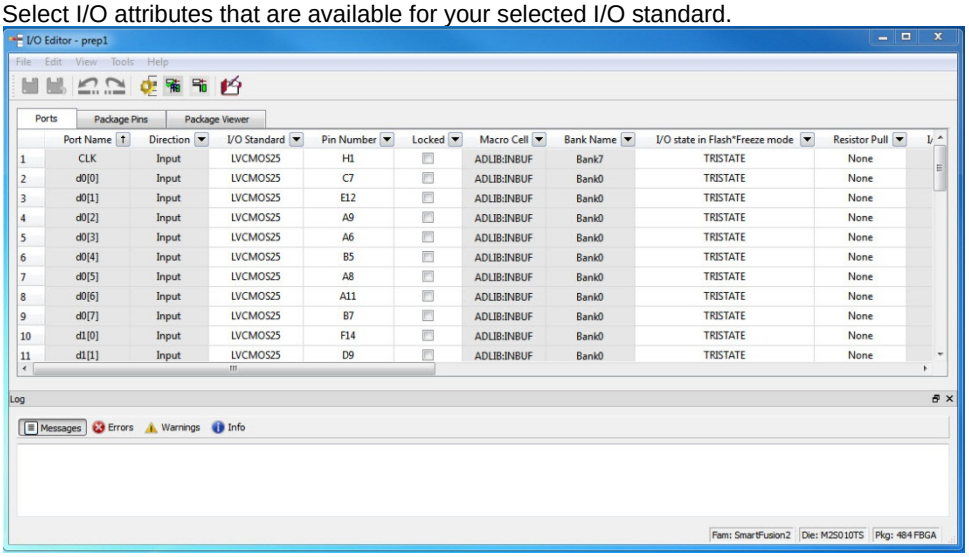


Figure 27 · I/O Editor

Assigning Pins in Package View

I/O Editor includes a Package Pins view in addition to its Ports view. Click the **Package Pins** tab to display your I/O attributes by package pin number. This view makes it easier to assign address/data ports to adjacent pins. Additionally, it enables you to assign VREF pins (which you cannot do in Ports view) and to sort on banks.

Package Pins View

The Package Pins View displays all columns shown in the Ports view plus the following additional columns:

- Function
- Dedicated
- VREF
- User Reserved

Function is the functionality of the I/O (for example, GND or ground). See the datasheet for your device for details about each function.

Dedicated determines whether the pin is reserved for some special functionality, such as UJTAG / Analog Block / XTL pads inputs.

VREF (Voltage referenced), if checked, assigns the selected pin as a VREF. This column only appears for devices that support VREF (IGLOOe, Fusion, SmartFusion, ProASIC3L, ProASIC3E, SmartFusion2, IGLOO2, RTG4). A device supports VREF if one or more of its I/O banks support VREF.IGLOO (excluding IGLOOe) and ProASIC3 (excluding ProASIC3L A3PE3000L and ProASIC3E) devices are not supported.

User Reserved, if checked, reserves the pin for use in another design. When a pin is reserved, you cannot assign it to a port. To unreserve the pin, deselect the **User Reserved** check box.

Edit Multiple Rows in I/O Editor

To edit multiple rows:

1. Select the rows to edit. To select consecutive rows, click the first row, press and hold down the **SHIFT** key, and then click the last row. To select rows that are not consecutive, press and hold down the **CTRL** key, and then click each row to select. Continue to hold down the **SHIFT** or **CTRL** key.
2. While still holding down the **SHIFT** or **CTRL** key, click in the cell containing the value you want to change. Release the **SHIFT** or **CTRL** key, and then release the mouse button.

The change occurs in all selected rows.

Note: You can also select an entire column, which enables you to edit all rows in that column.

Sorting I/O Attributes

You can sort rows by column in either ascending or descending order.

To sort I/O macros by attributes:

- Double-click a column heading to sort the table rows in ascending order.
- Double-click the column again to sort the table rows in descending order.

When sorted, an arrowhead appears in the column header to indicate the sort order.

Specifying an I/O Standard

Use the I/O Standard column to select an I/O specification for each pin.

If required to match the I/O standard, other I/O attributes, such as I/O threshold, slew, and loading, are automatically set to their default settings; you cannot edit these defaults.

You can change the I/O standards only for a generic I/O buffer to any of the legal I/O standards.

To specify an I/O standard:

1. Click the **I/O Standard** cell in the desired macro row.
2. Type or select a supported I/O standard from the drop-down list.

For devices that support I/O banks, the list is restricted to legal choices only. When an I/O is assigned, the I/O standards available for that I/O are limited to what the I/O bank location can support.

Note: Changing an I/O standard may also unassign existing I/Os. In addition, when a macro is assigned an I/O standard, the I/O bank is automatically assigned the voltages VCCI and VREF, if necessary. Unassigning this macro will undo these assignments as well.

I/O Bank Settings

To change I/O Bank Settings in SmartFusion2, IGLOO2, RTG4 devices:

1. Click the **I/O Bank Settings** button .
2. Choose the Bank you wish to modify. When you select your I/O Bank, the list of Available and Disabled Technologies appears in the I/O Bank Settings dialog box (as shown in the figure below).

