Libero SoC v2023.1

Timing Constraints Editor User Guide

Introduction (Ask a Question)

The Timing Constraints Editor allows you to create, view, and edit timing constraints. It includes a powerful user interface that allows you to capture your timing requirements and timing exceptions.

Table of Contents

1. About the Timing Constraints Editor (Ask a Question)

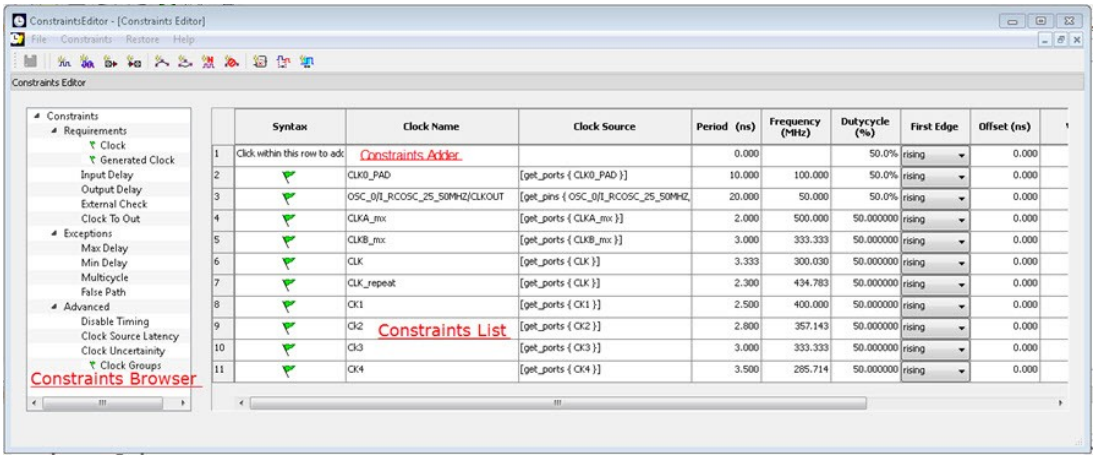
This chapter provides an overview of the Timing Constraints Editor.

1.1 Constraints Editor Window (Ask a Question)

The **Constraints Editor** window is organized into the following areas:

- Constraint Browser
- Constraint List
- Constraint Adder

Figure 1-1. Constraints Editor



1.1.1 Constraints Browser (Ask a Question)

The **Constraint Browser** categorizes constraints based on the following types of constraints:

- Requirements are constraints to meet the design's timing requirements and specifications. Examples are clock constraints and generated clock constraints.
- Exceptions are constraints on certain timing paths for special considerations. Examples are false path constraints and multicycle path constraints.
- Advanced are special timing constraints such as clock latency and clock groups.

1.1.2 Constraints List (Ask a Question)

The **Constraints List** is a spreadsheet of constraints with detailed values and parameters of the constraint displayed in individual cells. You can click the individual spreadsheet cells to change the values of the constraint parameters.

1.1.3 Constraints Adder (Ask a Question)

Constraints Adder is the first row of the constraint list spreadsheet. There are two ways to add a constraint from this row.

- To add a constraint of the same type to the **Constraint List**, right-click a row and select **Add Constraint**. This method displays the specific add constraint dialog.
- Click in a cell, and then double-click and start typing. This method is for the experienced users who know the design well and do not need to rely on the dialog box for guidance.

You can perform the following tasks in the Constraints View:

- From the **Constraint Browser**, select a constraint type, and then create or edit the constraint.
- Add a new constraint and check the syntax.
- Right-click a constraint in the **Constraints List** to edit or delete.
- Use the first row to create a constraint, and then add it to the **Constraints List**.

1.1.4 Constraint Editor Icons (Ask a Question)

You can click the icons across the top of the Constraint Editor to add constraints. Tool tips are available to identify the constraints. The following table lists the the icons.

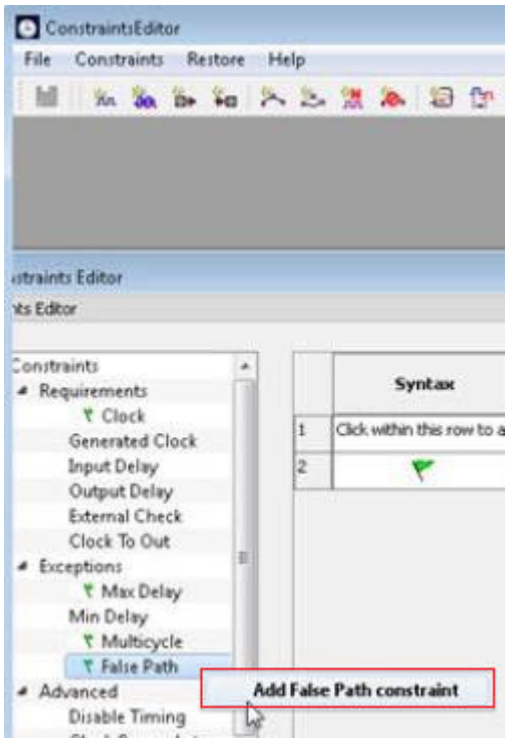
Table 1-1. Constraint Icons

Icon	Name (Tool tip)
	Add Clock Constraint
	Add Generated Clock Constraint
	Add Input Delay Constraint
	Add Output Delay Constraint
	Add Maximum Delay Constraint
	Add Minimum Delay Constraint
	Add Multicycle Path Constraint
	Add False Path Constraint
	Add Disable Timing Constraint
	Add Clock Source Latency
	Add Clock to Clock Uncertainty

1.2 Adding Constraints (Ask a Question)

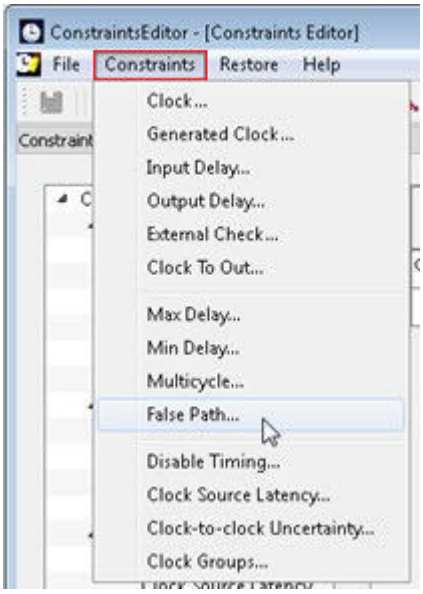
The Constraints Editor provides four ways to add constraints using the **Add Constraints** dialog box.

1. Option 1: Click the **Add Constraint**  icon.
2. Option 2:
   In the **Constraints Browser**, click the type of constraints you want to add.
   Figure 1-2. Adding Constraints Using Constraints Browser



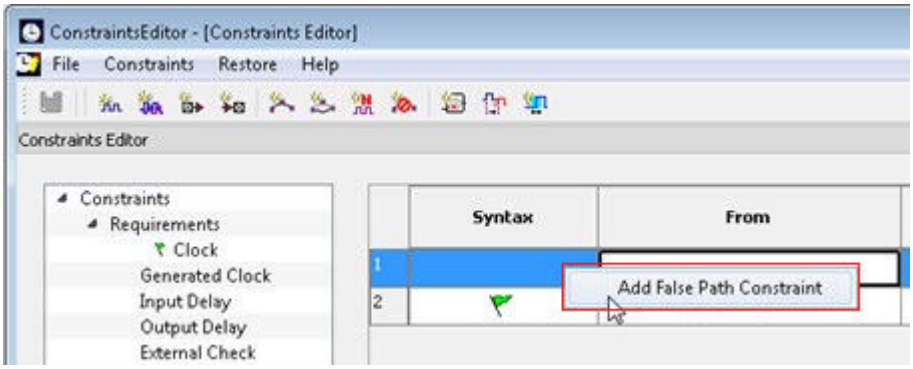
3. Option 3:
   From the **Constraints** menu, click a constraint.

Figure 1-3. Adding Constraints Using Constraints Menu



4. Option 4:
   In the **Constraints Browser**, click the type of constraints you want to add. Right-click the first row and in the context menu that appears, click the constraint.

Figure 1-4. Adding Constraints Using Constraint Adder



2. Required Constraints (Ask a Question)

This chapter describes how to add constraints. It also lists the supported constraints.

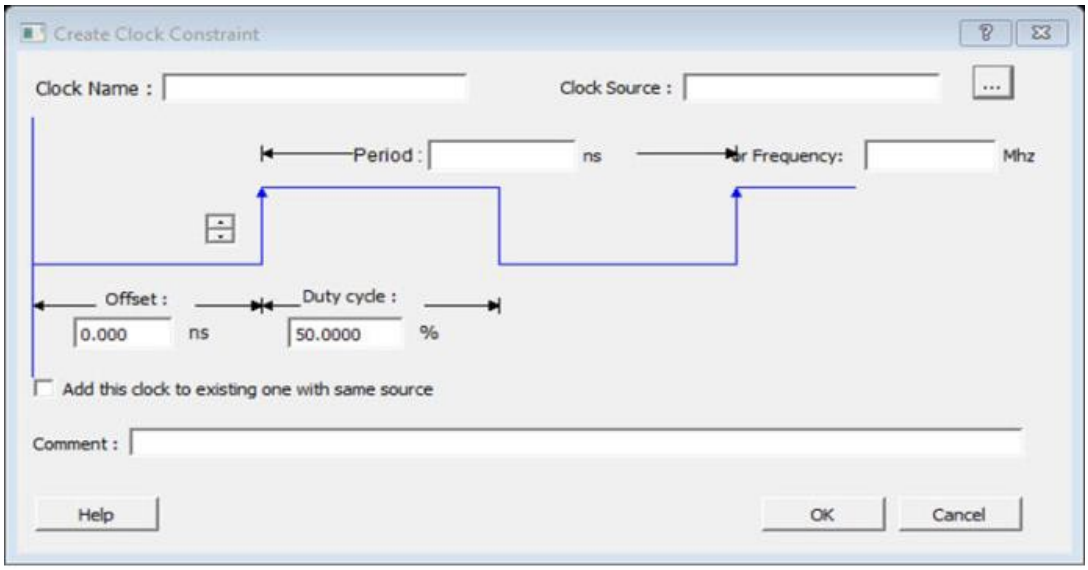
2.1 Set a Clock Constraint (Ask a Question)

Adding a clock constraint is the most effective way to constrain and verify that the timing behavior of a sequential design meets your performance goals.

To set a clock constraint, use one of the following ways to open the **Create Clock Constraint** dialog box:

- In the **Constraints Browser**, double-click **Clock**.
- Click the **Add Clock Constraint**  icon.
- In the **Constraints** list, click **Clock**.
- In the **Clock Constraints Table**, right-click the first row or any other row (if they exist) and click **Add Clock Constraint**.

Figure 2-1. Create Clock Constraint Dialog Box



The following table lists all the **Create Clock Constraint** dialog box options.

Table 2-1. Create Clock Constraint Options

Option	Description
Clock Name	Specifies the name of the clock constraint.

.....continued	
Option	**Description**
Clock Source	Select the pin to use as clock source. To display the **Select Source Pins for the Clock Constraint** dialog box, click **Browse**. The following options are available on the **Select Source Pins for the Clock Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are: – Input Ports – All Pins – All Nets • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list displays the available Clock Pins. If you change the pattern value, the list box shows the available pins based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays the pins selected in the **Available Pins** list. To add the source pins to the constraint, click pins from this list and click **OK**.
Period/ Frequency	Specifies the period in nanoseconds (ns) or frequency in MegaHertz (MHz). When you edit the period, the tool updates the frequency value automatically. The frequency must be a positive real number. Accuracy is up to three decimal places.
Starting Clock Edge Selector	Click the up or down arrow to use the rising or falling edge as the starting edge for the created clock.
Offset	Indicates the shift (in nanoseconds) of the first clock edge with respect to instant zero common to all clocks in the design. The offset value must be a positive real number. Accuracy is up to two decimal places. Default value is 0.
Duty cycle	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 four decimal places. Default value is 50%.
Add this clock to existing one with same source	Select this check box to add a new clock constraint on the same source without overwriting the existing clock constraint. The new clock constraint name must be different than the existing name. Otherwise, the new constraint overwrites the existing one, even if you select this check box.
Comment	Enter a single line of text that describes the purpose of the clock constraints.

2.1.1 Specifying Clock Constraints (Ask a Question)

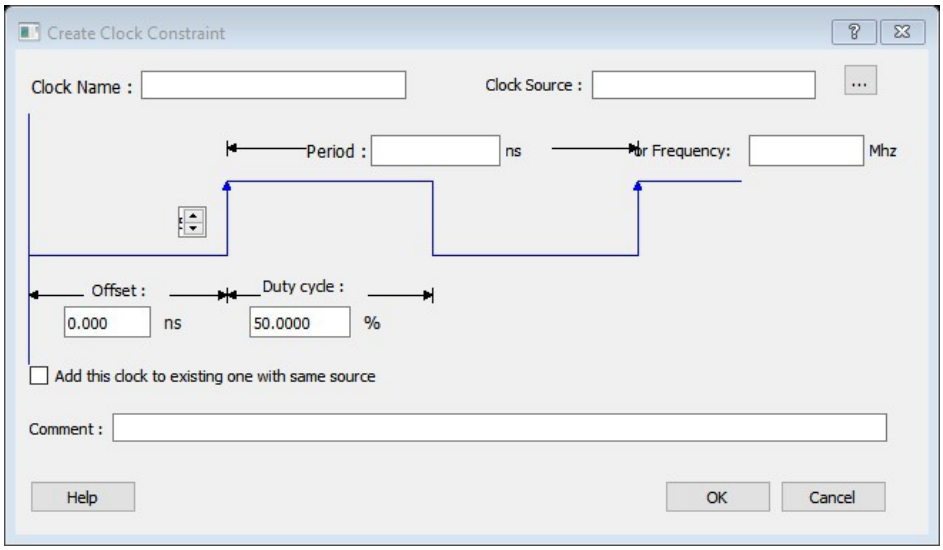
Specifying clock constraints is the most effective way to constrain and verify the timing behavior of a sequential design. Use clock constraints to meet your performance goals.

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:
   - In the **Constraints Editor**, click the  icon.
   - In the Constraint Browser, right-click the **Clock** and select **Add Clock Constraint**.
   - In the Constraint Browser, double-click **Clock**.

– In the Constraints list (**Constraints > Clock**), select **Clock**. The **Create Clock Constraint** dialog box appears.

Figure 2-2. Create Clock Constraint Dialog Box



2. Select the pin to use as the clock source. You can click **Browse** to display the **Source Pins for Clock Constraint** dialog box.
   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 is used to synchronize I/Os.
3. 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 select 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.
   Note: Multiple source pins can be specified for the same clock when a single clock is entering the FPGA using multiple inputs with different delays.
4. To save these dialog box settings, click **OK**.
5. Specify the **Period** in nanoseconds (ns) or **Frequency** in megahertz (MHz).
6. Modify **Clock Name**. The name of the first clock source is provided as default.
7. Modify **Duty cycle**, if needed.
8. Modify **Offset** of the clock, if needed.
9. Modify the first edge direction of the clock, if needed.
10. Select the **Add this clock to an existing one with the same source** option, if needed.
11. Click **OK**. The new constraint appears in the **Constraints List**.
12. Click **File** > **Save**, the **Timing Constraints Editor** saves the newly created constraint in the database.

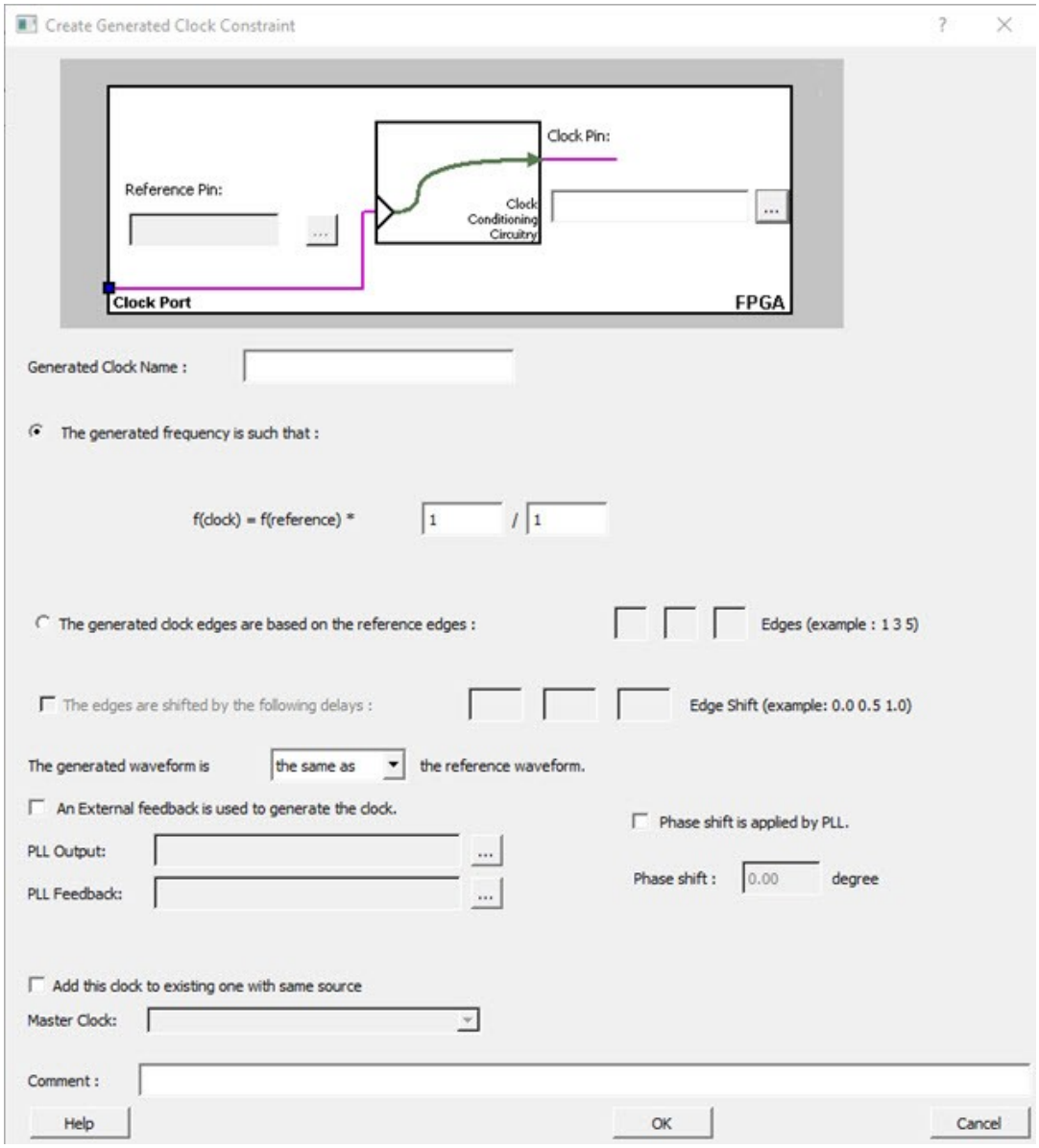
2.2 Set a Generated Clock Constraint (Ask a Question)

Use the generated clock constraint to define an internally generated clock for your design and verify its timing behavior to meet your performance goals.

To set a generated clock constraint, use one of the following ways to open the **Create Generated Clock Constraint** dialog box:

- In the **Constraints Browser**, double-click **Generated Clock**.
- Click the **Add Generated Clock Constraint** icon.
- In the **Constraints** list, click **Generated Clock** .
- In the **Generated Clock Constraints Table**, right-click any row and click **Add Generated Clock Constraint**.

Figure 2-3. Create Generated Clock Constraint Dialog Box



The following table lists all the **Create Generated Clock Constraint** dialog box options.

Table 2-2. Create Generated Clock Constraint Options

Option	Description
Clock Pin	Select a clock pin to use as the generated clock source. To display a list of the available generated clock source pins, click **Browse**. The **Select Generated Clock Source** dialog box appears. The following options are available on the **Select Generated Clock Source** dialog box: • **Pin Type**: Displays the available pin types. The pin type options for the generated clock reference are: – Output Ports – All Register Output Pins – All Pins – All Nets – Input Ports • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Filter** to filter the available pins based on the specified pin type and pattern. The list box displays the available pins based on the filter. To save the dialog box settings, click the pins from this list and click **OK**. When prompted, modify the clock name if necessary.
Reference Pin	Specifies a clock reference. To display the list of available clock reference pins, click **Browse**. The **Select Generated Clock Reference** dialog box appears. The following options are available on the **Select Generated Clock Reference** dialog box: • **Pin Type**: Displays the available pin types. The pin type options for the generated clock reference are: – Input Ports – All Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. To filter the available pins based on the specified Pin Type and Pattern, click **Filter**. The list box displays the available pins based on the filter. To save the dialog box settings, click the pins from this list and click **OK**.
Generated Clock Name	Specifies the name of the generated clock constraint. **Note:** This field is required for virtual clocks when no clock source is provided.
Generated Frequency	Specifies the values to calculate the generated frequency. A multiplication factor, division factor, or both is applied to the reference clock to compute the generated clock. The multiplication or division factor must be a positive integer.
Generated Clock Edges	Specifies the frequency of the generated clock. The specified integer value represents the edges from the source clock that form the edges of the generated clock. For more information, see 2.2.2. Generating Clock Edges
Edge Shift	Specifies a list of three floating-point numbers that represents the amount of shift, in library time units, that the specified edges are to undergo to yield the final generated clock waveform. The floating-point values can be positive or negative: • A positive value indicates a shift later in time. • A negative indicates a shift earlier in time. For example, an edge shift of `{1 1 1}` on the LSB generated clock shifts each derived edge by 1 time unit. To create a 200 MHz clock from a 100 MHz clock, use edge `{ 1 2 3}` and edge shift `{0 -2.5 -5.0}`.
Generated Waveform	Specify whether the generated waveform is the same as the reference waveform or inverted with respect to the reference waveform. When you finish, click **OK**.