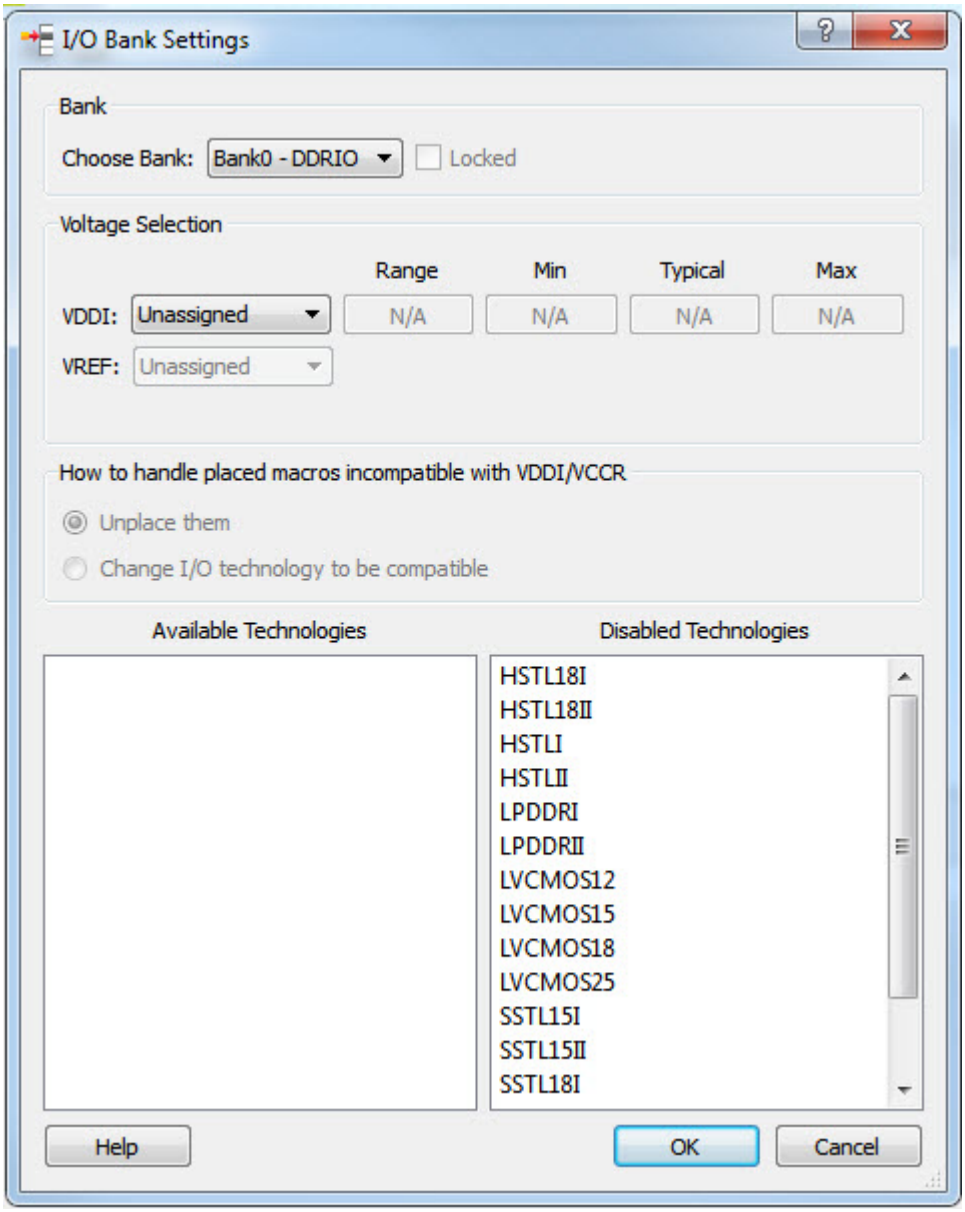


Figure 28 · I/O Bank Settings

3. Leave the **Use default pins for VREFs** option selected to set default VREF pins and unset non-default VREF pins. If you unselect this option when setting a new VREF technology, no VREF pins are set. If you unselect this option when default VREF pins are already set, it unsets them.

If the **Use default pins for VREFs** option is selected when you click **OK** or **Apply**, the software: 1) determines if setting default VREF pins causes any I/O macros to become unassigned, and if so, displays a warning message enabling you to cancel this operation, 2) determines if unsetting non-default VREF pins causes any I/O macros to become unassigned, and if so, displays a warning message enabling you to cancel this operation, and 3) sets default VREF pins and unsets non-default VREF pins.

4. Click to **Unplace pins on VCCI/VCCR change**, if necessary.
5. Click **OK**.

Auto-Assign I/O Banks

The I/O Bank Assigner tool runs automatically when you run Layout. You can also use this tool from within the I/O Editor. The I/O Bank Assigner tool automatically assigns technologies and VREF pins (if required) to every I/O bank that does not currently have any technologies assigned to it. This tool is available when at least one I/O bank is unassigned.

Each time you run the I/O Bank Assigner, it unassigns all technologies from all I/O banks and then re-assigns them when it finds a feasible solution. To prevent I/O Bank Assigner from unassigning and re-assigning I/O technologies each time you run it, lock the I/O banks by selecting **Locked** in the I/O Bank Settings dialog box or by importing the `set_iobanks` PDC command with its `-fixed` argument set to YES.

To automatically assign technologies to I/O banks, in Project Manager, in the **I/O Attribute Editor** choose **Tools>Auto-Assign I/O Banks**.

Messages appear in the Output window informing you when the automatic I/O bank assignment begins and ends. If the assignment is successful, **I/O Bank Assigner completed successfully** appears in the Output window.

If the assignment is not successful, an error message appears in the Output window.

Note: All I/O technologies assigned to I/O banks by the I/O Bank Assigner in Layout are unlocked.

To undo the I/O bank assignments, choose **Undo** from the **Edit** menu. Undo removes the I/O technologies assigned by the I/O Bank Assigner. It does not remove the I/O technologies previously assigned.

To redo the changes undone by the Undo command, choose **Redo** from the **Edit** menu.

Common I/O Attributes

The I/O Attribute Editor displays common attributes for all I/O macros:

- **Port Name** indicates the I/O macro name.
- **Group** identifies the group name.
- **Macro Cell** indicates the type of I/O macro.
- **Pin Number** indicates the current pin assignment.
- **Locked**, if checked, indicates that you cannot change the current pin assignment during layout.

In addition to the common I/O attributes, the I/O Attribute Editor displays device-specific attributes such as I/O Standard, Skew, and Output Load. Only attributes applicable to a specific device appear in the I/O Attribute Editor table.

I/O Attributes by Family

The following table displays the attributes supported for each family.

Attribute	Family			
	IGLOO	SmartFusion and Fusion	SmartFusion2, IGLOO2, and RTG4	ProASIC3
Bank Name	X	X	X	X
I/O Standard	X	X	X	X
I/O Threshold	X, IGLOO PLUS only			
Output Drive	X	X	X	X
Slew	X	X	X	X
Resistor Pull	X	X	X	X
Schmitt Trigger	X, IGLOOe and IGLOO PLUS only	X	X	X, ProASIC3e and ProASIC3L only
Input Delay	X, IGLOOe and IGLOO PLUS only	X	X	X, ProASIC3e and ProASIC3L only
Skew	X	X	X	X
Output Load	X	X	X	X
Use Register	X	X	X	X
Hot Swappable	X	X	X	X
Hold State	X, IGLOO PLUS only			
User Reserved	X	X	X	X, ProASIC3e and ProASIC3L only

Refer to the appropriate datasheet for information about I/O standards for different families.

Note: For Fusion and ProASIC3L devices, not all attributes apply to all banks for a given I/O standard. Refer to the Fusion and ProASIC3L datasheets for details.

See Also

Common I/O attributes (all families)

I/O Attributes by Family (all families)

Floorplanning

Chip Planner and Floorplanning for SmartFusion2, IGLOO2, RTG4

The Chip Planner is a graphical interface tool that allows you to create regions, edit regions, and make logic assignments to these regions. It is a floorplanning tool used to improve the timing performance and routability of your design by providing maximum control over your design placement.

You can also cross-probe from SmartTime into Chip Planner to browse your design and investigate timing problems.

Use Chip Planner to:

- View macro assignments made during layout
- Assign, unassign, or move macros
- Lock macro assignments
- View net connections using a ratsnest or route view
- View architectural boundaries
- View and edit silicon features, such as I/O banks
- Create Regions and assign macros or nets to regions (floorplanning)
- View logic placement and net connections to investigate timing problems together with SmartTime's Cross-Probing feature

For more information, refer to the Chip Planner User Guide.

Starting Chip Planner - SmartFusion2, IGLOO2, RTG4

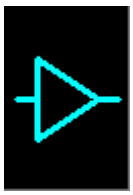
Chip Planner requires a compiled design. If you start Chip Planner before compiling your design, Libero SoC guides you through the compile process before opening Chip Planner.

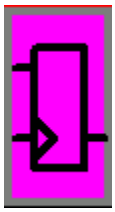
- For Classic Constraint Flow:
  In the Design Flow window; under **Implement Design > Edit Constraints**, double-click **Chip Planner**.
- For Enhanced Constraint Flow:
  In the Design Flow window; under **Constraints** double-click **Manage Constraints** to open the Constraint Manager window.
    - Click on the **Floor Planner** tab
    - Click on the **Edit with Chip Planner** button

Colors and Symbols

Colors and symbols differentiate the I/O and logic macros in ChipPlanner. The following table defines the default colors assigned to symbols. You can change these colors per design.

Color	Definition
White Border	A white border denotes a selected object.
Black Background	A black background denotes an unused or unassigned module.
Blue	Blue denotes a combinatorial module.