.....continued	
Option	**Description**
Phase	This field is primarily used to report the information captured from the CCC configuration process, and when constraint is auto-generated. Meaningful phase values are: 0, 45, 90, 135, 180, 225, 270, and 315.
PLL Output	Refers to the CCC GL0/1/2/3 output that is fed back to the PLL (in the CCC). It reports the information captured from the CCC configuration process and when the constraint is auto-generated.
PLL Feedback	Refers to the way that the GL/0/1/2/3 output signal of the CCC is connected to the PLL's FBCLK input. This field is primarily used to report the information captured from the CCC configuration process and when constraint is auto-generated.
Add Clock to Existing Clock	Specifies that the generated clock constraint is a new clock constraint in addition to the existing one at the same source. The name of the clock constraint should be different from the existing clock constraint. When this option is selected, master clock must be specified.
Master Clock	Specifies the master clock used for the generated clock when multiple clocks fan into the master pin. It can be selected from the list. This option is used with the add option of the generated clock.
Comment	Enter a single line of text that describes the purpose of the generated clock constraints.

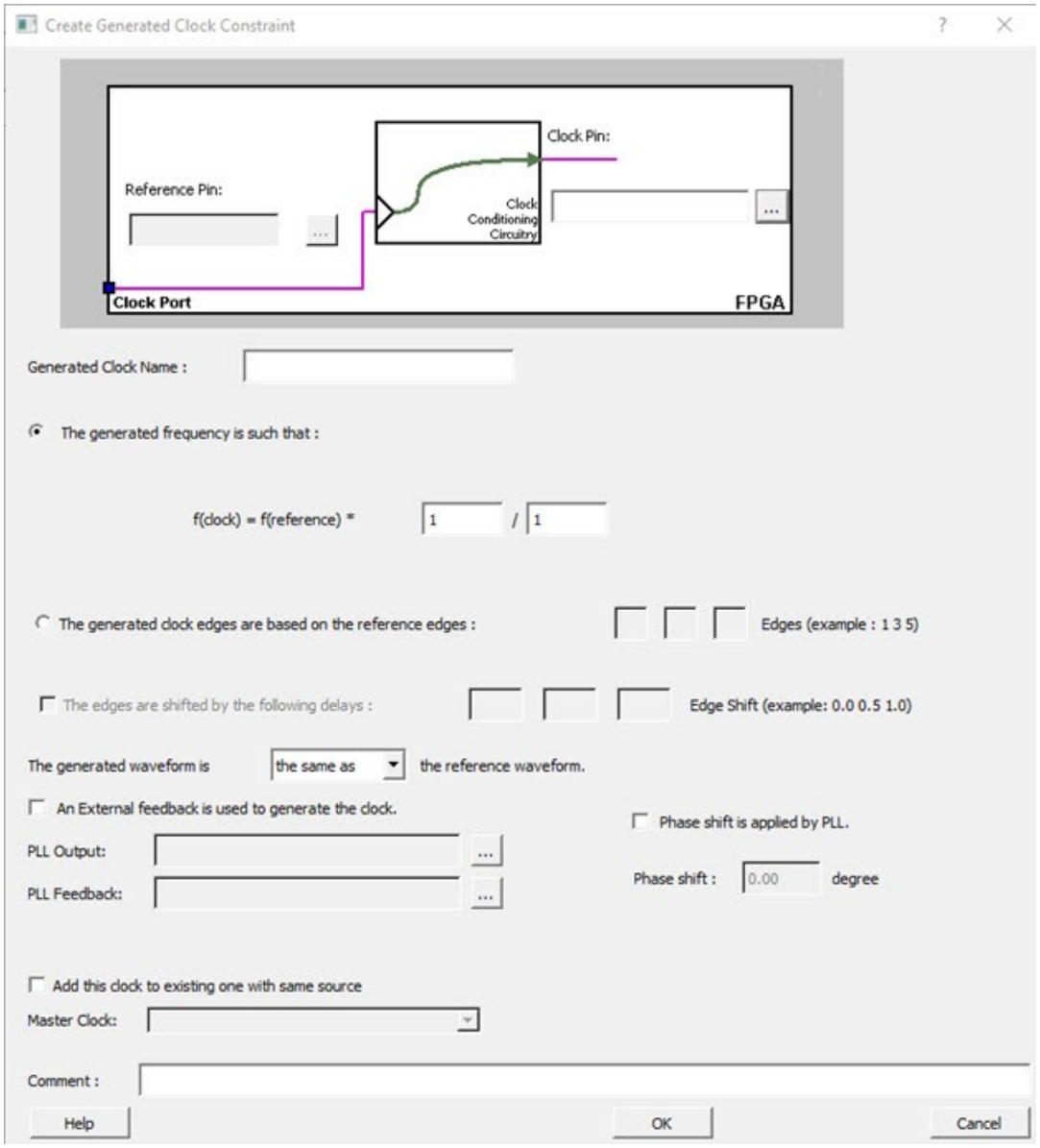
2.2.1 Specifying Generated Clock Constraints (Ask a Question)

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:

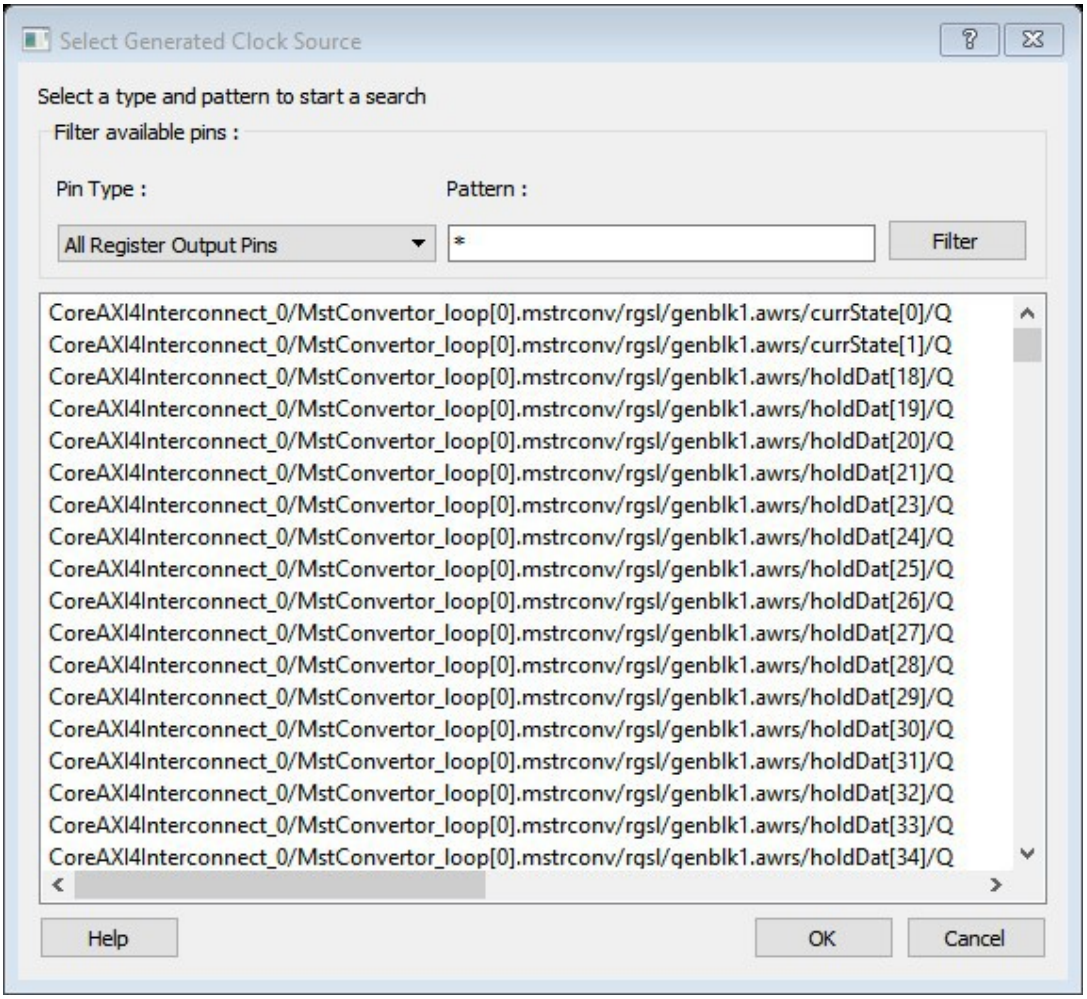
1. Open the **Create Generated Clock Constraint** dialog box using one of the following methods:
   - Click the  icon.
   - In the Constraint Browser, right-click the **Generated Clock** and click **Add Generated Clock**.
   - Double-click the **Generated Clock Constraints** grid. The **Create Generated Clock Constraint** dialog box appears.

Figure 2-4. 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 **Browse**. The **Select Generated Clock Source** dialog box appears.

Figure 2-5. Select Generated Clock Source Dialog Box



3. Specify a **Reference Pin**. To display a list of available clock reference pins, click **Browse**. The **Select Generated Clock Reference** dialog box appears.
4. Specify the **Generated Clock Name** (optional).
5. Specify the values to calculate the generated frequency: a multiplication factor and/or a division factor (both positive integers).
6. Specify the orientation of the generated clock edges based on the reference edges by entering values for the edges and the edge shifts. This is optional.
7. Specify the first edge of the generated waveform either same as or inverted with respect to the reference waveform.
8. Specify the PLL output and PLL feedback pins, if an external feedback is used to generate the clock.
9. Specify the Phase shift applied by the PLL in degrees.
10. Specify the Master Clock, if you want to add this to an existing one with the same source.
11. Click **OK**. The new constraint appears in the **Constraints List**.

Tip: Select **File** > **Save** to save the newly created constraint in the database.

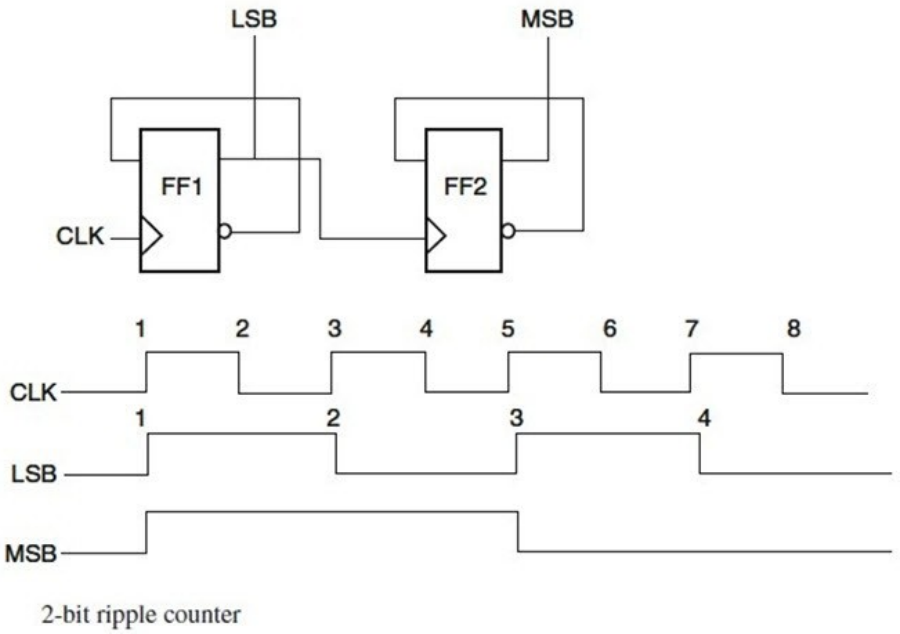
2.2.2 Generating Clock Edges (Ask a Question)

The frequency of the generated clock can also be specified by selecting the **Generated Clock Edges** option. Specify the integer values that represent the edges from the source clock that form the edges of the generated clock.

Three values must be specified to generate the clock. If you specify less than three, a tool tip indicates an error. The following example shows how to specify the clock edges.

If LSB is the generated clock from CLK clock source, the edge values must be `[1 3 5]`. If MSB is the generated clock from CLK clock source, the edge values must be `[1 5 9]`.

Figure 2-6. Example of Clock Edges



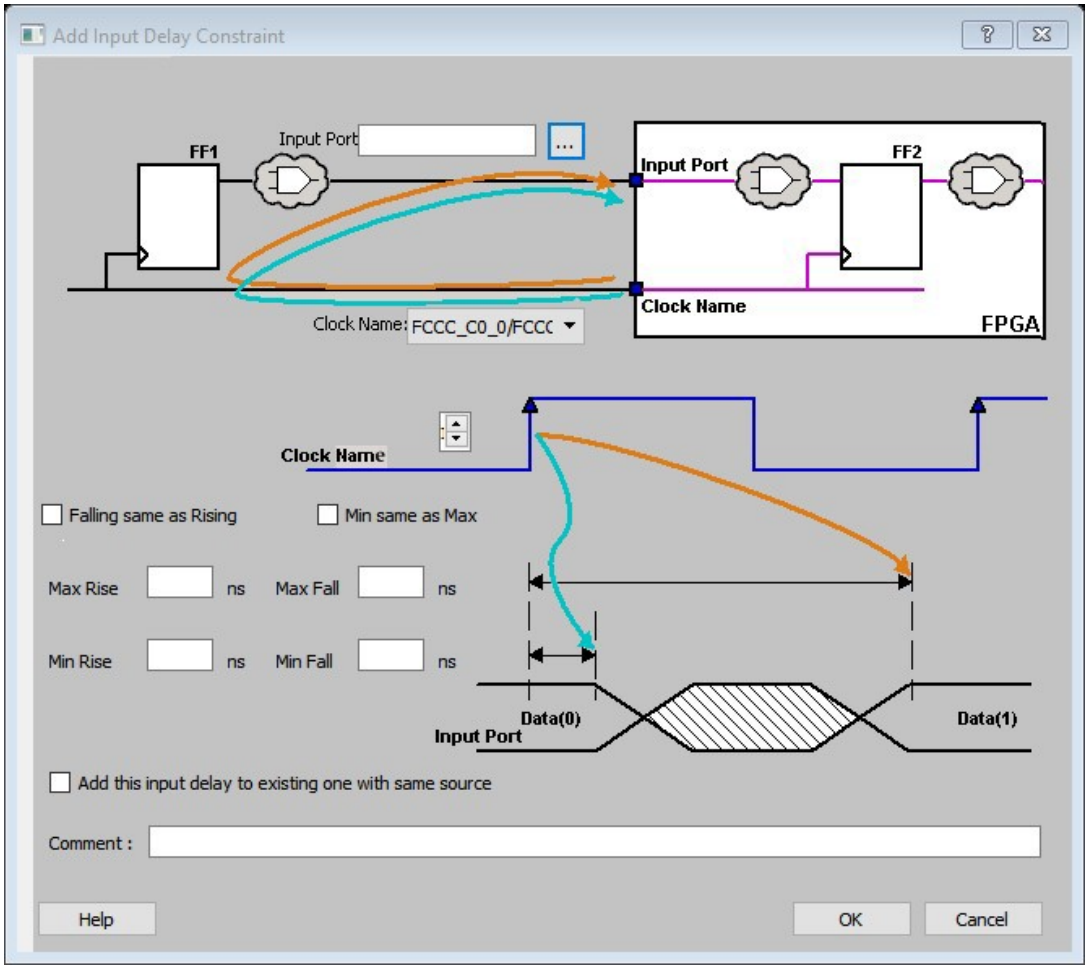
2.3 Set an Input Delay Constraint (Ask a Question)

The input delay constraint defines the arrival time of an input relative to a clock. You specify the input delay constraint in the **Input Delay** dialog box. This dialog box allows you to enter an input delay constraint by specifying the timing budget outside the FPGA. You can enter the maximum delay, minimum delay, or both.

To specify an input delay constraint, use one of the following ways to open the **Add Input Delay Constraint** dialog box:

1. In the **Constraints Browser**, double-click **Input Delay**.
2. Click the **Add Input Delay Constraint** icon.
3. From the **Constraints** menu, click **Input Delay** .
4. In the **Input Delay Constraints Table**, right-click any row and click **Add Input Delay Constraint**.

Figure 2-7. Add Input Delay Constraint Dialog Box



The following table describes the **Add Input Delay Constraint** dialog box options.

Table 2-3. Add Input Delay Constraint Options

Option	Description
Input Port	Specify the input port or click **Browse** next to **Input Port** to display the **Select Ports for Input Delay** dialog box. You can apply the input delay constraint on multiple input ports. The following options are available on the **Select Ports for Input Delay** dialog box: • **Type**: Displays the type of the available pins in the design. The only valid selection is **Input Ports**. • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available input ports. If you change the pattern value, the list box shows the available input ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays the pins selected from the **Available Pins** list. To add the input port, click the pins from this list and click **OK**.

.....continued	
Option	**Description**
Clock Name	Specifies the clock reference to which the specified input delay is based.
Clock edge	Selects rising or falling as the launching edge of the clock.
Falling same as Rising	Select this check box to use the same delay value for the falling input value and the rising input value.
Min same as Max	Select this check box to use the same delay value for min and max delay.
Max Rise and Max Fall	Specifies the delay in nanoseconds for the longest path arriving at the specified input port.
Min Rise and Min Fall	Specifies the delay in nanoseconds for the shortest path arriving at the specified input port.
Add this input delay to existing one with same source	Specifies that this input delay constraint should be added to an existing constraint on the same port(s). Use this option to capture information on multiple paths with different clocks or clock edges leading to the same input port(s).
Comment	Enter a one-line comment for this constraint.

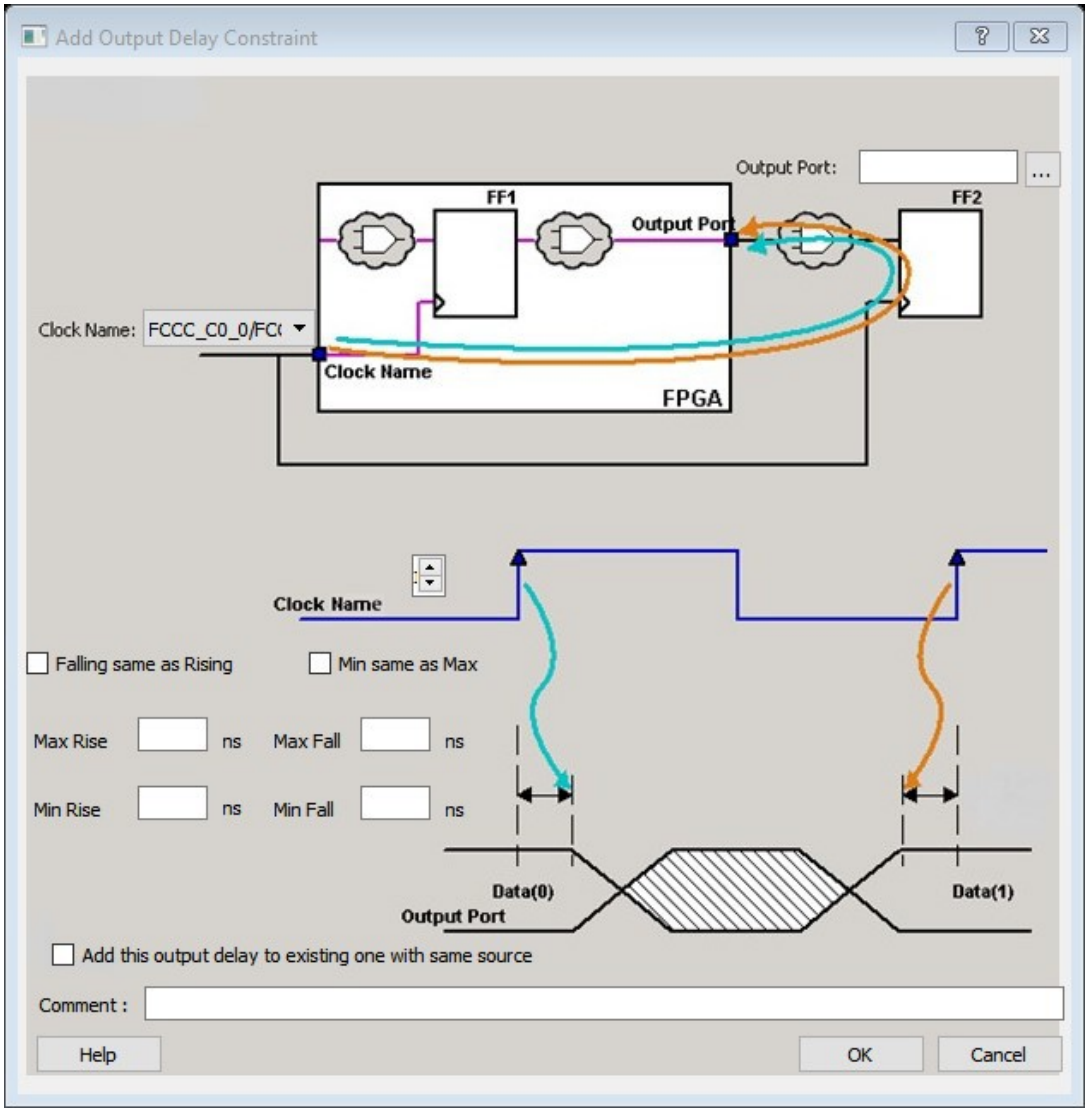
2.4 Set an Output Delay Constraint (Ask a Question)

The output delay constraints defines the output delay of an output relative to a clock. You specify the output delay constraints in the **Output Delay** dialog box. This dialog box allows you to enter an output delay constraint by specifying the timing budget outside the FPGA. You can enter the maximum delay, the minimum delay, or both.

To specify an output delay constraint, use one of the following ways to open the Add Output Delay Constraint dialog box:

- In the **Constraints Browser**, double-click **Output Delay**.
- Click the **Add Output Delay Constraint**  icon.
- From the **Constraints** menu, click **Output Delay**.
- In the **Output Delay Constraints Table**, right-click any row and click **Add Output Delay Constraint**.

Figure 2-8. Add Output Delay Constraint Dialog Box



The following table describes the **Add Output Delay Constraint** dialog box options.