Color	Definition
Yellow	Yellow denotes *locked* logic modules. If the module is selected, the symbol appears yellow. If the module is unselected, the border appears yellow.
Green	Green denotes I/O modules.
Red	Red denotes clock modules.
Magenta	Magenta denotes sequential modules.
	Reserved modules that are not user definable are gray, crossed-out symbols on a black background.
	Clock modules are red. Unused/unassigned modules are red symbols on a black background. Used/assigned modules are black symbols on a red background.
	Input/Output modules are green. Unused/unassigned modules are green symbols on a black background. Used/assigned modules are black symbols on a green background.
	Combinatorial modules are blue. Unused/unassigned modules are blue symbols on a black background. Used/assigned modules are black symbols on a blue background.
	Sequential modules are magenta. Unused/unassigned modules are magenta symbols on a black background. Used/assigned modules are black symbols on a magenta background.
	Buffer modules are blue.
	RAM modules are green. Unused/unassigned modules are green symbols (RAM) on a black background. Used/assigned modules are black on a green background.
	PLL modules are green. Unused/unassigned modules are green symbols (PLL) on a black background. Used/assigned modules are black on a green background.

Color	Definition
	
	I/O Inbuff modules are pink on a black background. Used/assigned modules are black on a pink background.

Changing Colors

You can control which objects are visible in your design and what color they are.

To set display properties:

1. From the **View** menu, choose **DisplaySettings**. The Display Settings dialog box displays a list of all the architectural features you can turn on and off in your tool.
2. To make an object visible, select the **Visible** check box.
3. To change the color used to display the object, click its color bar and select another color.

To save or open previously saved display settings, click:

1. **Save** to save your display settings to a file.
2. **Load** to open a saved display settings file.
3. **Default** to load the default display settings.
4. Click **Apply** to see your changes.
5. Click **OK** to dismiss the dialog box.

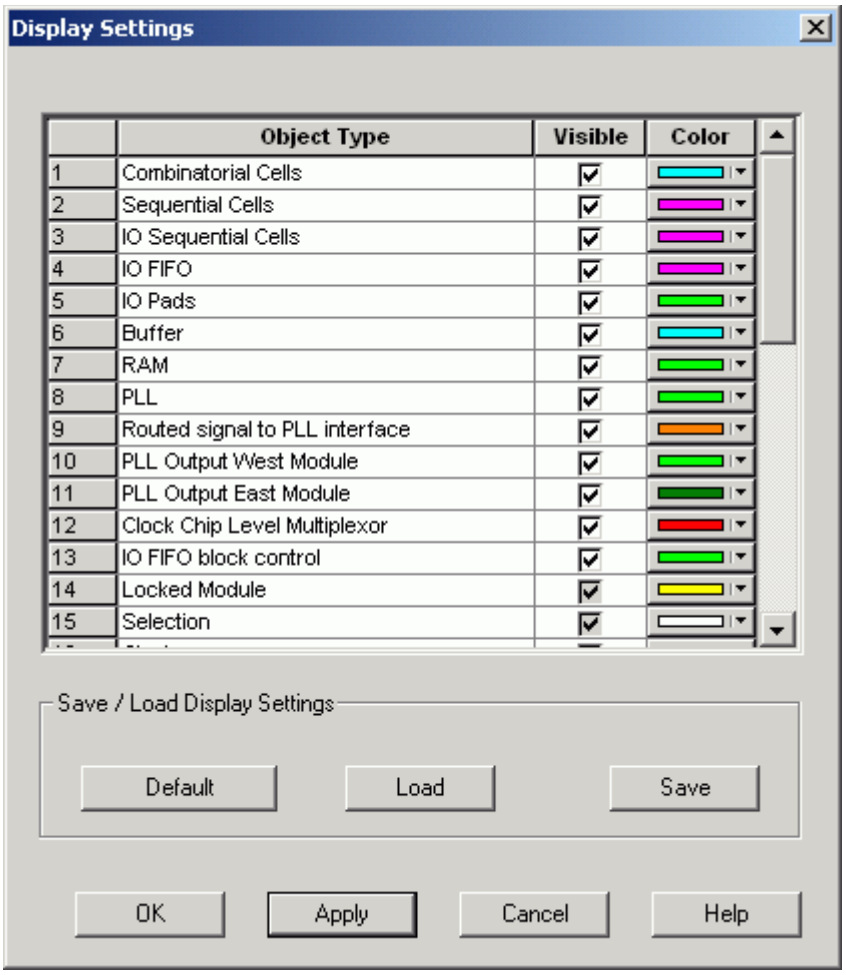


Figure 29 · Display Settings Dialog Box in ChipPlanner

Floorplanning

Floorplanning includes creating regions and making logic assignments to those regions. It is an optional methodology you can use to improve the performance and routability of your design. The objective in floorplanning is to assign logic to specific regions on the chip to enhance performance and routability.

When floorplanning, you analyze your design to see if certain logic can be clustered within regions. Clustering is especially helpful for hierarchical designs with plenty of local connectivity within a block. If your timing analysis indicates several paths with negative slack, try clustering the logic included in these paths into their own regions. This forces the placement of logic within the path closer together and may improve timing.