Table 2-4. Add Output Delay Constraint Options

Option	Description
Output Port	Specifies a list of output ports in the current design to which the constraint is assigned. You can select multiple output ports to apply the output delay constraints. Specify the name of the output port or click **Browse** to display the **Select Ports for Output Delay** dialog box. The following options are available on the **Select Ports for Output Delay** dialog box: • **Type**: Displays the type of the available pins in the design. The only valid selection is **Output Ports**. • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available output ports. If you change the pattern value, the list box shows the available output ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the Output Ports for the Output Delay Constraint, click pins from this list and click **OK**.
Clock Name	Specifies the clock reference to which the specified output delay is related.
Clock Edge Selector	Use the Up or the Down arrow to select the rising or falling edge as the launching edge of the clock.
Falling same as Rising	Select this check box to use the same delay value for the **Falling output** value as well as the **Rising output** value.
Min same as Max	Select this check box to use the same delay value for Min and Max delay.
Max Rise and Max Fall	Specifies the delay in nanoseconds for the longest path from the specified output port to the captured edge. This represents a combinational path delay to a register outside the current design plus the library setup time.
Min Rise and Min Fall	Specifies the delay in nanoseconds for the shortest path from the specified output port to the captured edge. This represents a combinational path delay to a register outside the current design plus the library hold time.
Add this output delay to existing one with same source	Specifies that this output delay constraint should be added to an existing constraint on the same port(s). This is used to capture information on multiple paths with different clocks or clock edges leading from the same output port(s).
Comment	Enter a one-line comment for the constraint.

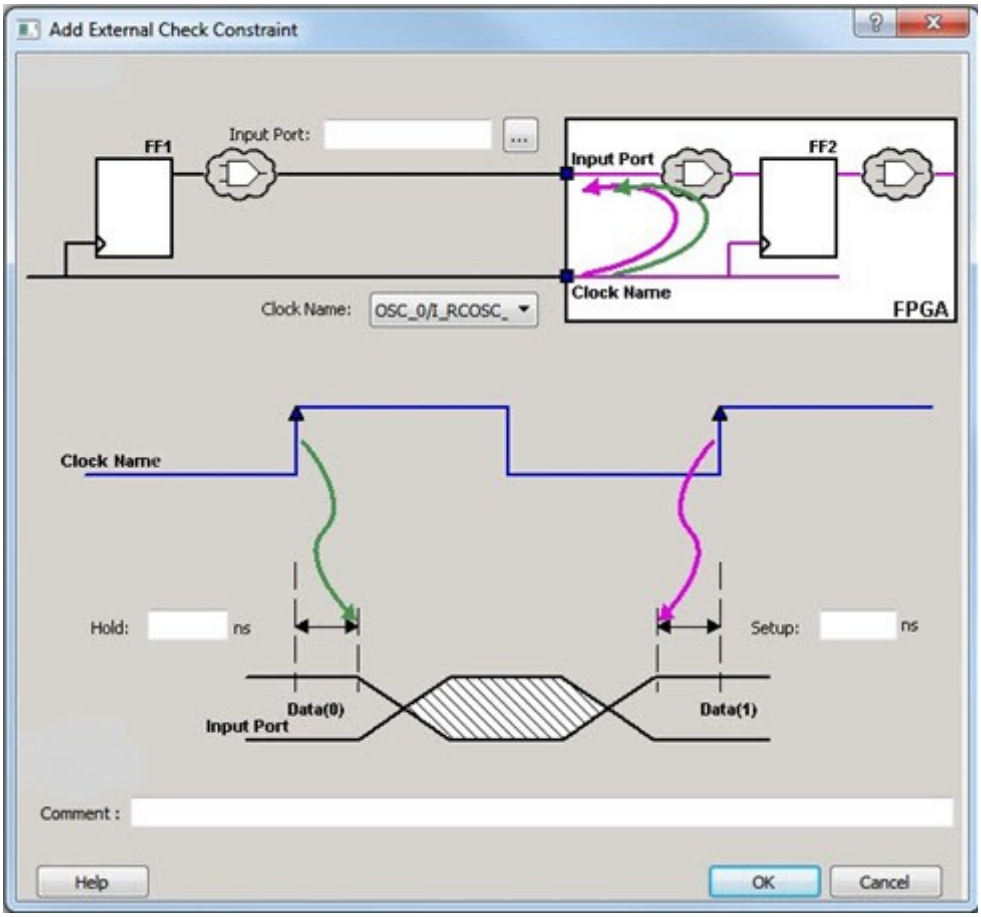
2.5 Set an External Check Constraint (Ask a Question)

Use the Add External Check Constraint to specify the timing budget inside the FPGA.

To specify an External Check constraint, open the **Add External Check Constraint** dialog box in one of the following three ways:

- In the **Constraints Browser**, double-click **External Check**.
- From the **Constraints** menu, click **External Check**.
- In the **External Check Constraints Table**, right-click any row and click **Add External Check Constraint**. The **Add External Check Constraint** dialog box appears.

Figure 2-9. Add External Check Constraint Dialog Box



The following table describes the **Add External Check Constraint** dialog box options.

Table 2-5. Add External Check Constraint Options

Option	Description
Input Port	Specify the Input Port or click **Browse** next to **Input Port** to display the **Select Ports for External Check** dialog box. You can apply the External Check constraint on multiple input ports.
Clock Name	Specifies the clock reference to which the specified External Check is related.
Hold	Specifies the external hold time requirement in nanoseconds for the specified input ports.
Setup	Specifies the external setup time requirement in nanoseconds for the specified input ports.
Comment	Enter a one-line comment for this constraint.

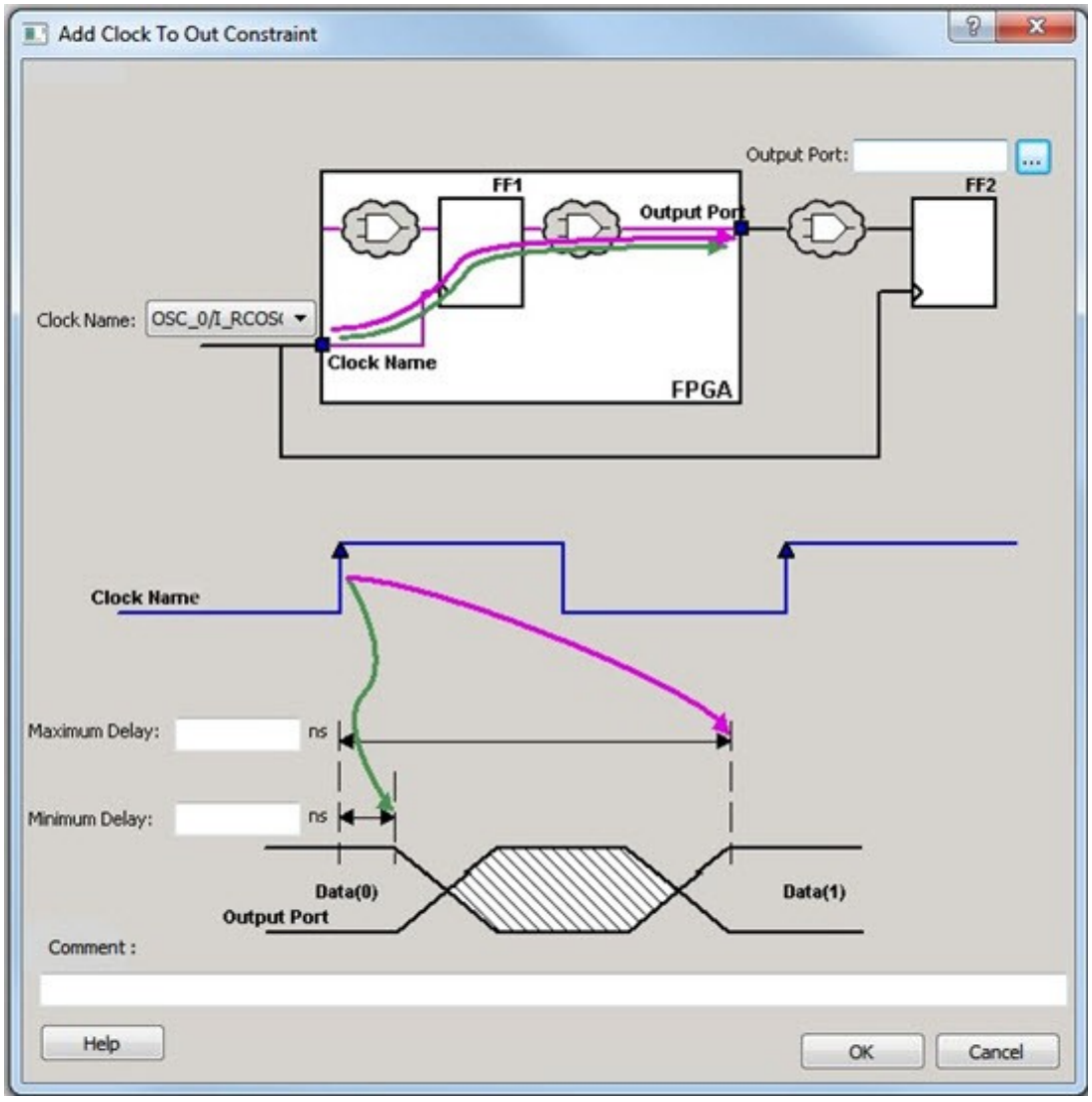
2.6 Set a Clock To Out Constraint (Ask a Question)

Enter a clock to output constraint by specifying the timing budget inside the FPGA.

To specify a Clock to Out constraint, open the **Add Clock to Out Constraint** dialog box in one of the following three ways:

- In the **Constraints Browser**, double-click **Clock to Out**.
- From the **Constraints** menu, click **Clock to Out** .
- Right-click any row of the **Clock To Out Constraints Table** and click **Add Clock to Out Constraint**. The **Add Clock To Out Constraint** dialog box appears.

Figure 2-10. Add Clock to Out Constraint Dialog Box



The following table describes the **Add Clock to Out Constraint** dialog box options.

Table 2-6. Add Clock to Out Constraint Options

Option	Description
Output Port	Specify the name of the output port or click **Browse** to display the **Select Ports for Clock to Output** dialog box. You can select multiple output ports to apply the Clock to Out constraint.
Clock Name	Specifies the clock reference to which the specified Clock to Out delay is related.
Maximum Delay	Specifies the delay in nanoseconds for the longest path from the specified output to the captured edge. This represents a combinational path delay to a register inside the current design plus the library setup time.
Minimum Delay	Specifies the delay in nanoseconds for the shortest path from the specified output to the captured edge. This represents a combinational path delay to a register inside the current design plus the library hold time.
Comment	Enter a one-line comment for this constraint.

3. Timing Exceptions (Ask a Question)

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

- Setting multicycle constraint to specify paths that (by design) 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/minimum delay constraint on specific paths to relax or to tighten the original clock constraint requirement.

3.1 Set a Maximum Delay Constraint (Ask a Question)

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

The Timing Constraints Editor 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.

Note: When the same timing path has more than one timing exception constraint, the **Timing Constraints Editor** honors the timing constraint with the highest precedence and ignores the other timing exceptions according to the order of precedence as listed in the following table.

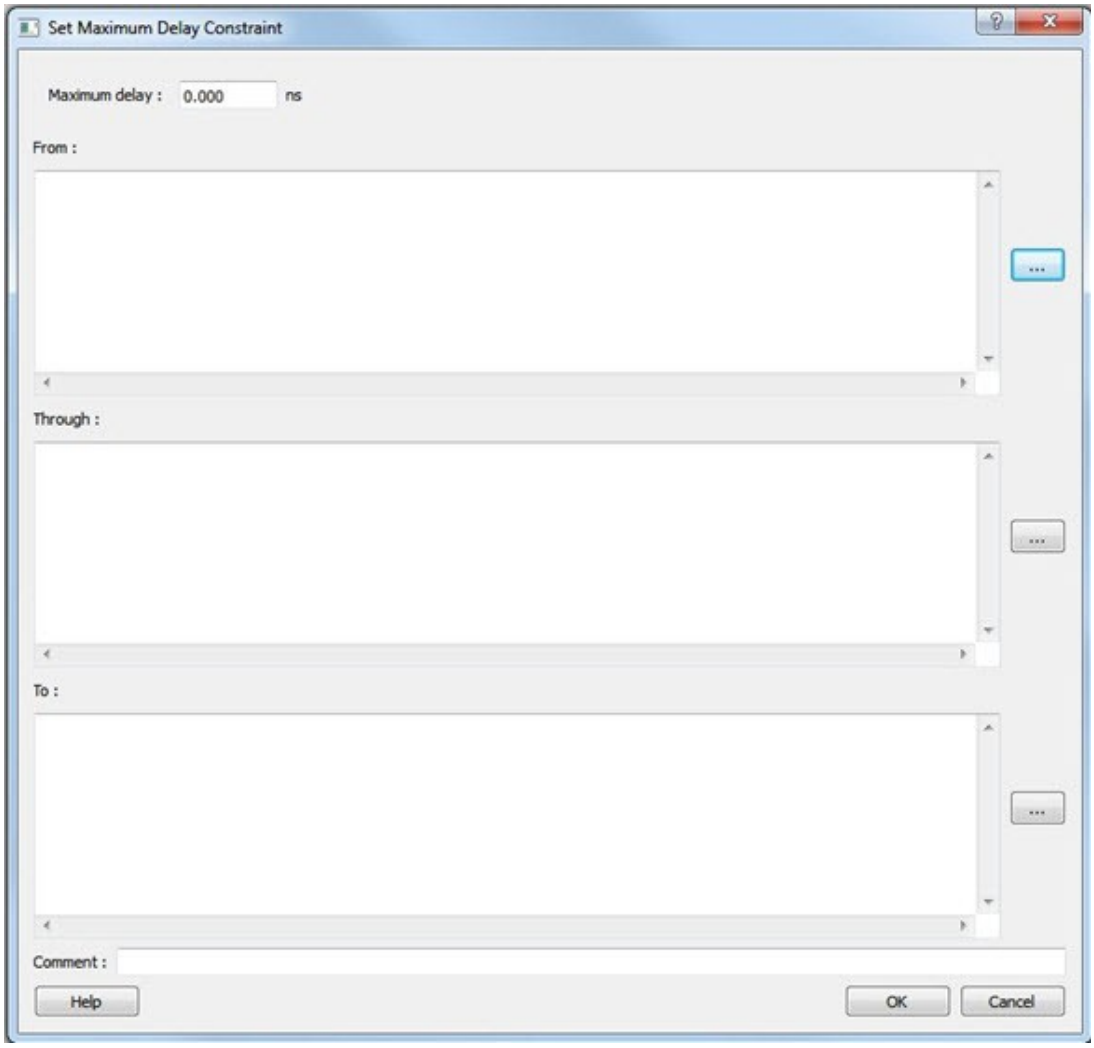
Timing Exception Constraints	Order of Precedence
set_disable_timing	1
set_false_path	2
set_maximum_delay/set_minimum_delay	3
set_multicycle_path	4

Note: The `set_maximum_delay_constraint` has a higher precedence over `set_multicycle_path` constraint and therefore the former overrides the latter when both constraints are set on the same timing path.

To set a Maximum Delay constraint, open the **Set Maximum Delay Constraint** dialog box in one of the following four ways:

- In the **Constraints Browser**, double-click **Max Delay**.
- Click the **Add Max Delay Constraint**  icon.
- From the **Constraints** menu, click **Max Delay** .
- In the **Max Delay Constraints** table, right-click any row and choose **Add Maximum Delay Constraint**. The **Set Maximum Delay Constraint** dialog box appears.

Figure 3-1. Set Maximum Delay Constraint Dialog Box



The following table describes the **Set Maximum Delay Constraint** dialog box options.

Table 3-1. Set Maximum Delay Constraint Options

Option	Description
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, the Timing Constraints Editor includes clock skew in the computed delay. • If the path starting point has an input delay specified, the Timing Constraints Editor adds that delay value to the path delay. • If the path ending point is on a sequential device, the Timing Constraints Editor includes clock skew and library setup time in the computed delay. • If the ending point has an output delay specified, the Timing Constraints Editor adds that delay to the path delay.

.....continued

Option	Description
Source/From Pins	Specifies the starting points for max delay constraint path. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell. To specify the Source pins(s), click the **Browse** next to **From** to open the **Select Source Pins for Max Delay Constraint** dialog box. The following options are available on the **Select Source Pins for Max Delay Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are listed as follows: – Clock Pins – Input Ports – All Register Clock Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list displays the available pins or ports. If you change the pattern value, the list shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the source pins to the constraint, in this list, click the pins and click **OK**.
Through Pins	Specifies the through pins in the specified path for the Maximum Delay constraint. To specify the Through pin(s), click **Browse** next to **Through** textbox to open the **Select Through Pins for Max Delay Constraint** dialog box. The following options are available on the **Select Through Pins for Max Delay Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The **Type** options for source pins are listed as follows: – All Ports – All Pins – All Nets – All Instances • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list displays the available pins, ports, nets, or instances. If you change the pattern value, the list box shows the available pins, ports, nets, or instances based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the through pins to the constraint, in this list, click the Pins and click **OK**.

.....continued	
Option	**Description**
Destination/To Pins	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. To specify the destination pin(s), click **Browse** next to the **To** box to open the **Select Destination Pins for Max Delay Constraint** dialog box. The following options are available on the **Select Destination Pins for Max Delay Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The **Type** options for source pins are listed as follows: – Clock Pins – Output Ports – All Register Data Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the destination pins to the constraint, click pins from this list and click **OK**.
Comment	Enter a one-line comment for the constraint.

3.2 Set a Minimum Delay Constraint (Ask a Question)

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

The **Timing Constraints Editor** 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.

Note: When the same timing path has more than one timing exception constraint, the **Timing Constraints Editor** honors the timing constraint with the highest precedence and ignores the other timing exceptions according to the order of precedence as listed in the following table.

Table 3-2. Timing Exceptions

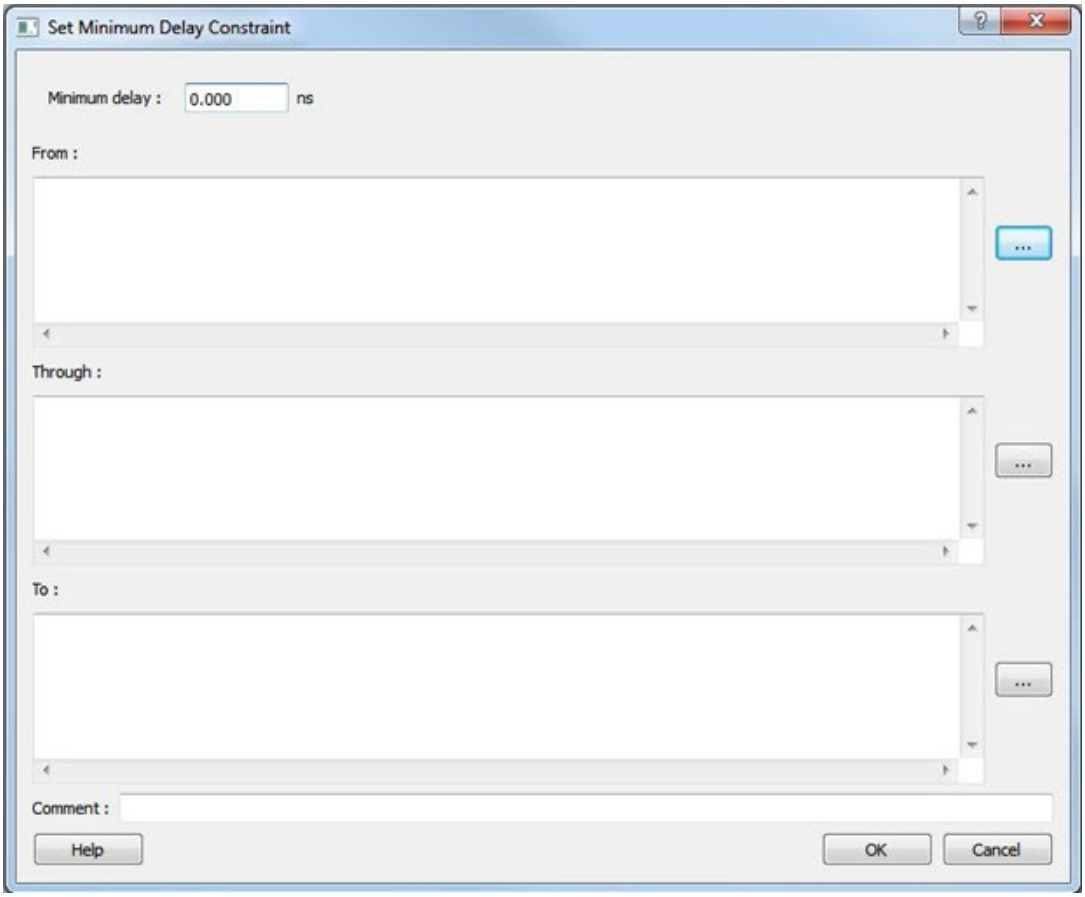
Timing Exception Constraints	Order of Precedence
`set_disable_timing`	1
`set_false_path`	2
`set_maximum_delay`/`set_minimum_delay`	3
`set_multicycle_path`	4

Note: : The `set_minimum_delay_constraint` has a higher precedence over `set_multicycle_path` constraint and therefore the former overrides the latter when both constraints are set on the same timing path.

To set a Minimum Delay constraint, open the **Set Minimum Delay Constraint** dialog box in one of the following four ways:

- In the Constraints Browser, click **Min Delay**.
- Click the **Add Min Delay Constraint**  icon.
- From the **Constraints** menu, click **Min Delay**.
- Right-click any row of **Min Delay Constraints Table** and click **Add Minimum Delay Constraint**. The **Set Minimum Delay Constraint** dialog box appears.

Figure 3-2. Set Minimum Delay Constraint Dialog Box



The following table describes the **Set Minimum Delay Constraint** dialog box options.

Table 3-3. Set Minimum Delay Constraint Options

Option	Description
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, the **Timing Constraints Editor** includes clock skew in the computed delay. • If the path starting point has an input delay specified, the **Timing Constraints Editor** adds that delay value to the path delay. • If the path ending point is on a sequential device, the **Timing Constraints Editor** includes clock skew and library setup time in the computed delay. • If the ending point has an output delay specified, the **Timing Constraints Editor** adds that delay to the path delay.

.....continued

Option	Description
Source Pins/ From	Specifies the starting point 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. To specify the source pins(s), click **Browse** next to the **From** box to open the **Select Source Pins for Minimum Delay Constraint** dialog box. The following options are available on the **Select Source Pins for Minimum Delay Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The **Type** options for source pins are: – Clock Pins – Input Ports – All Register Clock Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the source pins to the constraint, click pins from this list and click **OK**.
Through Pins	Specifies the through points for the Minimum Delay constraint. To specify the Through pin(s), click **Browse** next to the **Through** box to open the **Select the Through Pins for Min Delay** dialog box. The following options are available on the **Select the Through Pins for Min Delay** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are: – All Ports – All Pins – All Nets – All Instances • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins, ports, nets, or instances. If you change the pattern value, the list box shows the available pins, ports, nets, or instances based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the Available Pins list. To add the through pins to the constraint, click the pins from this list and click **OK**.

.....continued	
Option	**Description**
Destination/To Pins	Specifies the ending points for minimum delay constraint. A valid timing ending point is a clock, a primary output, or a data pin of a sequential cell. To specify the Destination pin(s), click **Browse** next to the **To** box to open the **Select the Destination Pins for Min Delay Constraint** dialog box. The following options are available on the **Select the Destination Pins for Min Delay Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The **Type** options for source pins are: – Clock Pins – Output Ports – All Register Data Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the destination pins to the constraint, click the pins from this list and click **OK**.
Comment	Enter a one-line comment for the Constraint.

3.3 Set a Multicycle Path (Ask a Question)

Use this constraint to identify paths in the design that take multiple clock cycles.

You can set multicycle path constraints in an SDC file, which you can either create yourself or generate with Synthesis tools, and at the same time you can import the netlist.

You can use one or more of the following to set the multicycle paths constraints:

- set_multicycle_path
- 3.4.1. Specifying a Multicycle Constraint

Families Supported

The following table lists the families which support this constraint and the file formats and tools which you can use to enter or modify it:

Table 3-4. Families Supported

Families	SDC	Constraints Editor
PolarFire®	X	X
RTG4™	X	X
IGLOO® 2	X	X
SmartFusion® 2	X	X

3.4 Set a Multicycle Constraint (Ask a Question)

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

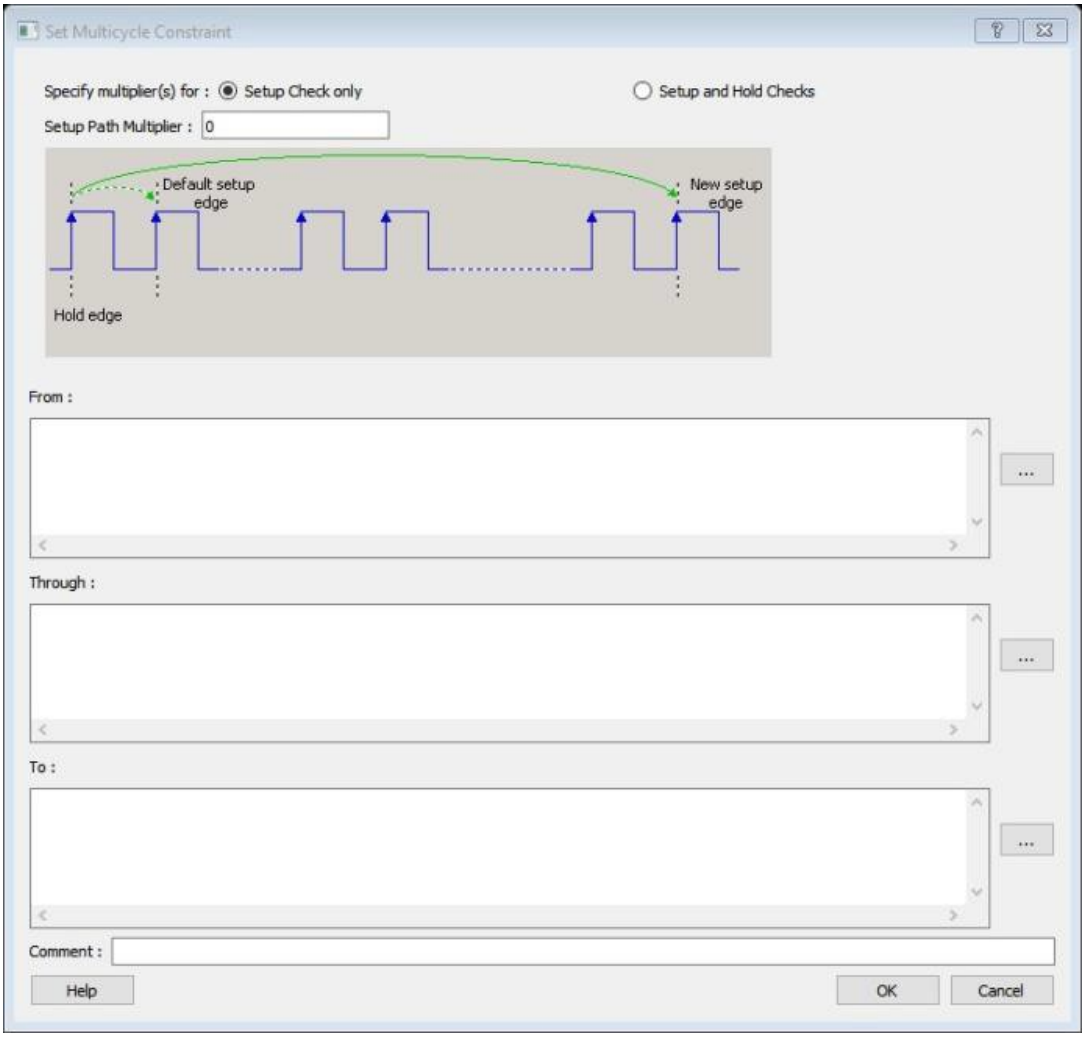
Setting the multiple-cycle path constraint overrides the single-cycle timing relationships (default) between sequential elements by specifying the number of cycles (two or more) that the data path must have for setup or hold checks.

Note: The false path information always takes precedence over multiple cycle path information. A specific maximum delay constraint overrides a general multiple cycle path constraint.

To set a multicycle constraint, open the **Set Multicycle Constraint** dialog box in one of the following four ways:

- In the **Constraints Browser**, double-click **Multicycle**.
- Click the **Add Multicycle Constraint** icon.
- From the **Constraints** menu, click **Multicycle**.
- Right-click any row of **Multicycle Constraints Table** and click **Add Multicycle Path Constraint**. The **Set Multicycle Constraint** dialog box appears.

Figure 3-3. Set Multicycle Constraint Dialog Box



The following table describes the **Set Multicycle Constraint** dialog box options.

Table 3-5. Set Multicycle Constraint Options

Option	Description
Setup Check Only	Select this check box to apply multiple clock cycle timing consideration for setup check only.
Setup and Hold Checks	Select this check box to apply multiple clock cycle timing consideration for both setup and hold checks.
Hold Path Multiplier	Specifies an integer value that represents the number by which the edge will move towards the left, or the edge number, where the check will be performed.
Setup Path Multiplier	Specifies an integer value that represents the number of clock cycles (more than one) the data path must have for a setup check.
Source Pins/ From	Specifies the starting points for the multiple cycle path. A valid starting point is a clock, a primary input, an inout port, or the clock pin of a sequential cell. To specify the source pins(s), click **Browse** next to the **From** box to open the **Multicycle Constraint** dialog box. The following options are available on the **Multicycle Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are: – Clock Pins – Input Ports – All Register Clock Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected in the **Available Pins** list. To add the source pins to the constraint, click pins in this list and click **OK**.

.....continued

Option	Description
Through Pins	Specifies the through points for the multiple cycle path. To specify the through pin(s), click **Browse** next to the **Through** box to open the **Select Through Pins for Multicycle Constraint** dialog box. The following options are available on the **Select Through Pins for Multicycle Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are: – All Ports – All Pins – All Nets – All Instances • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins, ports, nets. or instances. If you change the pattern value, the list box shows the available pins, ports, nets, or instances based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the Available Pins list. To add the through pins to the constraint, click the pins from this list and click **OK**.
Destination/To Pins	Specifies the ending points for multiple cycle path. To specify the destination pin(s), click **Browse** next to the **To** box to open the **Select Destination Pins** dialog box. The following options are available on the **Select Destination Pins** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are: – Clock Pins – Output Ports – All Register Data Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected in the **Available Pins** list. To add the destination pins to the constraint, click pins in this list and click **OK**.
Comment	Enter a one-line comment for the constraint.

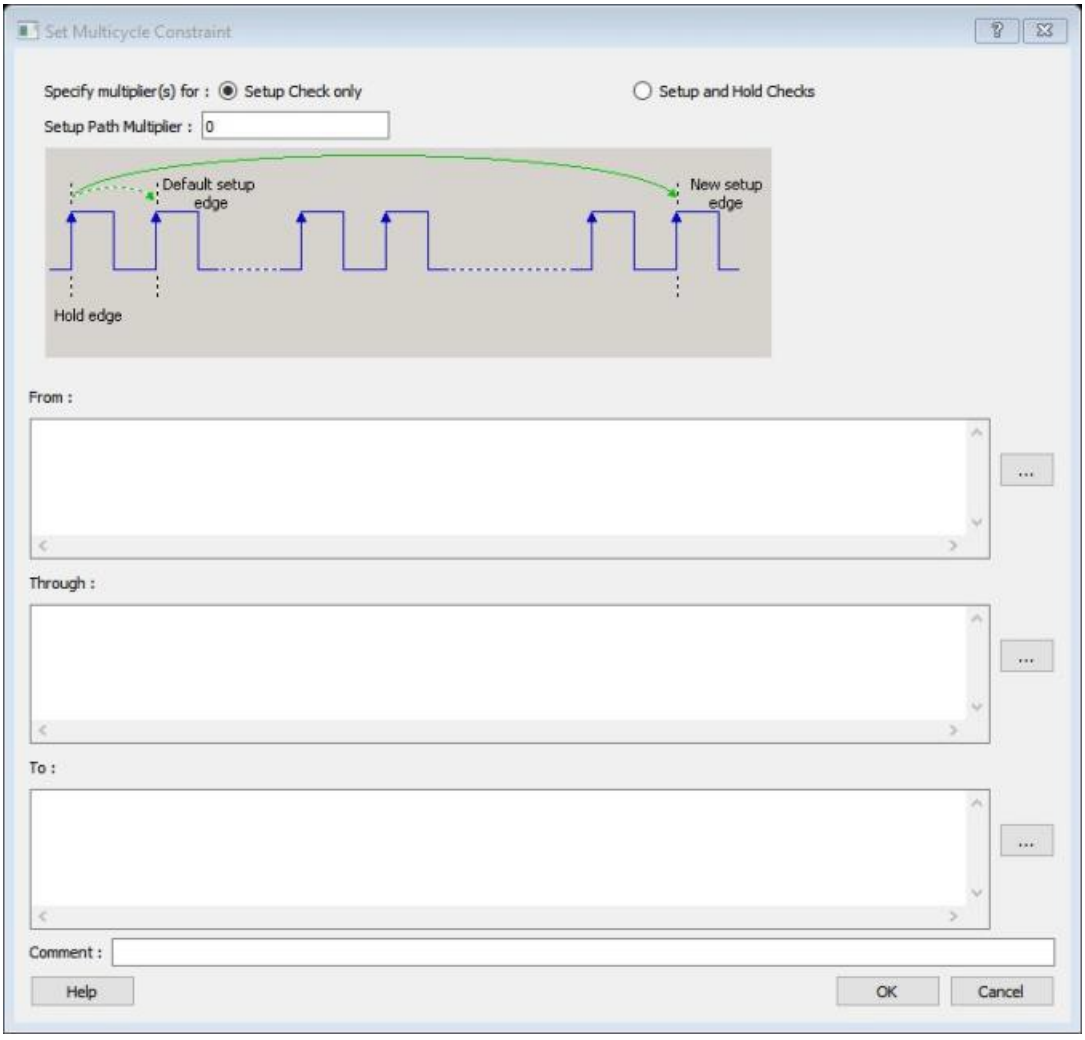
3.4.1 Specifying a Multicycle Constraint (Ask a Question)

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:

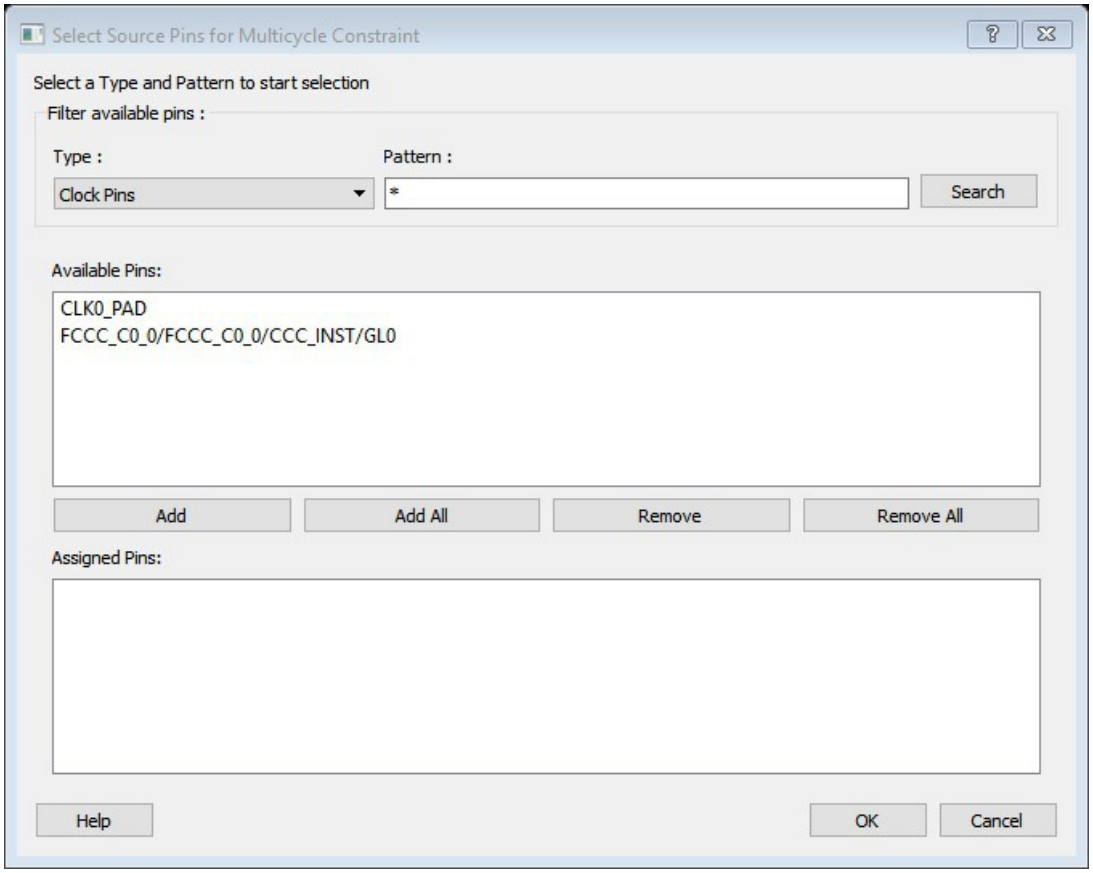
1. Add the constraint in the **Editable Constraints Grid** or open the **Set Multicycle Constraint** dialog box using one of the following methods:
   - From the **Constraints** menu, click **MultiCycle**.
   - Click the  icon.
   - In the **Constraints Browser**, double-click **Multicycle**.
   - Right-click the **Multicycle** option in the **Constraint Browser** and click **Add Multicycle Path Constraint**. The **Set Multicycle Constraint** dialog box appears.

   Figure 3-4. Set Multicycle Constraint Dialog Box



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

Figure 3-5. Select Source Pins for Multicycle Constraint Dialog Box



4. Use **Filter available pins** to narrow the pin list based on the selected **Type** and **Pattern**. In the **Available Pins** list, click the pin(s). You can select multiple pins in this window.
5. Click **Add** or **Add All** to add the pins from the **Available Pins** list to the **Assigned Pins** list. Click **Remove** or **Remove All** to remove the pins from the **Assigned Pins** list.
6. Click the pins from the **Assigned Pins** list and click **OK**. The **Set Multicycle Constraint** dialog box displays the updated **From** pin(s) list.
7. 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
8. Enter comments in the **Comment** section.
9. Click **OK**. The **Timing Constraints Editor** adds the multicycle constraints to the **Constraints List**.

3.5 Set a False Path (Ask a Question)

Use this constraint to identify paths in the design that should be disregarded during timing analysis and timing optimization.

By definition, false paths are paths that cannot be sensitized under any input vector pair. Therefore, including false paths in timing calculation may lead to unrealistic results. For accurate static timing analysis, it is important to identify the false paths.

You can set false paths constraints in an SDC file, which you can either create yourself or generate with Synthesis tools, at the same time you import the netlist.

You can use one or more of the following commands or GUI tools to set false paths:

- set_false_path
- Specifying False Path Constraint

Families Supported
The following table lists the families that support this constraint, the file formats, and tools you can use to enter or modify it.

Table 3-6. Families Supported

Families	SDC	Constraints Editor
PolarFire®	X	X
RTG4™	X	X
IGLOO® 2	X	X
SmartFusion® 2	X	X

3.6 Set a False Path Constraint (Ask a Question)

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

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: When the same timing path has more than one timing exception constraint, the Timing Constraints Editor honors the timing constraint with the highest precedence and ignores the other timing exceptions according to the order of precedence as listed in the following table.

Table 3-7. Timing Exception Constraints

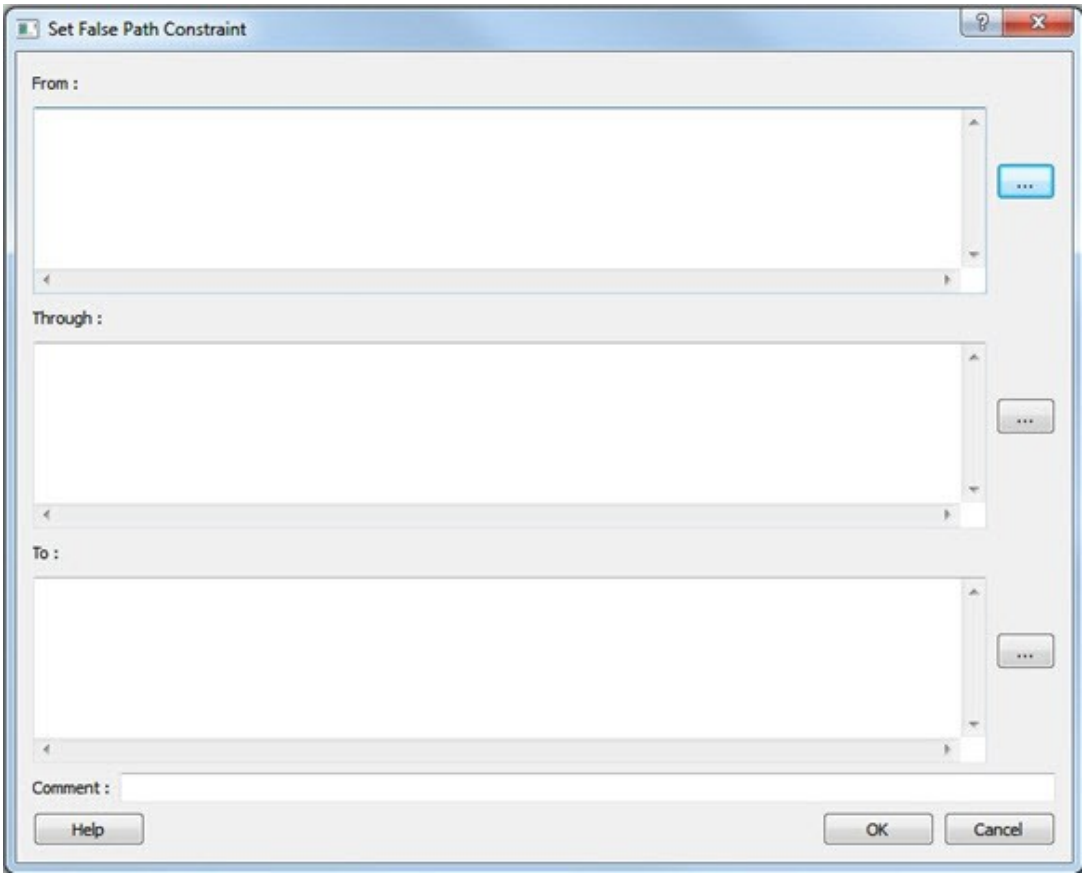
Timing Exception Constraints	Order of Precedence
set_disable_timing	1
set_false_path	2
set_maximum_delay/set_minimum_delay	3
set_multicycle_path	4

Note: The `set_false_path` constraint has the second highest precedence and always overrides the `set_multicycle_path` constraints and `set_maximum/minimum_delay` constraints.

To set a false path constraint, open the **Set False Path Constraint** dialog box in one of the following four ways:

- From the Constraints Browser, double-click **False Path**.
- Click the **Add False Path Constraint**  icon.
- From the **Constraints** menu, click **False Path** .
- Right-click any row of **False Path Constraints Table** and click **Add False Path Constraint**. The **Set False Path Constraint** dialog box appears.

Figure 3-6. Set False Path Constraint Dialog Box



The following table describes the **Set False Path Constraint** dialog box options.

Table 3-8. Set False Path Constraint Options

Option	Description
Source/From Pins	Specifies the starting point for false path constraint. To specify the source pins(s), click **Browse** next to the **From** box to open the **Select Source Pins for False Path Constraint** dialog box. The following options are available on the **Select Source Pins for False Path Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for Source Pins are listed as follows: – Clock Pins – Input Ports – All Register Clock Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the **Available Pins** list. To add the source pins, click the pins from this list and click **OK**.
Through Pins	Specifies the through points for the false path constraint. To specify the through pin(s), click on the **Browse** button next to **Through** textbox to open the **Select the Through Pins for False Path Constraint** dialog box. The following options are available on the **Select the Through Pins for False Path Constraint** dialog box: • **Type**: Displays the Type of the Available Pins in the design. The type options for source pins are listed as follows: – All Ports – All Pins – All Nets – All Instances • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins, ports, nets, or instances. If you change the pattern value, the list box shows the available pins, ports, nets, or instances based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the through pins to the constraint.