Use ChipPlanner before and after running Layout to help you floorplan.

See Also

Creating Regions

Assigning a Macro to a Region

Assigning a Net to a Region

About Regions

A region is a user-defined area on the device. When floorplanning, you can assign logic to regions to improve the design performance. You can use ChipPlanner to create regions or create them with PDC commands.

ChipPlanner supports the following types of regions:

Logic Region (Inclusive/Exclusive)

A logic region is a region that has logic assigned to it. Logic can include core logic, memory, and I/O modules. The place-and-route tool will place all the logic assigned to a logic region inside that region. The floorplanning process usually requires you to create several regions and assign logic to them.

Logic regions can either be inclusive or exclusive. If a logic region is exclusive, it means that the place-and-route tool cannot place any logic within the region other than what you have previously assigned to it. If a logic region is inclusive, the place-and-route tool can place any logic within the region.

Empty Region

To prevent logic from being placed within a predefined area in the device, you can create an empty region. The place-and-route tool will not place any logic within an empty region; however, the routing resources within the region can be used.

Local Clock Regions

Not available for all families; see Using Local and Quadrant Clock Regions for supported families.

A local clock is a portion of the global clock network on the device. Local clock regions are inclusive by default and cannot be changed. Each family has different local clock capabilities. For specific details, see the datasheet for your device.

Quadrant Clock Regions

Not available for all families; see Using Local and Quadrant Clock Regions for supported families.

A quadrant clock is a portion of the global clock network on the device. Each family has different quadrant clock capabilities. For specific details, see the datasheet for your device. You create and delete a quadrant clock in the same way that you create a local clock in IGLOO and ProASIC3 devices.

Overlapping Regions

If you create Logic regions whose areas intersect, the regions are defined to be overlapping. The place-and-route tool will detect the area where these regions intersect and try to place logic common to both of them within this area.

Blocks and Regions

Nets driving a block can be assigned to regions, local clocks, and quadrant clocks. VREF I/Os are not supported in the top design if the block has I/Os.

See Also

Creating Regions

Using Empty Regions

Using Logic Regions

Using Local Clock and Quadrant Clock Regions

Editing Regions

Creating Regions

With ChipPlanner, you can create empty, exclusive, inclusive, QuadrantClock and LocalClock regions under certain conditions:

Table 1 · Types of Regions

Region Type	Conditions
Empty	• Cannot assign macros to an empty region • Cannot create empty regions in areas that contain locked macros
Exclusive	• Only contains macros assigned to the region
Inclusive	• Contains all macros, both assigned and unassigned to the region
LocalClock	• Can create LocalClock regions for SmartFusion, IGLOO, ProASIC3 and Fusion devices in a PDC file • Cannot resize or move a LocalClock region • Cannot assign logic to a LocalClock region
QuadrantClock	• Can assign CORE, RAM, and I/Os to QuadrantClock regions that are inclusive • Can create QuadrantClock regions only for IGLOO and ProASIC3 devices

Note: To create an empty or logic region:

1. From the **Region** menu, choose **CreateEmpty, Create Exclusive,** or **Create Inclusive**.
2. Click and drag the mouse over the area where you want the region to be placed. As you drag out the region, a tooltip appears in its lower-right corner, showing you how many tiles, RAMs, and I/Os are in the region.

To create a LocalClock region (not available for all families; see Using Local and Quadrant Clock Regions for supported families):

1. In the **Net** tab of the Hierarchy window, select a clock net. Clock nets have a clock icon next to them in the Net view.
2. From the **Region** menu, choose **Create LocalClock**,or click its icon in the toolbar.
3. Click and drag a rectangle from the top-left corner of the new LocalClock region to its bottom-right corner. As you drag out the region, a tooltip appears in its lower-right corner, showing you how many tiles, RAMs, and I/Os are in the region.

To create a QuadrantClock region (not available for all families; see Using Local and Quadrant Clock Regions for supported families):

1. In the **Net** tab of the Hierarchy window, select a clock net. Clock nets have a clock icon next to them in the Net view.
2. From the **Region** menu, choose **Create QuadrantClock**,or click its icon in the toolbar. Select a point in the Chip appears in the status bar.
3. Select a point in the chip that is driven by QCLK. ChipPlanner creates a QuadrantClock region for the given net. The region name is Qclock_<netname> and appears in the **Regions** tab of the Hierarchy View window. A tickmark appears next to the net in the Net tab and next to all the macros driven by it.

See Also

Creating LocalClock Regions

Creating QuadrantClock regions

Empty Regions

Empty regions allow you to create exclusive areas on the device where no logic placement can occur. Empty regions help guide the placer to pack your logic closer together and thereby use more local routing resources to connect it. You cannot create empty regions in areas that contain locked macros. Use the following guidelines for empty regions.

Use Empty Regions to Guide the Place-and-Route Process

If your design does not completely use up your target device (for example 60% utilization or lower), use empty regions to cluster your logic placement into specific subareas of the chip. This helps when you have originally placed-and-routed the design into a smaller device but want to fit it to a larger part while still preserving the performance you have achieved in the smaller device.