.....continued	
Option	**Description**
Destination/To Pins	Specifies the ending points for false path constraint. To specify the destination pin(s), click **Browse** next to the **To** box to open the **Select the Destination Pins for False Path Constraint** dialog box. The following options are available on the **Select the Destination Pins for False Path Constraint** dialog box: • **Type**: Displays the type of the available pins in the design. The type options for source pins are listed as follows: – Clock Pins – Output Ports – All Register Data Pins • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified pin type and pattern. • **Available Pins**: The list box displays the available pins or ports. If you change the pattern value, the list box shows the available pins or ports based on the filter. To add the pins from the **Available Pins** list to the **Assigned Pins** list, click **Add** or **Add All**. To remove the pins from the **Assigned Pins** list, click **Remove** or **Remove All**. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the destination pins to the constraint.
Comment	Enter a one-line comment for the Constraint.

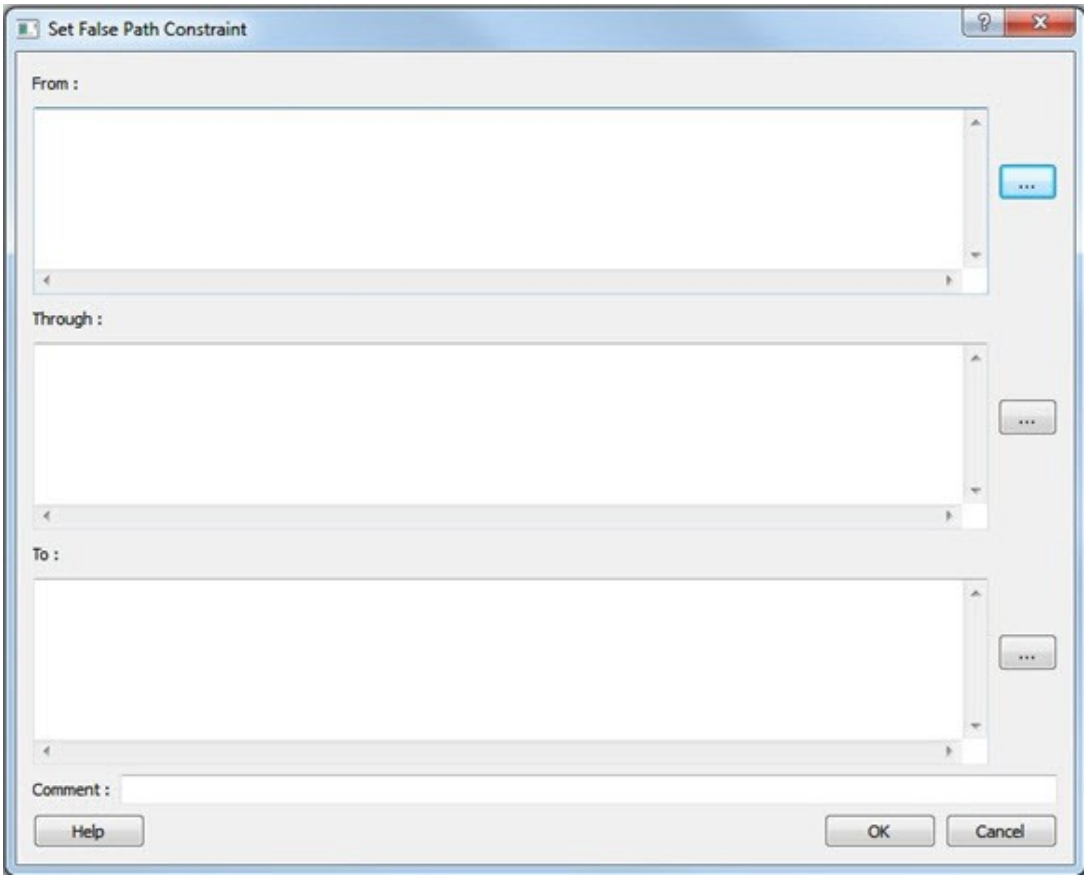
3.6.1 Specifying a False Path Constraint (Ask a Question)

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

To specify False Path constraints:

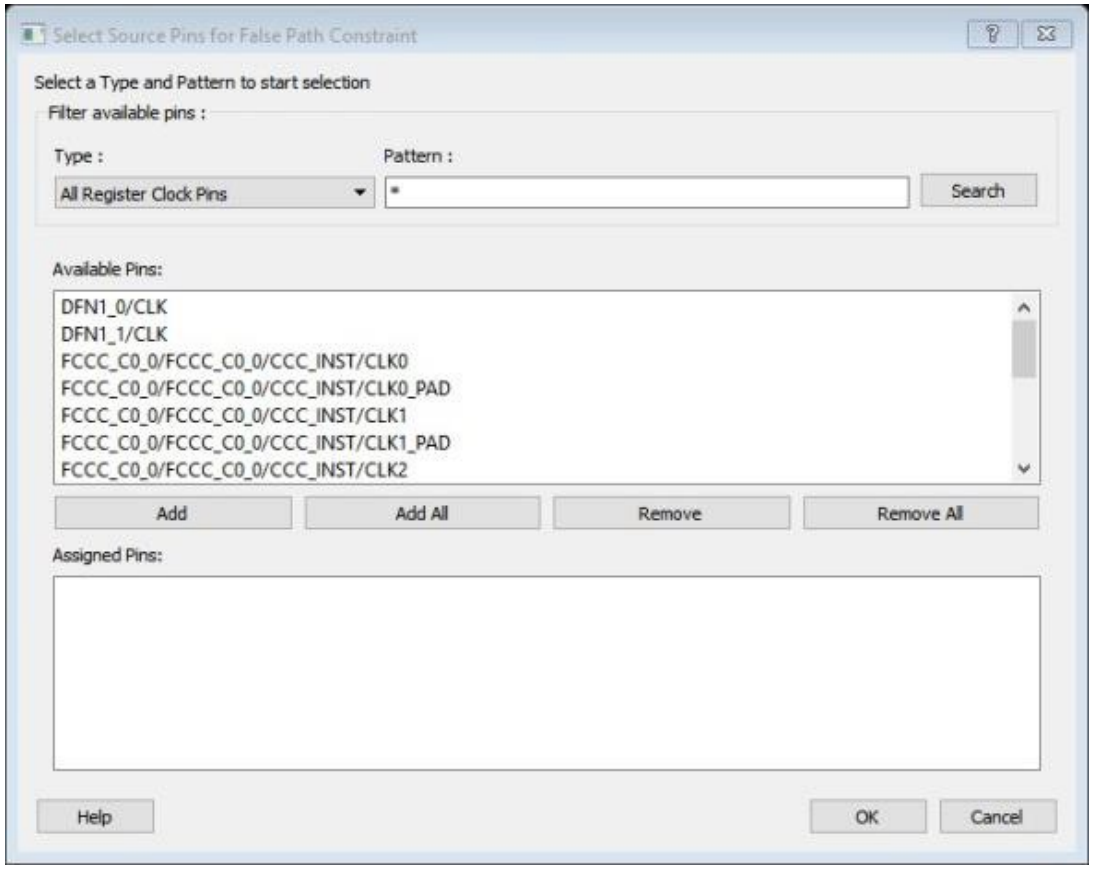
1. Add the constraint in the Editable Constraints Grid or open the Set False Path Constraint dialog box. You can do this by using one of the following methods:
   – From the **Constraints** menu, choose **False Path**.
   – Click the  icon.
   – From the Constraints Browser, double-click **False Path**.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

Figure 3-7. 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.

Figure 3-8. Select Source Pins for False Path Constraint Dialog Box



3. Use **Filter available pins** to narrow the pin list based on the selected **Type** and **Pattern**. Select the pin(s) from the **Available Pins** list. You can select multiple pins in this window.
4. Click **Add** or **Add All** to add the pins from the **Available Pins** list to the **Assigned Pins** list. Click **Remove** or **Remove All** to remove the pins from the **Assigned Pins** list**.**
5. Select the pins from the **Assigned Pins** list and click **OK**. The **Set False Path Constraint** dialog box displays the updated **From** pin(s) list.
6. 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.
7. Enter comments in the **Comment** section.
8. Click **OK**.
   The False Path constraints are added to the **Constraints List** in the **Timing Constraints Editor**.

4. Adjusting Clock Information (Ask a Question)

After you define all required constraints, including clock information, enter any information related to clock noise when such information is not or cannot be generated automatically by the tools. This is generally the case for external clock jitter information occurring at the primary inputs of the device. There can be multiple sources of clock jitter including:

- Global jitter (or internal system jitter)
- PLL/DLL jitter
- External jitter

For more information about these types of jitter, see the appropriate product datasheets and documents.

Starting with Libero SoC v2022.1, the automatic computation of system clock jitter for Place & Route and Static Timing Analysis was introduced to the design flow for PolarFire and PolarFire SoC devices. With this feature, you need only specify the clock jitter of any FPGA primary input using the `set_input_jitter` constraint. These jitter components are analyzed automatically, and a specific jitter value computed for each clock is passed on to the appropriate design flow steps. With the automatic clock jitter flow, you can add a margin to the estimated jitter of a clock using the `set_clock_uncertainty` constraint.

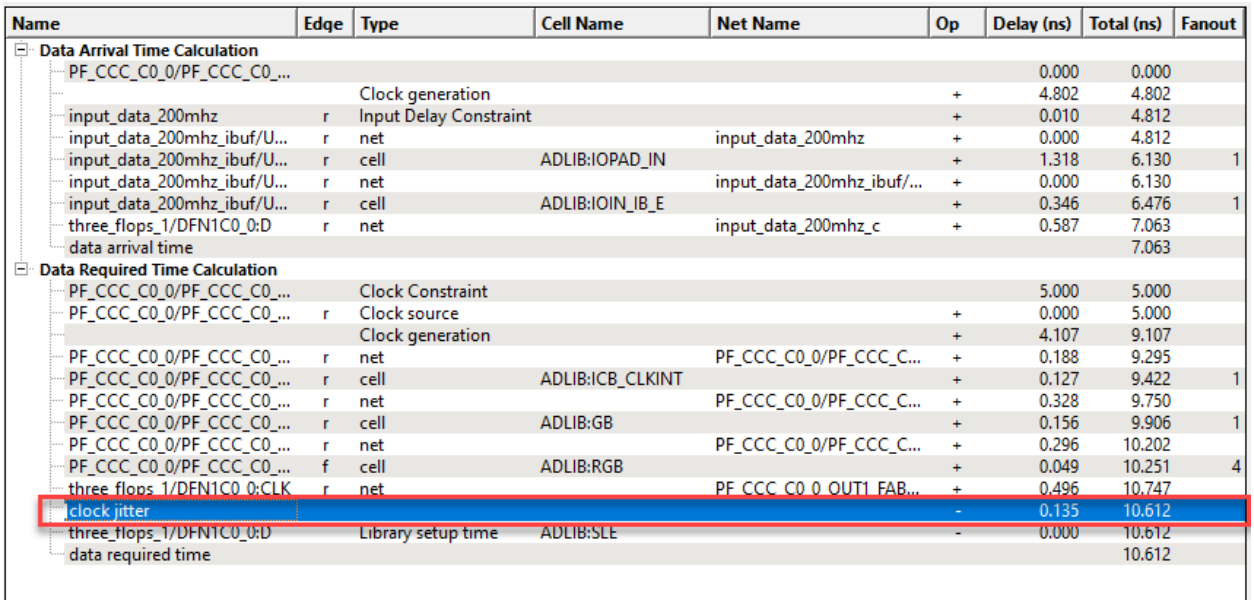
In rare cases or when instructed by user guides, you can use the `set_system_jitter` constraint to override the automatically computed global jitter value. System jitter is estimated based on the count of registers and RAM blocks in the synthesized netlist. A report similar to the following, detailing the jitter estimate, is generated upon completing the Place & Route and Verify Timing steps.

Figure 4-1. Sample Report Detailing Jitter Estimation

```
Jitter Estimation Report
========================

Design     : fft_points1024_bit18_buffer0
Family     : PolarFire
Die        : MPF500T
Speed grade: STD


System Jitter Calculation
----------------------

Worst aggressor based on load:    fft_points1024_bit18_buffer0|CLK
System jitter (worst aggressor):  0.001 ns


Jitter Calculation per Clock Domain
-----------------------------

Clock: slowClock|divider_inferred_clock[2]

    System jitter (worst aggressor):          0.001 ns
    Resulting clock jitter:                   0.001 ns


Clock: fft_points1024_bit18_buffer0|CLK

 (1) System jitter (worst aggressor):         0.001 ns
 (2) Input jitter:                            0.000 ns

    Resulting clock jitter (max of (1) and (2)):  0.001 ns
```

The expanded timing analysis path reports include the clock jitter as an extra line item.

Figure 4-2. Clock Jitter Shown as an Extra Line Item



Name	Edge	Type	Cell Name	Net Name	Op	Delay (ns)	Total (ns)	Fanout
Data Arrival Time Calculation								
PF_CCC_C0_0/PF_CCC_C0_...						0.000	0.000	
		Clock generation			+	4.802	4.802	
input_data_200mhz	r	Input Delay Constraint			+	0.010	4.812	
input_data_200mhz_ibuf/U...	r	net		input_data_200mhz	+	0.000	4.812	
input_data_200mhz_ibuf/U...	r	cell	ADLIB:IOPAD_IN		+	1.318	6.130	1
input_data_200mhz_ibuf/U...	r	net		input_data_200mhz_ibuf/...	+	0.000	6.130	
input_data_200mhz_ibuf/U...	r	cell	ADLIB:IOIN_IB_E		+	0.346	6.476	1
three_flops_1/DFN1C0_0:D	r	net		input_data_200mhz_c	+	0.587	7.063	
data arrival time							7.063	
Data Required Time Calculation								
PF_CCC_C0_0/PF_CCC_C0_...		Clock Constraint				5.000	5.000	
PF_CCC_C0_0/PF_CCC_C0_...	r	Clock source			+	0.000	5.000	
		Clock generation			+	4.107	9.107	
PF_CCC_C0_0/PF_CCC_C0_...	r	net		PF_CCC_C0_0/PF_CCC_C...	+	0.188	9.295	
PF_CCC_C0_0/PF_CCC_C0_...	r	cell	ADLIB:ICB_CLKINT		+	0.127	9.422	1
PF_CCC_C0_0/PF_CCC_C0_...	r	net		PF_CCC_C0_0/PF_CCC_C...	+	0.328	9.750	
PF_CCC_C0_0/PF_CCC_C0_...	r	cell	ADLIB:GB		+	0.156	9.906	1
PF_CCC_C0_0/PF_CCC_C0_...	r	net		PF_CCC_C0_0/PF_CCC_C...	+	0.296	10.202	
PF_CCC_C0_0/PF_CCC_C0_...	f	cell	ADLIB:RGB		+	0.049	10.251	4
three_flops_1/DFN1C0_0:CLK	r	net		PF_CCC_C0_0_OUT1_FAB...	+	0.496	10.747	
clock jitter					-	0.135	10.612	
three_flops_1/DFN1C0_0:D		Library setup time	ADLIB:SLE		-	0.000	10.612	
data required time							10.612	

You can also provide the latency of a clock from its source outside the device to the primary input using the `set_clock_latency` constraint.

For more information about clock jitters, see the PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide.

4.1 Set a Clock Input Jitter Value (Ask a Question)

Use the `set_input_jitter` constraint to specify the peak-to-peak jitter value at the primary input of the clock (see section 6.61. set_input_jitter).

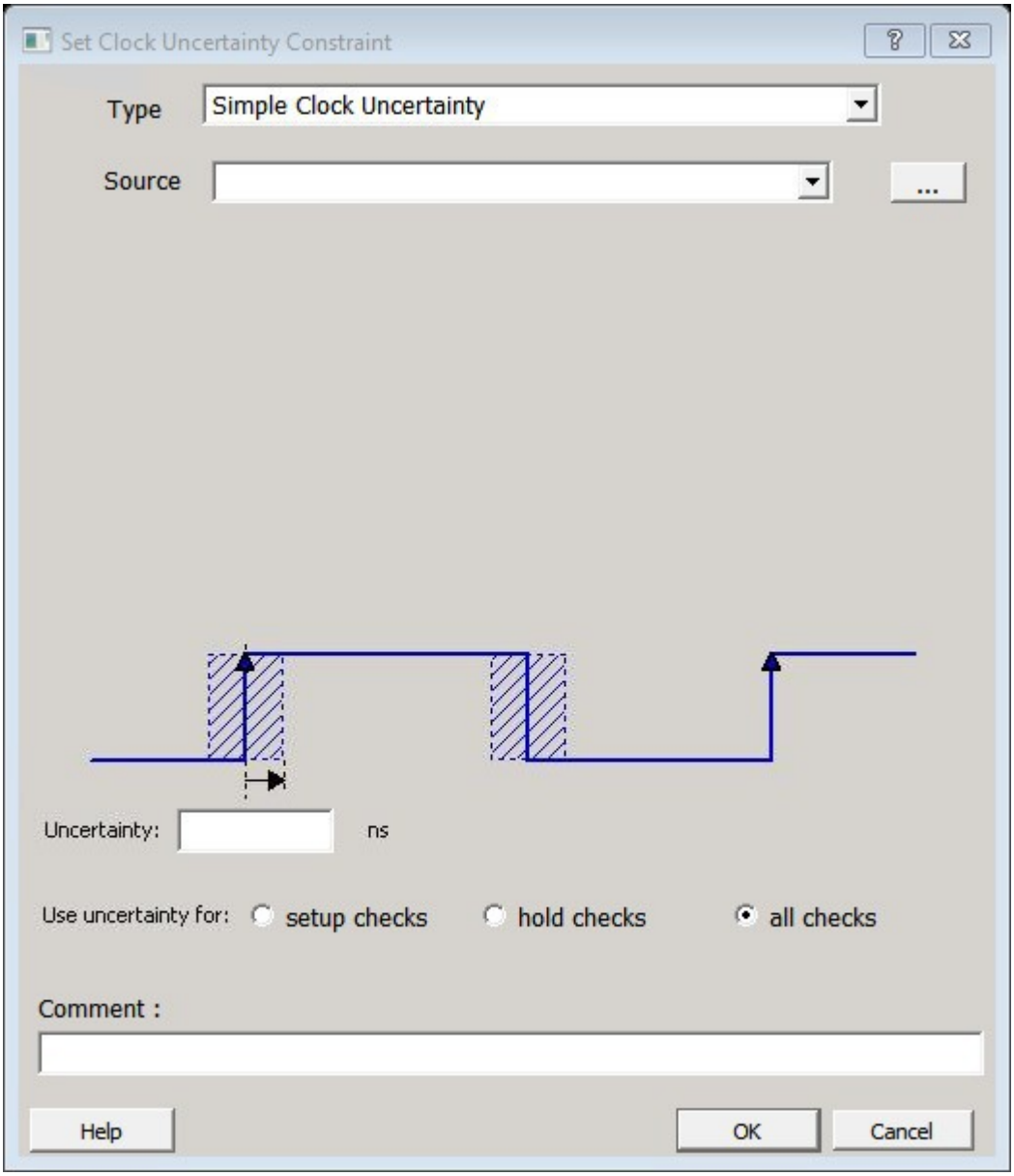
4.2 Set a Clock Uncertainty Constraint (Ask a Question)

Use the **Set Clock Uncertainty Constraint** dialog box to set either the Simple Clock Uncertainty constraint or the Clock To Clock Uncertainty constraint from the **Type** drop-down menu. The default is Simple Clock Uncertainty.

4.2.1 Set Simple Clock Uncertainty Constraint (Ask a Question)

To open the **Set Clock Uncertainty Constraint** dialog box from the **Constraints** menu, choose **Clock Uncertainty** and select **Simple Clock Uncertainty** from the **Type** drop-down menu in **Set Clock Uncertainty Constraint** dialog box.

Figure 4-3. Set Clock Uncertainty Constraint dialog box



The following table describes the **Set Clock Uncertainty Constraint** dialog box options.

Table 4-1. Set Clock Uncertainty Constraint Options

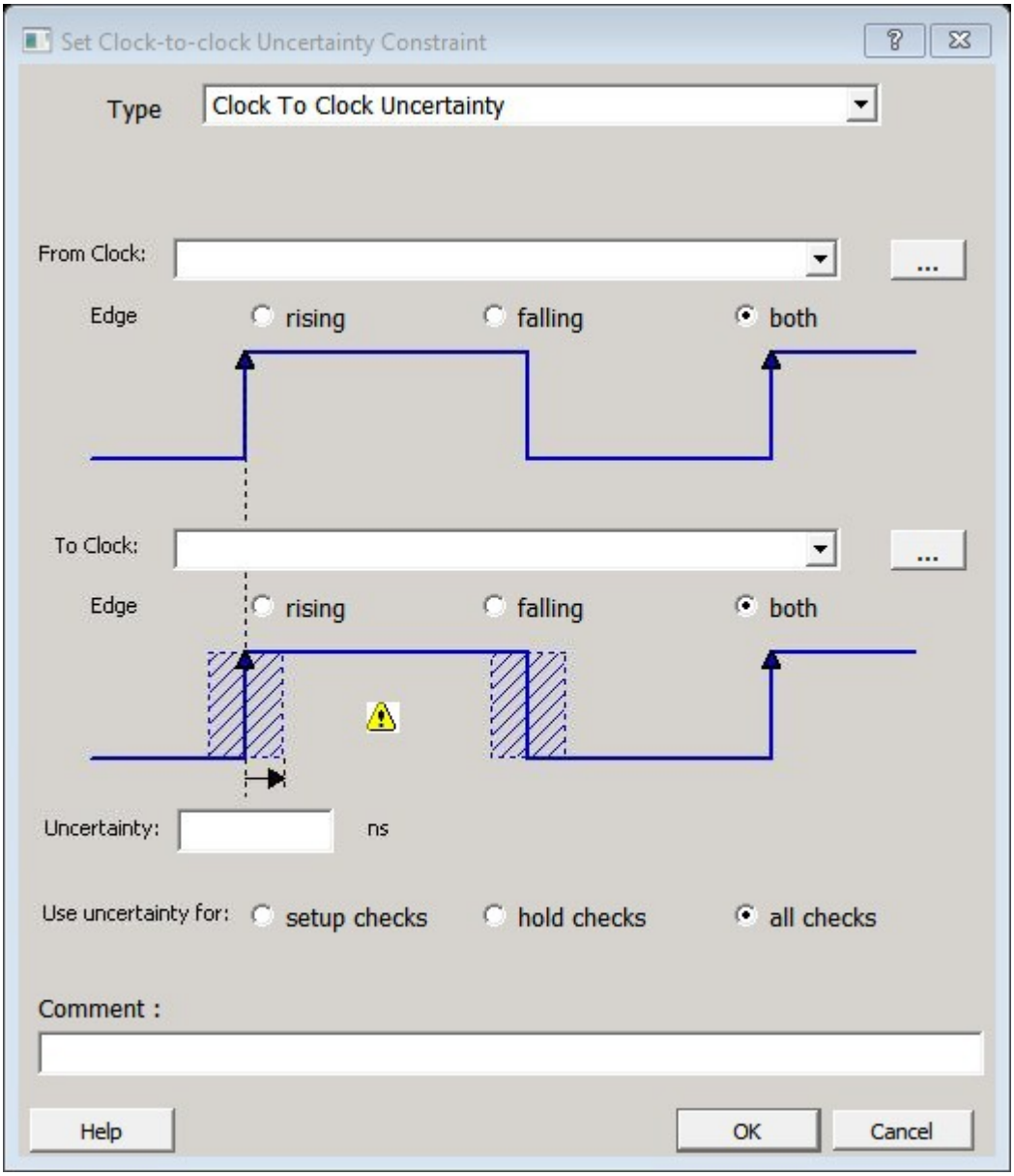
Option	Description
Type	Select **Simple Clock Uncertainty** (default) to set clock uncertainty constraint on a single clock.