Use Empty Regions to Reduce Routing Congestion

Creating empty regions next to the congested area(s) of your design helps reduce congestion. When you place an empty region next to congested logic blocks or regions, the placer cannot place any logic next to your region or logic block. Logic, which would normally be placed there, is forced to be placed somewhere else. Routing resources next to the congested area are, therefore, freed up and provide the router more options to route signals into the congested block.

Before deciding to place empty region(s), analyze your design for congestion areas. Use the **Ratsnest** view in ChipPlanner to see dense areas of connectivity into and out of your logic blocks or regions. Create empty regions in these congested areas and see it if improves the routability of your logic.

Use Empty Regions to Reserve Device Resources

If you want to preserve the placement of your existing design but plan additional modifications in the future, create empty regions in the areas of the chip where you plan to add additional logic. As you add new logic, remove or resize your empty regions accordingly to fit your new logic. Empty regions placed over I/O pins reserve them for future use as the I/O needs of your design changes. There are some restrictions for using empty regions in this manner.

See Also

Creating Regions

Using Logic Regions

Using LocalClock and QuadrantClock Regions

Editing Regions

Logic Regions

Use Logic regions to condense the placement of certain logic blocks in your design. This allows you to control logic placement at the region or block level. This may simplify your floorplanning task, since you might not have to place logic instances individually on the device. The following sections contain guidelines for using Logic regions.

Use Logic Regions to Localize Placement of Logic Blocks

If you partitioned your design into several modules, and some of these modules contain regular structures (such as arithmetic logic, register arrays, counters, or multiplexors), place these modules into Logic regions. These logic functions have a good amount of both local connectivity and regularity to their structure, which

makes them good candidates for regions. Interconnects between your regions now become interconnects between hierarchical blocks in your design. Floorplan your regions so there is a smooth horizontal or vertical data flow between each Logic region.

For Pipelined Logic, Place Registers on Region Boundaries

If you assigned logic to a region so its inputs and outputs are bounded by a register array (pipeline registers), it is a good idea to place these pipeline registers close to the boundary of the region. If you plan to manually fix the placement of your pipeline registers, make sure you orient them in the correct direction to assure a smooth data flow between them and their interfacing logic.

Aligning RAM I/O with Placement

Before placing your memory blocks, review your design and understand how data is flowing into and out of them. Determine what logic blocks are driving the memory inputs (for example, address line, control signals) and what logic is driven by the memory outputs (for example, databus lines). Follow these guidelines:

- Place pins that drive or are driven by your memory blocks close to where your memory blocks are placed.
- Create an empty region next to your memory block to free up local routing resources that may need to be used to connect to the memory blocks.
- If you are driving high fan-in memory inputs such as read/write clocks or read/write enables, try using low-skew routing resources such as global nets or clock spines to connect them. Make sure your clock spine assignments are aligned with your RAM placement.

See Also

Creating Regions

Using Empty Regions

Using LocalClock and QuadrantClock Regions

Editing Regions

Editing Regions

After creating regions, with the exception of LocalClock regions, you can rename, delete, move, and re-size them. LocalClock regions can only be renamed or deleted.

Regions must have unique names. Two regions cannot have the same name.

To change the name of a region:

1. In the **Hierarchy** window, click the **Regions** tab.
2. Select the region with the name you want to change.
3. From the **Region** menu, choose **Properties**.
4. In the **Properties** dialog box, type the new region name over the existing one.

Tip: You can also right-click a region, choose **Properties**, and type a new region name in the **Properties** dialog box.

To delete a region:

- Right-click the region, and chose **Delete**.

To move a region:

- Select the region and drag it to a new location.

Note: You cannot move the region if a macro assigned to the region is locked.

To re-size a region:

1. Select the region.
2. Grab and drag the sides and corners to re-size the region. You cannot resize a region smaller than the logic it already contains.

To merge a region:

1. In the **Hierarchy** window, click the **Regions** tab.
2. Select the regions you want to merge.
3. Right-click any selected region and choose **Merge**.

   The merge operation:

- Creates a new region with rectangles of all selected regions
- Assigns logic and nets that were assigned to the selected region to the new region
- Removes the selected regions

See Also

Creating Regions

Using Empty Regions

Using Logic Regions

Using Local Clock and Quadrant Clock regions

Locking Regions

Assigning a Macro to a Region

During floorplanning, you can improve design performance by assigning macros to regions.

Note: You can use the Logical or Physical tab right-click menus to bypass the Assign Instances to Region dialog box and assign instances to regions directly. To do so, in the Hierarchy window, Logical/Physical tab, select one or more instances, right-click and choose Assign to Region.

Note: You can also use the right-click menu to assign instances in the results from the ActiveLists and in results from the Find command (in the Log window).

To assign a macro to a region:

1. In the Regions tab of the Hierarchy window, right-click a region and choose **Assign/Unassign Instance** from the right-click menu. The **Assign Instances to Region** dialog box appears with the name of the selected region in the **Region name** field. The dialog box displays all the macros that you can assign to the selected region.

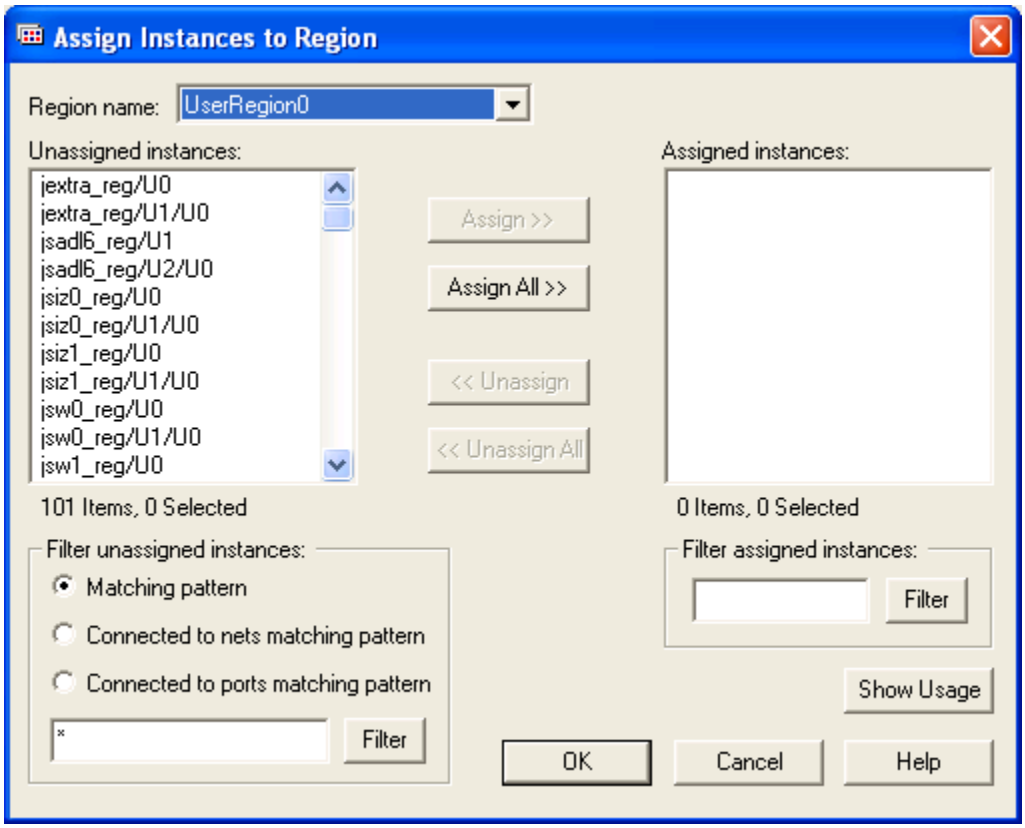


Figure 30 · Assign Instances to Region Dialog Box

2. To display a subset of the unassigned instances, you can create and apply filters. To filter the unassigned instances list by a pattern, enter the pattern (string) in the text box to the left of the Filter button, select **Matching pattern**, and then click **Filter**. Only the instances that match the pattern appear in the **Unassigned Instances** list. For example, enter *U18* to display only unassigned instances containing the characters U18. You can also display only instances connected to net or port names matching the specified pattern. These filters are valid for both regular and Block flows. In regular flows, each port is connected to only one instance, which is an I/O.
3. To assign specific instances to the region, select one or more instances in the **Assignable Instances** list box, and then click **Assign**. To assign all instances to the region, click **Assign All.**
4. Click **OK**.

The total number of instances that you can assign as well as the number of currently selected unassigned instances appears under the list box.

Tip: You can also assign logic to regions from the Hierarchy window. To do so, drag and drop the logic from the Hierarchy window to the region.

See Also

Unassigning a Macro from a Region

Assigning a Net to a Region

Assigning a net to a region results in assigning all instances connected by the net to the specified region. The assignment of all instances of a net to a region packs logic more closely together, which improves (reduces) net delays.

Assigning nets to regions also enables you to apply floorplanning constraints correctly over design iterations. During design iterations you may change your design, causing a change in the instances connected by a particular net. The assignment of a net to a region assigns all instances connected by that net in the current design iteration, enabling your constraints to be applied correctly.

Note: You can use the right-click menu to assign a net to a region. To do so, select one or more nets in the Nets tab of the Hierarchy menu, right-click, and choose Assign to Region. Using the right-click menu assigns all instances except the driver macro.

Note: You can also use the right-click menu to assign nets in the ActiveList, and in the results from the Find command (in the Log window).

To assign a net to a region:

1. In the Regions tab of the Hierarchy window, right-click a region and choose **Assign/Unassign Net** from the right-click menu.

   The **Assign Nets to Region** dialog box appears with the name of the selected region in the **Region name** field. The dialog box displays all the nets that you can assign to the selected region (as shown in the figure below).