.....continued	
Option	**Description**
Source	Specifies the clock name as the uncertainty source. To set the source clock, click the **Browse** button to open the **Select Source for Simple Uncertainty Constraint** dialog box. The following options are available in the **Select Source for Simple Uncertainty Constraint** dialog box: • **Type**: Displays the type of available pins in the design. The available **Pin Type** options are: – Clock Pins – All Pins – All Ports • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins or ports (ports are defined as clock signal in the design). If you change the pattern value, the list box displays the available pins or ports based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the source to the constraint.
Uncertainity	Enter the time in `ns` that represents the amount of deviation between two clock edges.
Use uncertainty for	Enables you to 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.

4.2.2 Set Clock-to-Clock Uncertainty Constraint (Ask a Question)

To open the **Set Clock-to-clock Uncertainty Constraint** dialog box from the **Constraints** menu, choose **Clock Uncertainty**, and select **Clock To Clock Uncertainty** from the **Type** drop-down menu.

Figure 4-4. Set Clock-to-Clock Uncertainty Constraint Dialog Box



The following table describes the **Set Clock-to-clock Uncertainty Constraint** dialog box options.

Table 4-2. Set Clock-to-Clock Uncertainty Constraint Options

Option	Description
Type	Select **Clock To Clock Uncertainty**.

.....continued	
Option	**Description**
From Clock	Specifies clock name as the uncertainty source. To set the From Clock, click the **Browse** button to open the **Select Source Clock List for Clock-to-clock Uncertainty** dialog box. The following options are available in the **Select Source Clock List for Clock-to-clock Uncertainty** dialog box: • **Type**: Displays the type of available pins in the design. The only choice available for Pin Type is **Clock Pins**. • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins. If you change the pattern value, the list box displays the available pins based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the source clock for the Clock-to-Clock Uncertainty constraint.
Edge	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. To set the To Clock, click the **Browse** button to open the **Select Destination Clock List for Clock-to-clock Uncertainty Constraint** dialog box. The following options are available in the **Select Destination Clock List for Clock-to-clock Uncertainty Constraint** dialog box: • **Type**: Displays the type of available pins in the design. The only valid selection is Clock Pins. • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins. If you change the pattern value, the list box displays the available pins based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the destination clock for the Clock-to-Clock Uncertainty constraint.
Uncertainty	Enter the time in `ns` that represents the amount of deviation between two clock edges.
Use Uncertainty For	Enables you to 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.

4.3 Set Clock Source Latency Constraint (Ask a Question)

Use the clock source latency constraint 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.

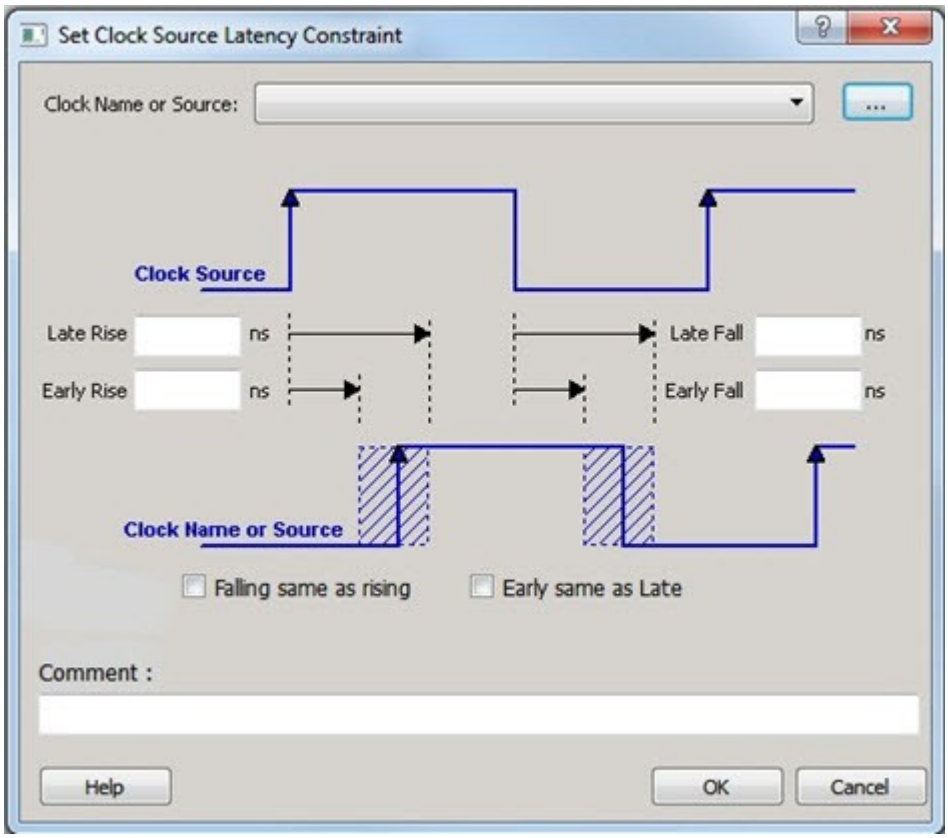
You can specify both an "early" delay and a "late" delay for this latency, providing an uncertainty that the timing analyzer can use for propagating through its calculations. Rising and falling edges of the same clock can have different latencies. If only one clock source latency value is provided, the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

To specify a Clock Source Latency constraint, open the **Set Clock Source Latency Constraint** dialog box in one of the following four ways:

- In the **Constraints Browser**, double-click **Clock Source Latency**.

- Click the **Clock Source Latency Constraint** icon.
- From the **Constraints** menu, click **Clock Source Latency** .
- Right-click any row of **Clock Latency Constraints Table** and click **Add Clock Source Latency**. The **Set Clock Source Latency Constraint** dialog box appears.

Figure 4-5. Set Clock Source Latency Constraint Dialog Box



The following table describes the **Set Clock Source Latency Constraint** dialog box options.

Table 4-3. Set Clock Source Latency Constraint Options

Option	Description
Clock Name or Source	To select the clock source, click the **Browse** button to open the **Choose the Clock Source Pin** dialog box. The only choice available for Pin Type is **Clock Pins**.
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.
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 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.
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.

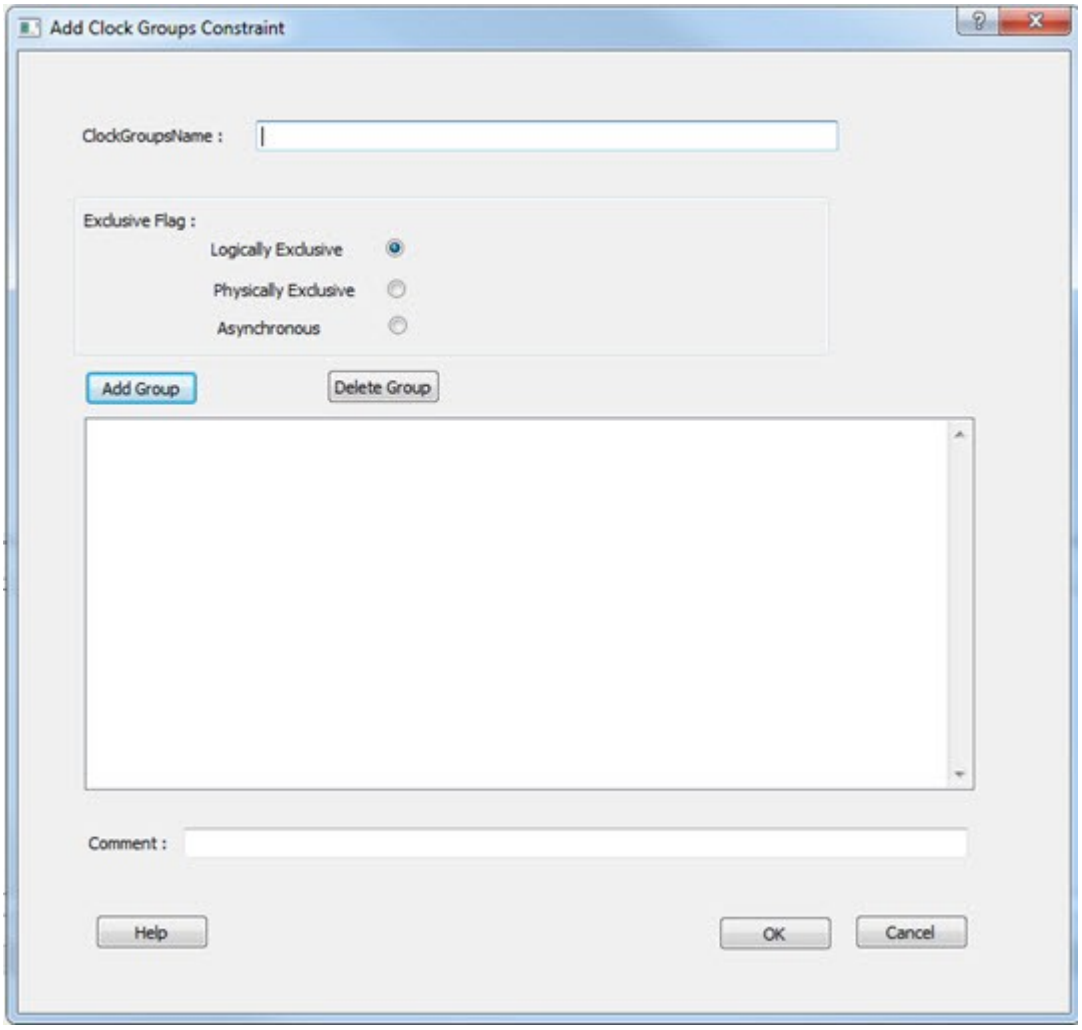
.....continued	
Option	**Description**
Falling same as rising	Specifies that the rising and falling clock edges have the same latency.
Early same as Late	Specifies that the clock source latency must be considered a single value, not a range from **early** to **late**.
Comment	Enter a one-line comment to describe the clock source latency.

4.4 Set a Clock Group (Ask a Question)

To add or delete a clock group constraint, open the **Add Clock Groups Constraint** dialog box in one of the following ways:

- From the **Constraints** menu, click **Clock Groups**.
- In the **Constraints Browser**, double-click **Clock Groups**.
- Right-click any row of **Clock Groups Constraints Table** and click **Add Clock Groups**.

Figure 4-6. Add Clock Groups Constraints Dialog Box



The following table describes the **Add Clock Groups Constraints** dialog box options.

Table 4-4. Add Clock Groups Constraints Options

Option	Description
ClockGroupsName	Enter a name for the clock groups to be added.
Exclusive Flag	Choose one of the following three clock group attributes for the clock group: • **Logically Exclusive** : Use this setting for clocks that can exist physically on the device at the same time, but are logically exclusive (for example, multiplexed clocks). • **Physically Exclusive** : Use this setting for clocks that cannot exist physically on the device at the same time (for example, multiple clocks defined on the same pin). • **Asynchronous** : Use this setting when there are valid timing paths between the two clock groups, but the two clocks have no frequency or phase relationship and therefore these timing paths can be excluded from timing analysis.
Add Group	Click **Add** to open a dialog box to add clocks to a clock group. In the **Available Pins** list, click the clocks, and then click **Add** to move them to the **Assigned Pins** list. Click **OK**.
Delete Group	Delete the clocks from a clock group. Select the group of clocks to be deleted and click **Delete Group**. This will delete the clock group.

4.5 Modifying a System Jitter Value (Ask a Question)

Use the `set_system_jitter` constraint to adjust the global jitter that is automatically generated by Libero.

5. Advanced Constraints (Ask a Question)

The following section details the advanced constraints.

5.1 Set a Disable Timing Constraint (Ask a Question)

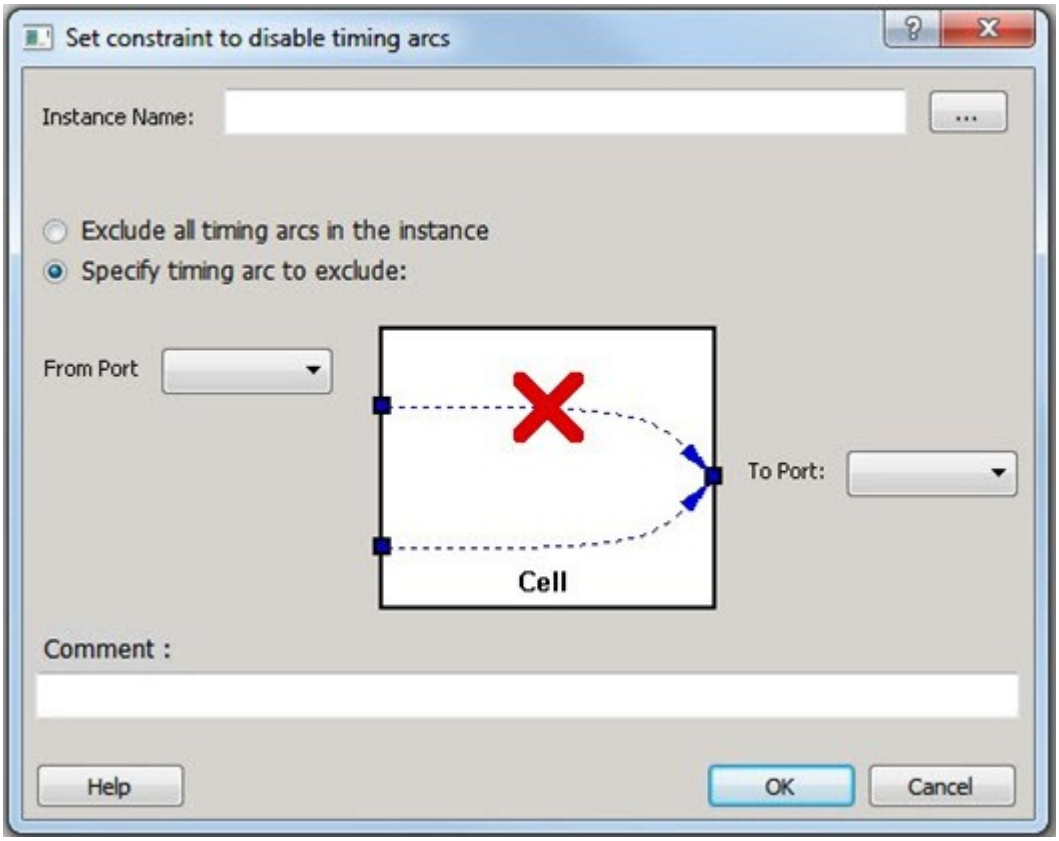
Use disable timing constraint to specify the timing arcs to be disabled for timing consideration.

To specify a Disable Timing constraint, open the **Set Constraint to Disable Timing Arcs** dialog box in one of the following four ways:

- In the **Constraints Browser**, double-click **Disable Timing**.
- Click the **Add Disable Timing Constraint**  icon.
- From the **Constraints** menu, click **Disable Timing**.
- Right-click any row of **Disable Timing Constraints Table** and click **Add Constraint to Disable Timing**. The **Set Constraint to Disable Timing Arcs** dialog box appears.

Note: This constraint is for the Place and Route tool and the Verify Timing tool. It is ignored by the Synthesis tool.

Figure 5-1. Set constraint to disable timing arcs Dialog Box



The following table describes the **Set Constraint to Disable Timing Arcs** dialog box options.

Table 5-1. Set Constraint to Disable Timing Arcs Options

Option	Description
Instance Name	Specifies the instance name for which the disable timing arc constraint will be created. Click **Browse** next to the **Instance Name** box to open the **Select instance to constrain** dialog box. The following options are available on the **Select instance to constrain** dialog box: • **Type**: Displays the type of the available pins in the design. All instances is the only valid type. • **Pattern**: The default is *, which is a wild-card match for all. You can specify any string value. Click **Filter** to filter the available pins based on the specified pin type and pattern. The list box displays the available instances. If you change the pattern value, the list box shows the available instances based on the filter. Select instances from this list and click **OK** to select the instance to constrain.
Exclude All Timing Arcs in the Instance	Enables you to exclude all timing arcs in the specified instance.
Specify Timing Arc to Exclude	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	Enter a one-line comment for the constraint.

5.1.1 Specifying Disable Timing Constraint (Ask a Question)

Use disable timing constraint to specify the timing arcs being disabled.

To specify the disable timing constraint:

1. Add the constraint in the **Editable Constraints Grid** or open the **Set Constraint to Disable Timing Arcs** dialog box using one of the following methods:
   - From the **Constraints** menu, click **Disable Timing**.
   - Click the  icon in the **Constraints Editor**.
   - In the **Constraints Editor**, right-click **Disable Timing** and click **Add Disable Timing Constraints.**
2. Select an instance from your design.
3. Select whether you want to exclude all timing arcs in the instance or if you want to specify the timing arc to exclude. If you select **Specify timing arc to exclude**, select a from and to port for the timing arc.
4. Enter any comments to be attached to the constraint.
5. Click **OK**. The new constraint appears in the constraints list.
   Note: When you click **Save** from the **File** menu, the newly created constraint is saved in the database.

5.2 Set Clock Source Latency Constraint (Ask a Question)

Use the clock source latency constraint 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.

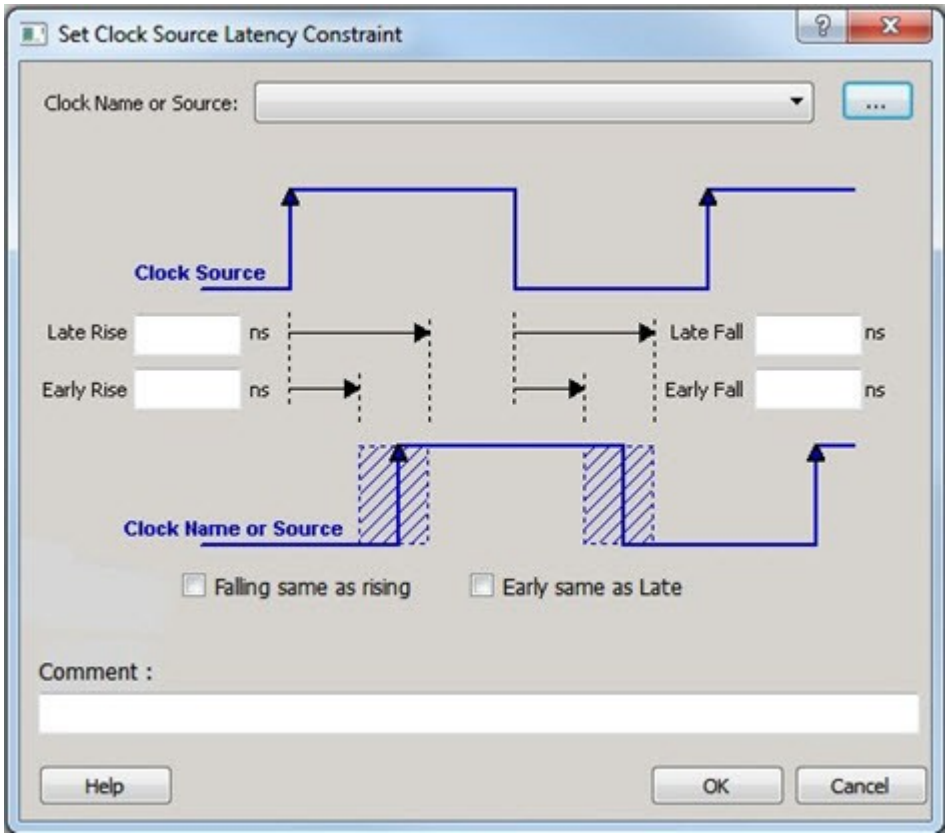
You can specify both an "early" delay and a "late" delay for this latency, providing an uncertainty that the timing analyzer can use for propagating through its calculations. Rising and falling edges of the same clock can have

different latencies. If only one clock source latency value is provided, the value is taken as the exact latency value for both rising and falling edges.

To specify a Clock Source Latency constraint, open the **Set Clock Source Latency Constraint** dialog box in one of the following four ways:

- In the **Constraints Browser**, double-click **Clock Source Latency**.
- Click the **Clock Source Latency Constraint** icon.
- From the **Constraints** menu, click **Clock Source Latency** .
- Right-click any row of **Clock Latency Constraints Table** and click **Add Clock Source Latency**. The **Set Clock Source Latency Constraint** dialog box appears.

Figure 5-2. Set Clock Source Latency Constraint Dialog Box



The following table describes the **Set Clock Source Latency Constraint** dialog box options.

Table 5-2. Set Clock Source Latency Constraint Options

Option	Description
Clock Name or Source	To select the clock source, click the **Browse** button to open the **Choose the Clock Source Pin** dialog box. The only choice available for Pin Type is **Clock Pins**.
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.
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.

.....continued	
Option	**Description**
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.
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.
Falling same as rising	Specifies that the rising and falling clock edges have the same latency.
Early same as Late	Specifies that the clock source latency must be considered a single value, not a range from **early** to **late**.
Comment	Enter a one-line comment to describe the clock source latency.

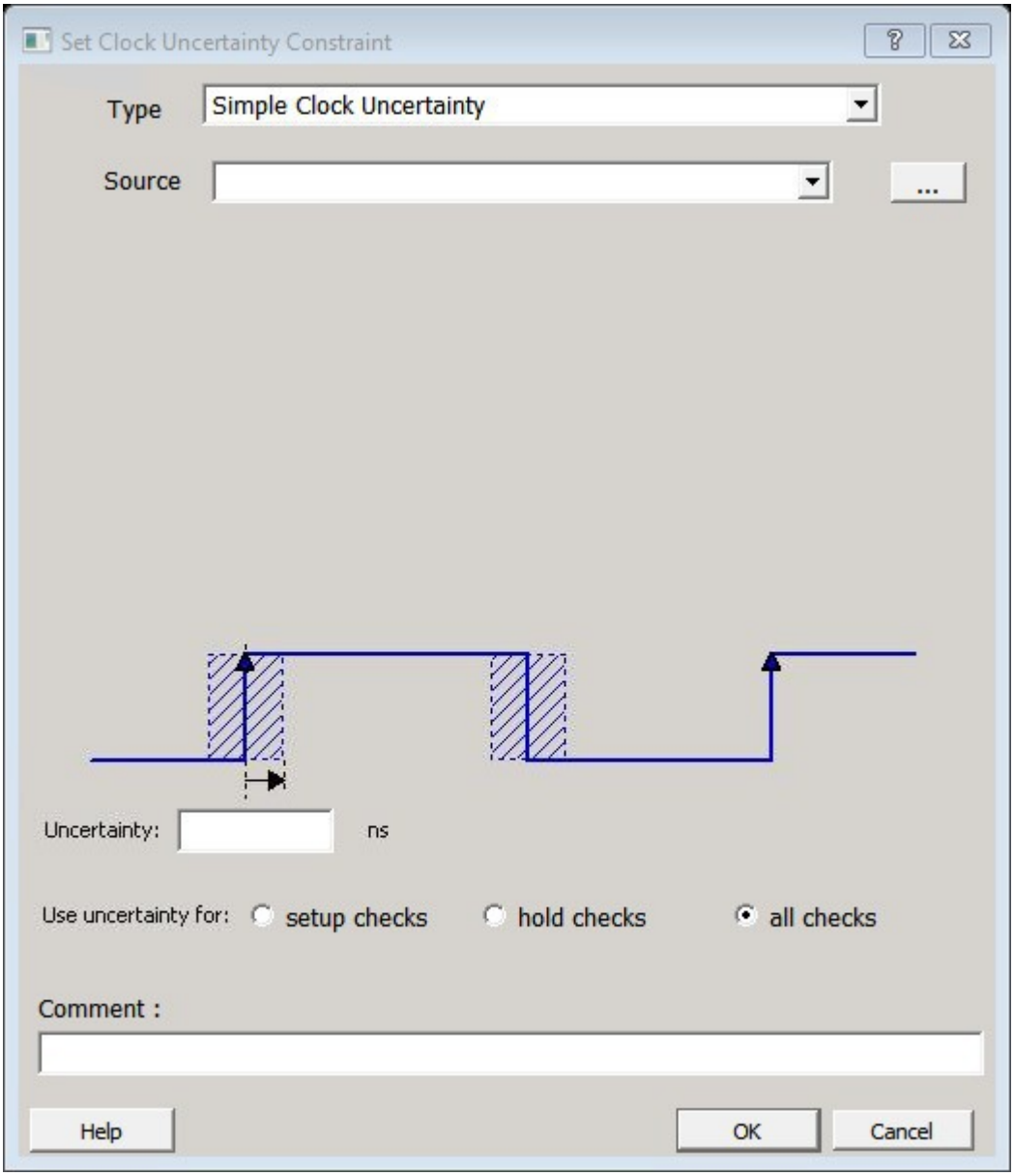
5.3 Set a Clock Uncertainty Constraint (Ask a Question)

Use the **Set Clock Uncertainty Constraint** dialog box to set either the Simple Clock Uncertainty constraint or the Clock To Clock Uncertainty constraint from the **Type** drop-down menu. The default is Simple Clock Uncertainty.

5.3.1 Set Simple Clock Uncertainty Constraint (Ask a Question)

To open the **Set Clock Uncertainty Constraint** dialog box from the **Constraints** menu, choose **Clock Uncertainty** and select **Simple Clock Uncertainty** from the **Type** drop-down menu in **Set Clock Uncertainty Constraint** dialog box.

Figure 5-3. Set Clock Uncertainty Constraint dialog box



The following table describes the **Set Clock Uncertainty Constraint** dialog box options.

Table 5-3. Set Clock Uncertainty Constraint Options

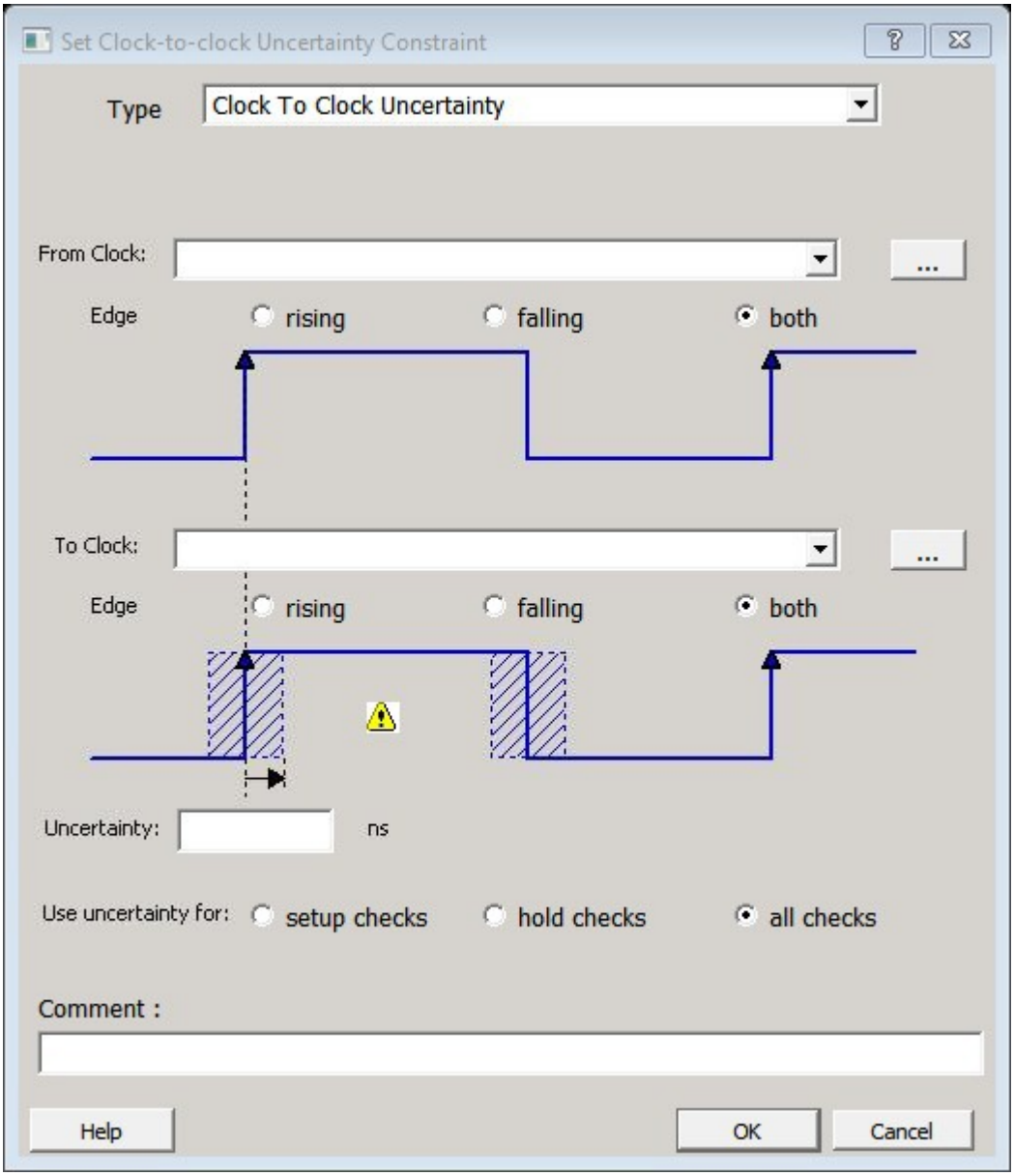
Option	Description
Type	Select **Simple Clock Uncertainty** (default) to set clock uncertainty constraint on a single clock.

.....continued	
Option	**Description**
Source	Specifies the clock name as the uncertainty source. To set the source clock, click the **Browse** button to open the **Select Source for Simple Uncertainty Constraint** dialog box. The following options are available in the **Select Source for Simple Uncertainty Constraint** dialog box: • **Type**: Displays the type of available pins in the design. The available **Pin Type** options are: – Clock Pins – All Pins – All Ports • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins or ports (ports are defined as clock signal in the design). If you change the pattern value, the list box displays the available pins or ports based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the source to the constraint.
Uncertainity	Enter the time in `ns` that represents the amount of deviation between two clock edges.
Use uncertainty for	Enables you to 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.

5.3.2 Set Clock-to-Clock Uncertainty Constraint (Ask a Question)

To open the **Set Clock-to-clock Uncertainty Constraint** dialog box from the **Constraints** menu, choose **Clock Uncertainty**, and select **Clock To Clock Uncertainty** from the **Type** drop-down menu.

Figure 5-4. Set Clock-to-Clock Uncertainty Constraint Dialog Box



The following table describes the **Set Clock-to-clock Uncertainty Constraint** dialog box options.

Table 5-4. Set Clock-to-Clock Uncertainty Constraint Options

Option	Description
Type	Select **Clock To Clock Uncertainty**.

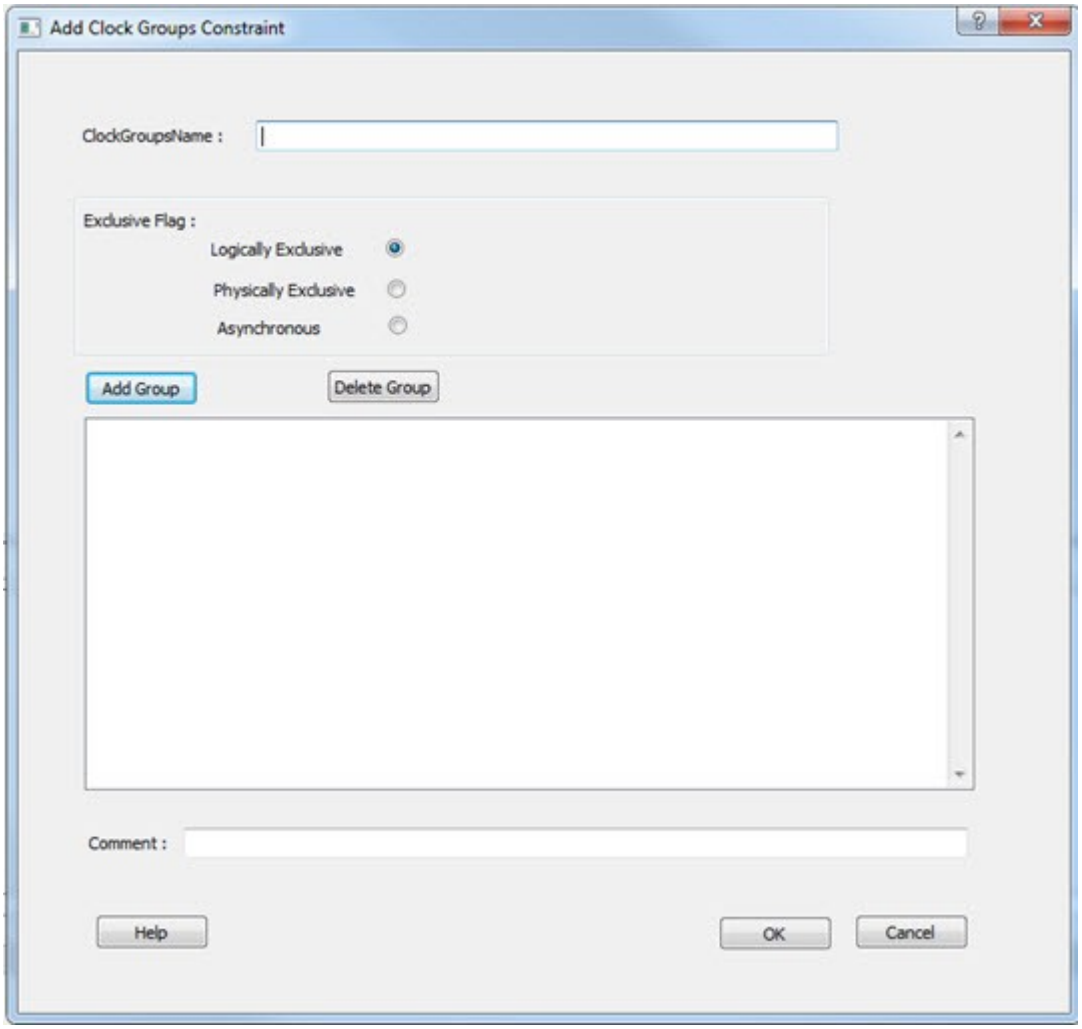
.....continued	
Option	**Description**
From Clock	Specifies clock name as the uncertainty source. To set the From Clock, click the **Browse** button to open the **Select Source Clock List for Clock-to-clock Uncertainty** dialog box. The following options are available in the **Select Source Clock List for Clock-to-clock Uncertainty** dialog box: • **Type**: Displays the type of available pins in the design. The only choice available for Pin Type is **Clock Pins**. • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins. If you change the pattern value, the list box displays the available pins based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the source clock for the Clock-to-Clock Uncertainty constraint.
Edge	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. To set the To Clock, click the **Browse** button to open the **Select Destination Clock List for Clock-to-clock Uncertainty Constraint** dialog box. The following options are available in the **Select Destination Clock List for Clock-to-clock Uncertainty Constraint** dialog box: • **Type**: Displays the type of available pins in the design. The only valid selection is Clock Pins. • **Pattern**: Default is *, which is a wild-card match for all. You can specify any string value. Click **Search** to filter the available pins based on the specified Pin Type and Pattern. • **Available Pins**: Displays the available clock pins. If you change the pattern value, the list box displays the available pins based on the filter. Use **Add**/**Add All** to add the clock pins from the Available Pins list to the Assigned Pins list or **Remove**/**Remove All** to delete the clock pins from the Assigned Pins list. • **Assigned Pins**: Displays pins selected from the Available Pins list. Select Pins from this list and click **OK** to add the destination clock for the Clock-to-Clock Uncertainty constraint.
Uncertainty	Enter the time in `ns` that represents the amount of deviation between two clock edges.
Use Uncertainty For	Enables you to 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.

5.4 Set a Clock Group (Ask a Question)

To add or delete a clock group constraint, open the **Add Clock Groups Constraint** dialog box in one of the following ways:

- From the **Constraints** menu, click **Clock Groups**.
- In the **Constraints Browser**, double-click **Clock Groups**.
- Right-click any row of **Clock Groups Constraints Table** and click **Add Clock Groups**.

Figure 5-5. Add Clock Groups Constraints Dialog Box



The following table describes the **Add Clock Groups Constraints** dialog box options.

Table 5-5. Add Clock Groups Constraints Options

Option	Description
ClockGroupsName	Enter a name for the clock groups to be added.
Exclusive Flag	Choose one of the following three clock group attributes for the clock group: • **Logically Exclusive** : Use this setting for clocks that can exist physically on the device at the same time, but are logically exclusive (for example, multiplexed clocks). • **Physically Exclusive** : Use this setting for clocks that cannot exist physically on the device at the same time (for example, multiple clocks defined on the same pin). • **Asynchronous** : Use this setting when there are valid timing paths between the two clock groups, but the two clocks have no frequency or phase relationship and therefore these timing paths can be excluded from timing analysis.
Add Group	Click **Add** to open a dialog box to add clocks to a clock group. In the **Available Pins** list, click the clocks, and then click **Add** to move them to the **Assigned Pins** list. Click **OK**.
Delete Group	Delete the clocks from a clock group. Select the group of clocks to be deleted and click **Delete Group**. This will delete the clock group.

5.5 Set External Delay Constraint (Ask a Question)

Use set external timing constraint to specify the external delay between user-specified From and To ports (outside of chip). The delay is considered during Timing Analysis for PLL external feedback delay calculation when the PLL output goes outside of the chip through the From pin, and re-enters the chip through the To pin, which then connects to the PLL feedback clock input pin.

Important: The Synplify Pro Synthesis software does not support this constraint. In Libero flow, this constraint is skipped for Synplify Pro Synthesis software.

To set an external delay constraint, right click any row of Set External Delay Constraints table, and then click **Add/ Edit Set External Delay**. When the Add Set External Delay Constraint dialog box appears, complete the options, and then click **OK**.

Figure 5-6. Add Set External Delay Constraint Dialog Box

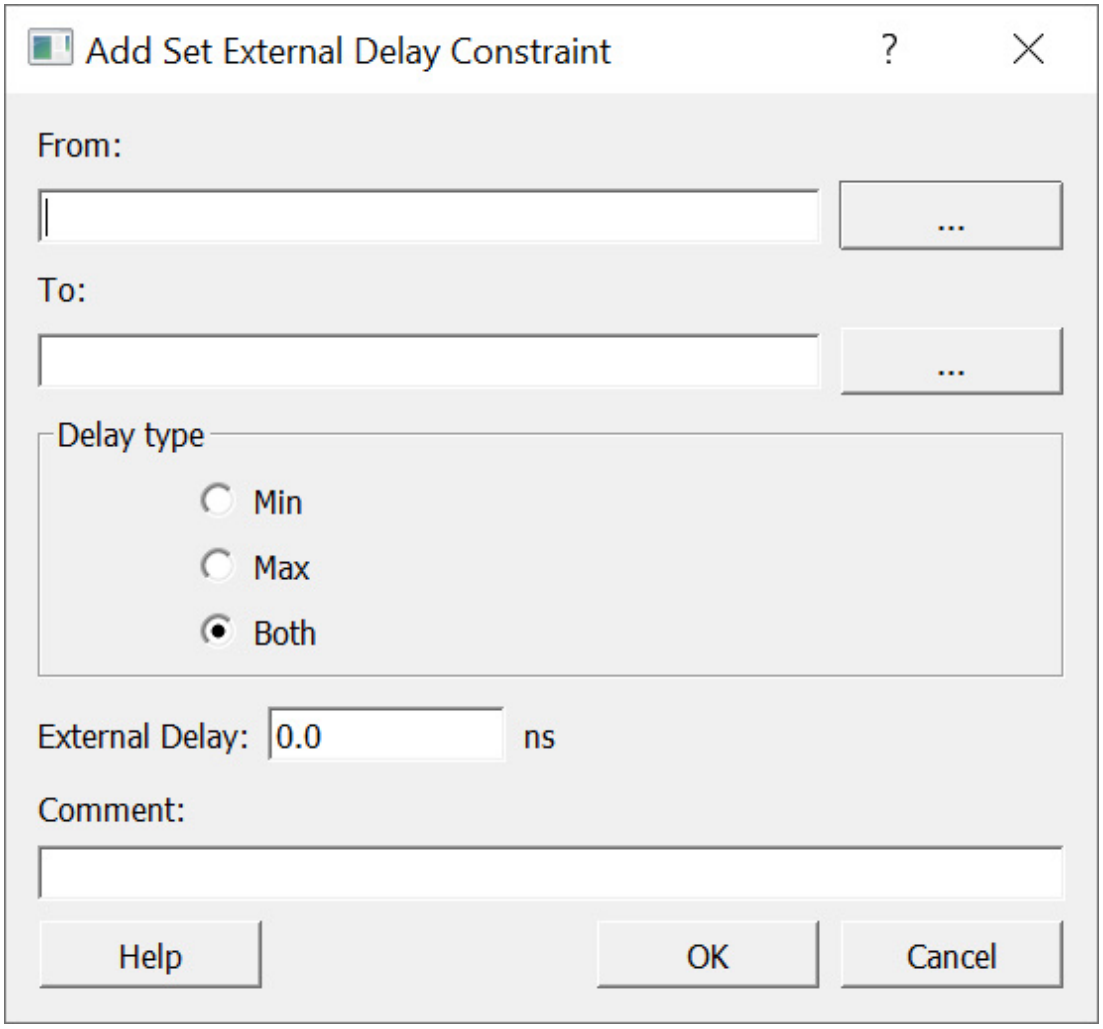


Table 5-6. Add Set External Delay Constraint Dialog Options

Option	Description
From	Specify the output port or click **Browse** to display the Select From Port for Set External Delay dialog box. Available options are: • **Pin Type**: Displays the pin type to search. The only valid selection is Output Ports and cannot be changed. • **Pattern**: The default value is *, which is a wildcard match for all. You can specify any string value. To filter the output port based on a specified pattern, click **Filter**, select one output port from the filtered results, and click **OK**.
To	Specify the input port or click **Browse** to display the Select To Port for Set External Delay dialog box. Available options are: • **Pin Type**: Displays the pin type to search. The only valid selection is Input Ports and cannot be changed. • **Pattern**: The default value is *, which is a wildcard match for all. You can specify any string value. To filter the input port based on a specified pattern, click **Filter**, select one input port from the filtered results, and click **OK**.
Delay type	Choose one of the following delay type attributes for the external delay: • **Min**: The External Delay is considered during the minimum analysis. • **Max**: The External Delay is considered during the maximum analysis. • **Both**: The External Delay is considered for both minimum and maximum analysis.
External Delay	Specify the external delay in nanoseconds between the From and To ports.
Comment	Enter a one-line comment for this constraint.