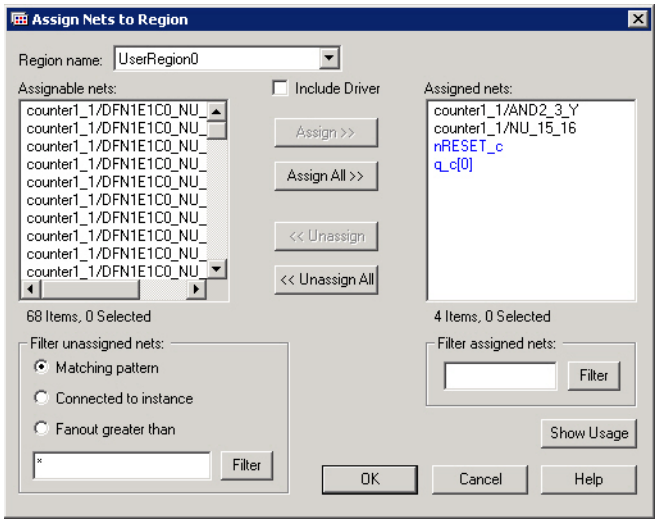


Figure 31 · Assign Nets to Region Dialog Box

   To display a subset of the unassigned nets, you can create and apply filters.

2. To filter the unassigned nets list by a pattern, enter it in the text box to the left of the Filter button, select **Matching pattern**, and then click **Filter**. Only the nets that match the pattern appear in the Assignable nets list. For example, enter *INV* to display only unassigned nets containing the characters INV.
3. To assign specific nets, select one or more nets in the **Assignable nets** list box, and click **Assign**.

   To select all nets, click **Assign All**.
   Check the Include Driver checkbox to assign all instances, including the driver connected by the selected net, to the region.

3. Click **OK** to continue.

The total number of nets that you can assign, as well as the number of currently selected unassigned nets, appears in the list box.

Tip: You can assign nets from the Hierarchy window. To do so, drag and drop the nets from the Hierarchy window to the region in ChipPlanner.

See Also

Editing Regions

Assigning a Macro to a Region

Assigning Nets to Regions Dialog Box

Unassigning a Macro from a Region

To unassign a macro from a region:

1. In the **Hierarchy** window (Logical or Physical tab), select the macro to unassign.
2. From the **Logic** menu, choose **Unassign From Region**.

Tip: You can also right-click the macro, and choose **Unassign From Region**.

You can also unassign all macros from a specific region.

To unassign all macros from a region:

1. Select the region from which to unassign all macros.
2. From the **Logic** menu, choose **Unassign All From Region**.

See Also

Assigning a Macro to a Region

Assigning Logic to Locations

Manually assigning logic is an optional methodology to help you improve the performance and density of your design.

You do not need to manually assign logic to particular locations in your design. However, should you have specific design requirements, ChipPlanner allows you to have maximum control over your design.

To assign logic to specific locations:

1. Select the logic in the **Physical** tab of the **Hierarchy** window or in an Active List.
2. Drag the logic to the desired location. As you drag, valid assignment locations are highlighted. To remove the assignment, from the **Edit** menu, choose **Undo**.

If the logic assignment is valid, the logic is assigned and locked. To save changes for this design session, commit your changes when exiting the MultiView Navigator.

To assign logic to multiple locations:

1. While holding down the **CTRL** or **SHIFT** key, select the logic in the order you want it placed.
2. From the **Logic** menu, choose **Assign To Location**.
3. One by one, select the desired locations. The macros are placed in the order selected.

To unassign logic from a location:

1. Select the logic.
2. From the **Logic** menu, choose **Unassign From Location.**

To unassign logic from multiple locations:

1. While holding down the **CTRL** or **SHIFT** key, select the logic you want to unassign.
2. From the **Logic** menu, choose **Unassign From Location**.

To unassign logic from all locations:

- From the **Logic** menu, choose **Unassign All From Location.**

See Also

Moving Logic to Other Locations

Locking Logic to Locations

Unassigning All Logic from a Location

To unassign all logic from a location:

- From the **Logic** menu, choose **Unassign All From Location.**

Moving Logic to Other Locations

You can move logic that was assigned manually or automatically during Layout.

To move logic:

1. Select the logic to move.
2. Drag the logic to the new location.

Tip: To remove the assigned macro, from the **Edit** menu, choose **Undo**.

See Also

Assigning Logic to Locations

Locking Logic to Locations

Locking Logic to Locations

Locked logic is not moved during Layout. Locked logic only becomes permanent if you commit the changes to your design before exiting.

To lock macros:

1. Select the macro to lock. To select multiple macros, hold down the **CTRL** key and select multiple macros with your mouse. To select all macros, from the **Edit** menu, choose **Select All**.
2. From the **Logic** menu, choose **Lock**.
3. From the **File** menu, choose **Commit** to save your changes in this session. To save your changes in the design file (.ADB), you must save your design in Designer as well.

To unlock a macro:

1. Select the macro. To select multiple macros, hold down the **CTRL** key and select multiple macros. To select all macros, from the **Edit** menu, choose **Select All**.
2. From the **Logic** menu, choose **Unlock**.

See Also

Assigning Logic to Locations

Moving Logic to Other Locations