6. SmartTime Tcl Command Reference (Ask a Question)

6.1 all_inputs (Ask a Question)

Description
Returns an object representing all input and inout pins in the current design. This command is usually used with a command which puts the same attributes on input ports. If you want only certain ports, use the `get_ports` command.

```
all_inputs
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
object	Returns an object representing all input and inout pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You can only use this command as part of a `-from`, `-to` argument in the following Tcl commands: `set_min_delay`, `set_max_delay`, `set_multicycle_path`, and `set_false_path`. It cannot be used with `-through` option.

Example
The following example sets a maximum delay by constraining all paths from all_inputs to ck1 clock with a delay less then 2 ns.

```
set_max_delay 2 -from [all_inputs] -to [get_clocks ck1]
```

Related Examples on GitHub

- all_inputs

6.2 all_outputs (Ask a Question)

Description

Returns an object representing all output and inout pins in the current design. This command is usually used with a command which puts the same attributes on output ports. If you want only certain ports, use `get_ports` command.

```
all_outputs
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
object	Returns an object representing all output and inout pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You can only use this command as part of a `-from`, `-to` argument in the following Tcl commands: `set_min_delay`, `set_max_delay`, `set_multicycle_path`, and `set_false_path`. It cannot be used with `-through` option.

Example

The following example sets a maximum delay by constraining all paths from `all_inputs` to `all_outputs` with a delay less then 2 ns.

```
set_max_delay 2 -from [all_inputs] -to [all_outputs]
```

Related Examples on GitHub

- all_outputs

6.3 all_registers (Ask a Question)

Description

Returns an object representing register pins or register cells (default) in the current scenario based on the given parameters. If you do not specify an option, this command returns an object representing registers cells.

```
all_registers [-clock clock_name ] [-async_pins] \
[-output_pins] [-data_pins] [-clock_pins]
```

Arguments

Parameter	Type	Description
`clock`	string	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`	None	Lists all register pins that are async pins for the specified clock (or all registers asynchronous pins in the design).
`output_pins`	None	Lists all register pins that are output pins for the specified clock (or all registers output pins in the design).
`data_pins`	None	Lists all register pins that are data pins for the specified clock (or all registers data pins in the design).
`clock_pins`	None	Lists all register pins that are clock pins for the specified clock (or all registers clock pins in the design).

Return Type	Description
`object`	Returns an object representing register pins or cells in the current scenario based on the given parameters.

Error Codes

Error Code	Description
Error: SDC0021	Invalid max delay constraint: the -from value is incorrect.
Error: SDC0023	Invalid max delay constraint: the -to value is incorrect.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

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

Example

The following example sets a maximum delay by constraining all paths from `ff_m:CLK` or `ff_s2:CLK` to `ff_m:Q pin` with a delay less than 2.000 ns.

```
set_max_delay 2.000 -from { ff_m:CLK ff_s2:CLK } \
-to [all_registers -clock_pins -clock {ff_m:Q}]
```

Related Examples on GitHub

- all_registers

6.4 check_constraints (Ask a Question)

Description
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.

When a constraint file is checked, the Constraint Checker does the following:

- Checks the syntax.
- Compares the design objects (pins, cells, nets, ports) in the constraint file versus the design objects in the netlist (RTL or post-layout ADL netlist). Any discrepancy (for example, constraints on a design object which does not exist in the netlist) are flagged as errors and reported in the `*_sdc.log` file.

```
check_constraints
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example checks timing constraints in the current scenario.

```
check_constraints
```

6.5 clone_scenario (Ask a Question)

Description
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. You must provide a unique name (that is, it cannot already be used by another timing scenario).
Note: It is recommended to use the `organize_tool_files` command instead of `clone_scenario`.

```
clone_scenario original new_scenario_name
```

Arguments

Parameter	Type	Description
`original`	string	Specifies the name of the source timing scenario to clone (copy). The source must be a valid, existing timing scenario.
`new_scenario_name`	string	Specifies the name of the new scenario to be created.

Supported Families

PolarFire®

PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example creates a new timing scenario with the name `my_new_scenario` by duplicating an existing one (primary).

```
clone_scenario primary my_new_scenario
```

See Also
- 6.8. create_scenario
- 6.46. remove_scenario
- 6.48. rename_scenario

Related Examples on GitHub
- clone-scenario

6.6 create_clock (Ask a Question)

Description
Creates a clock constraint on the specified sources in the current design, or a virtual clock if no source other than a name is specified. It also 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.

```
create_clock [ -name clock_name ] [-add] -period period_value \
[ -waveform edge_list ][ source_objects ]
```

Table 6-1. Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the clock constraint. You must specify either a clock name or a source. If the `-name` option is not used, the clock name is specified as source name. The clock name is used to refer to the clock in other commands. You can specify name as: `-name {clk}` or `-name clk`.
`add`	None	Specifies that a new clock constraint is created at the same source port as the existing clock without overriding the existing constraint. The name of the new clock constraint with the `-add` option must be different than the existing clock constraint. Otherwise, it will override the existing constraint, even with the `-add` option. The `-name` option must be specified with the `-add` option.

.....continued		
Parameter	**Type**	**Description**
period	real	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	real	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. So in the edge list, the falling edge value must be greater than the rising edge value. For example, a clock waveform of period 19 that has a rising edge at 3 ns and a falling edge at 8 ns will have the waveform defined as [3 8].
source_objects	list of string	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. Specify either a source or a clock name.

Return Type	Description
integer	Returns the ID of the clock constraint.

Table 6-2. Error Codes

Error Code	Description
Error: SDC0001	Invalid clock constraint: clock source is incorrect.
Error: SDC0006	Invalid clock constraint: clock period is incorrect for the specified clock.
Error: SDC0007	Invalid clock constraint: waveform is incorrect.
Error: SDC0061	Invalid clock constraint: Missing or Illegal parameter/value.
Error: SDC0069	Invalid clock constraint: Need to specify clock name with -add option.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example creates two clocks, one on port CK1 with a period of 6, and the other on port 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}]
```

The following example creates a new clock constraint `clk2`, in addition to `clk1`, on the same source port clk1 without overriding it.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports clk1]
```

```
create_clock -name clk2 -add -period 20 -waveform {0 10} [get_ports clk1]
```

The following example does not add a new clock constraint, even with the `-add` option, but overrides the existing clock constraint because of the same clock names.
Note: To add a new clock constraint in addition to the existing clock constraint on the same source port, the clock names must be different.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports clk1]
```

```
create_clock -name clk1 -add -period 50 -waveform {0 25} [get_ports clk1]
```

The following example shows the SDC constraint that must be added for 050 devices, with max accuracy of 4% and 52 MHz (clock period 19.230 ns).

```
create_clock -name {OSC_0/I_RCOSC_25_50MHZ/CLKOUT} -period 19.230 [ get_pins {OSC_0/
I_RCOSC_25_50MHZ/CLKOUT}]
```

Related Examples on GitHub

- create_clock

6.7 create_generated_clock (Ask a Question)

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.

```
create_generated_clock [-name clock_name ] [-add] \
                       [-master_clock clock_name ] \
                       -source reference_pin \
                       [- divide_by divide_factor ] \
                       [-multiply_by multiply_factor ] \
                       [-invert] source [-edges values ] \
                       [-edge_shift values ]
```

Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the clock constraint. If the `-name` option is not used, the generated clock receives the same name specified in the source. The clock name is used to refer to the clock in other commands. You can specify `-name {my_gen_clk}` or `-name my_gen_clk`.

.....continued

Parameter	Type	Description
add	None	Specifies that the generated clock constraint is a new clock constraint in addition to the existing one at the same source. Use this option to capture the case where multiple generated clocks must be specified on the same source, because multiple clocks fan into the master pin. If you specify this option, you must also use the -name option. The name of the clock constraint should be different from the existing clock constraint. With this option, the -master_clock and -name options must be specified.
master_clock	string	Specifies the master clock used for the generated clock when multiple clocks fan into the master pin. This option must be used in conjunction with -add option of the generated clock. **Notes:** 1. The master_clock option is used only with the -add option for the generated clocks. 2. If there are multiple master clocks fanning into the same reference pin, the first generated clock specified uses the first master clock as its source clock. 3. The subsequent generated clocks specified with the -add option can choose any of the master clocks as their source clock (including the first master clock specified).
source	string	Specifies the reference pin in the design from which the clock waveform is to be derived. You must specify the -source reference pin.
divide_by	integer	Specifies the frequency division factor. This option cannot be used with -egde list. If -egde is specified, the divide_by value defaults to one. For example, if the divide_factor is equal to 2, the generated clock period is twice the reference clock period. If you set the divide_by value to 1.2 or 4/2 or 8a2, it is being truncated as 1 or 4 or 8, and no warning is reported.
multiply_by	integer	Specifies the frequency multiplication factor. This option cannot be used with -egde list. If egde is specified, the multiply_by and divide_by values default to one. For example, if the multiply_factor is equal to 2, the generated clock period is half the reference clock period. If you set the multiply_by value to 1.2 or 4/2 or 8a2, it is being truncated as 1 or 4 or 8, and no warning is reported.
invert	None	Specifies that the generated clock waveform is inverted with respect to the reference clock.
source	list of strings	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. You must specify a source.
edges	list of integers	Specifies a list of positive integers that represents the edges from the source clock that are to form the edges of the generated clock. To generate the clock, three values must be specified. If you specify less than three values, a tool tip indicates an error. This option cannot be used with the -divide_by/-multiply_by factor.

.....continued

Parameter	Type	Description
edge_shift	list of floating point numbers	Specify a list of three floating point numbers that represents the amount of shift (in nanoseconds) that the specified edges are to undergo to yield the final generated clock waveform. These floating point values can be positive or negative. Positive value indicates a shift later in time, while negative indicates a shift earlier in time. With this option, the -edges option must be specified.

Return Type	Description
integer	Returns the ID of the generated clock constraint.

Error Codes

Error Code	Description
Error: SDC0004	Invalid generated clock constraint: Name does not match any clock name or source.
Error: SDC0015	Invalid generated clock constraint: Port list is incorrect.
Error: SDC0016	Invalid generated clock constraint: Port list is empty.
Error: SDC0061	Invalid generated clock constraint: The -edges argument is empty invoked from within command.
Error: SDC0062	Invalid generated clock constraint: The -edges list size must be three.
Error: SDC0063	Invalid generated clock constraint: The -edges list elements are not in increasing order.
Error: SDC0065	Invalid generated clock constraint: The -edges cannot be used with the -multiply_by or -divide_by option.
Error: SDC0066	Invalid generated clock constraint: The -edge_shift does not have accompanying -edges.
Error: SDC0069	Invalid clock constraint: Need to specify a clock name with the -add option.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

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 3⁄4 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 new generated clock gen2 in addition to gen1 derived from same master clock as the existing generated clock, and the new constraint is added to pin `r1/CLK`.

```
create_generated_clock -name gen1 -multiply_by 1 \
                       -source [get_ports clk1] [get_pins r1/CLK]
create_generated_clock -name gen2 -add -master_clock clk1 \
                       -source [get_ports clk1] \
                       -multiply_by 2 [get_pins r1/CLK]
```

The following example does not create a new generated clock constraint in addition to the existing clock, but overrides even with the `-add` option enabled because the same names are used.

```
create_generated_clock -name gen2 -source [get_ports clk1] \
                       -multiply_by 3 [get_pins r1/CLK]
create_generated_clock -name gen2 -source [get_ports clk1] \
                       -multiply_by 4 -master_clock clk1 \
                       -add [get_pins r1/CLK]
```

The following example shows an SDC constraint for a generated clock of 50 MHz reference clock and 100 MHz output clock with a 90° phase shift.

```
create_generated_clock \
          -name {FCCC_0/GL0} -multiply_by 4 -divide_by 2
\
          -source [get_pins {FCCC_0/CCC_INST/RCOSC_25_50MHZ}] \
          -phase 0 [get_pins {FCCC_0/CCC_INST/GL0}]
```

```
create_generated_clock \
          -name {FCCC_0/GL1} -multiply_by 4 -divide_by 2 \
          -source [get_pins {FCCC_0/CCC_INST/RCOSC_25_50MHZ}] \
          -phase 90 [get_pins {FCCC_0/CCC_INST/GL1}]
```

See Also

- 6.6. create_clock
- 6.40. remove_generated_clock

Related Examples on GitHub

- create_generated_clock

6.8 create_scenario (Ask a Question)

Description

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).

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.

Note: It is recommended to use the `organize_tool_files` command instead of this command.

```
create_scenario name
```

Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the new timing scenario. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example creates a new timing scenario with the "scenario_A" name.

```
create_scenario scenario_A
```

Related Examples on GitHub

- create_scenario

See Also

- 6.5. clone_scenario
- 6.31. list_scenario
- 6.46. remove_scenario
- 6.48. rename_scenario

6.9 create_set (Ask a Question)

Description
Creates a set of paths to be analyzed. Use the arguments to specify which paths to include. To create a set that is a subset of a clock domain, specify it with the `-clock` and `-type` arguments. To create a set that is a subset of an inter-clock domain set, specify it with the `-source_clock` and `-sink_clock` arguments. To create a set that is a subset (filter) of an existing named set, specify the set to be filtered with the `-parent_set` argument.

```
create_set\ -name <name>\ -parent_set <name>\ -type <set_type>\ -clock <clock name> \ -
source_clock <clock name>\ -sink_clock <clock name>\ -in_to_out\ -source <port/pin pattern> \
-sink <port/pin pattern>
```

Arguments

Parameter	Type	Description
name	string	Specifies a unique name for the newly created path set.
parent_set	string	Specifies the name of the set to filter from.
clock	string	Specifies that the set is to be a subset of the given clock domain. This argument is valid only if you also specify the -type argument.
type	string	Specifies the predefined set type on which to base the new path set. You can only use this argument with the -clock argument, not by itself. • reg_to_reg - paths between registers in the design. • async_to_reg - paths from asynchronous pins to registers. • reg_to_async - paths from registers to asynchronous pins. • external_recovery - the set of paths from inputs to asynchronous pins. • external_removal - the set of paths from inputs to asynchronous pins. • external_setup - paths from input ports to registers. • external_hold - paths from input ports to registers. • clock_to_out - paths from registers to output ports.
in_to_out	None	Specifies that the set is based on the "Input to Output" set, which includes paths that start at input ports and end at output ports.
source_clock	string	Specifies that the set will be a subset of an inter-clock domain set with the given source clock. You can only use this option with the -sink_clock argument.
sink_clock	string	Specifies that the set will be a subset of an inter-clock domain set with the given sink clock. You can only use this option with the -source_clock argument.
source	string	Specifies a filter on the source pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.
sink	string	Specifies a filter on the sink pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example creates set with "my_user_set" name. Filters all C* ports and D* pins in the current design.

```
create_set -name { my_user_set } -source { C* } -sink { D* }
```

The following example creates set with `my_other_user_set` name that is a subset (filter) of an existing "my_user_set" set.

```
create_set -name { my_other_user_set } -parent_set { my_user_set } -source { CL* }
```

The following example creates set with `another_set` name which is the subset of an inter-clock domain set with the given source clock.

```
create_set -name { another_set } -source_clock { EXTERN_CLOCK } \
-sink_clock { MY_GEN_CLOCK }
```

Related Examples on GitHub

- create_set

See Also

- 6.47. remove_set

6.10 expand_path (Ask a Question)

Description

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

Parameter	Type	Description
`index value`	list of integers	Specify the index of the path to be expanded in the list of paths and display them. The index starts at 1, and defaults to 1. If index value is less than 1, then it is considered as 1. List of specified indexes can be not sequential. Only the paths with indices lower than the max_paths option value will be expanded.
`analysis {min|max}`	string	Specify whether the timing analysis is done via max-delay (setup check) or min-delay (hold check). Valid values are min/max or mindelay/maxdelay.
`format {csv | text}`	string	Specify the file format of the output. It can be either text—ASCII text format (default) or csv (comma separated values).
`set`	string	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.

.....continued

Parameter	Type	Description
clock	string	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	string	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 t specify a set name. • reg_to_reg—paths between registers in the design. • async_to_reg—path from asynchronous pins to registers. • reg_to_async—path from registers to asynchronous pins. • external_recovery—set of paths from input ports to asynchronous pins. • external_removal—set of paths from input ports to asynchronous pins. • 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.
from_clock	string	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	string	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.

Return Type	Description
string	Displays expanded path information (path details) for paths.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example displays first expanded path information (path details) for paths between registers in the design.

```
puts [expand_path -clock { myclock } -type { reg_to_reg }]
```

The following example displays expanded paths details with 1, 2, and 3 indexes from list of paths.

```
puts [expand_path -clock { myclock } -type { reg_to_reg } -index { 1 2 3 } -format text]
```

Related Examples on GitHub

- expand_path

See Also

- 6.30. list_paths

6.11 get_cells (Ask a Question)

Description

Returns a collection of instance (cell) objects in the current design that match a specified search pattern. You can use this command only as part of a `-from`, `-to` argument in the following Tcl commands: `set_max delay`, `set_multicycle_path`, and `set_false_path`. Wildcards can be used to select multiple cells at once. If no objects match the criteria, the empty string is returned.

```
get_cells pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	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 wild card character that represents any character string. This is mandatory.

Return Type	Description
`object`	Returns an object representing the cells (instances) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets maximum delay constraining all paths from reg* cells to out ports with a delay less than 2 ns.

```
set_max_delay 2 -from [get_cells {reg*}] -to [get_ports {out}]
```

Related Examples on GitHub

- get_cells

See Also

- 6.12. get_clocks
- 6.14. get_nets
- 6.15. get_pins
- 6.16. get_ports

6.12 get_clocks (Ask a Question)

Description

Returns an object representing the clock(s) that match those specified in the current timing scenario. Wildcards can be used to select multiple clocks at once. If no objects match the criteria, the empty string is returned.

- 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.

```
get_clocks pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	Mandatory. Specifies the pattern to match to the SmartTime on which a clock constraint has been set.

Return Type	Description
`object`	Returns an object representing the clock(s) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter `_AtclParam0_` is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets maximum delay constraining all paths from datal port to ck1 clock with a delay less then 2 ns.

```
set_max_delay -from [get_ports datal] -to [get_clocks ck1]
```

Related Examples on GitHub

- get_clocks

See Also

- 6.6. create_clock
- 6.7. create_generated_clock

6.13 get_current_scenario (Ask a Question)

Description

Returns the name of the current timing scenario.

```
get_current_scenario
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Name of the current timing scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the name of the current timing scenario.

```
get_current_scenario
```

Related Examples on GitHub

- get_current_scenario

See Also

- 6.8. create_scenario
- 6.55. set_current_scenario
- 6.46. remove_scenario
- 6.48. rename_scenario

6.14 get_nets (Ask a Question)

Description
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. Wildcards can be used to select multiple nets at once. If no objects match the criteria, the empty string is returned.

```
get_nets pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	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. This is mandatory.

Return Type	Description
`object`	Returns an object representing the nets that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example sets maximum delay constraining all paths from `RDATA1` port passing `-through net_chkp1 net_chkqi` nets.

```
set_max_delay 2 -from [get_ports RDATA1] -through [get_nets {net_chkp1 net_chkqi}]
```

The following example specifies all paths through the nets Tblk/rm/n* to be false.

```
set_false_path -through [get_nets {Tblk/rm/n*}]
```

The following example creates a clock on cknet net with a period of 2.5 ns.

```
create_clock -name  mainCLK -period 2.5 [get_nets {cknet}]
```

Related Examples on GitHub

- get_nets

See Also

- 6.6. create_clock
- 6.7. create_generated_clock
- 6.59. set_false_path
- 6.63. set_min_delay
- 6.62. set_max_delay
- 6.64. set_multicycle_path

6.15 get_pins (Ask a Question)

Description

Returns an object representing the pin(s) that match those specified in the pattern argument. Wildcards can be used to select multiple pins at once. If no objects match the criteria, the empty string is returned.

```
get_pins pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	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. This is mandatory.

Return Type	Description
`object`	Returns an object representing the pin(s) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a clock on pin `clock_gen/reg2:Q` with a period of 10 ns.

```
create_clock -period 10 [get_pins clock_gen/reg2:Q]
```

Related Examples on GitHub

- get_pins

See Also

- 6.6. create_clock
- 6.7. create_generated_clock
- 6.59. set_false_path
- 6.63. set_min_delay
- 6.62. set_max_delay
- 6.64. set_multicycle_path

6.16 get_ports (Ask a Question)

Description

Returns an object representing the port(s) that match those specified in the pattern argument. Wildcards can be used to select multiple ports at once. If no objects match the criteria, the empty string is returned.

```
get_ports pattern
```

Arguments

Parameter	Type	Description
pattern	string	Specifies the pattern to match the ports.

Return Type	Description
object	Returns an object representing the port(s) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a clock on port CK1 with a period of 10 ns.

```
create_clock -period 10 [get_ports CK1]
```

Related Examples on GitHub

- get_ports

See Also

- 6.6. create_clock
- 6.7. create_generated_clock
- 6.60. set_input_delay
- 6.66. set_output_delay
- 6.59. set_false_path
- 6.63. set_min_delay
- 6.62. set_max_delay
- 6.64. set_multicycle_path

6.17 list_clock_groups (Ask a Question)

Description

Returns the details for all the existing clock groups in the current timing constraint scenario.

```
list_clock_groups
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Details about all of the clock groups constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the existing clock groups in the current timing constraint scenario.

```
puts [list_clock_groups]
```

Related Examples on GitHub

- list_clock_groups

See Also

- 6.51. set_clock_groups
- 6.35. remove_clock_groups

6.18 list_clock_latencies (Ask a Question)

Description
Returns details about all of the clock latencies in the current timing constraint scenario.

```
list_clock_latencies
```

Arguments

Return Type	Description
string	Returns details about all of the clock latencies in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the clock latencies in the current timing constraint scenario.

```
puts [list_clock_latencies]
```

Related Examples on GitHub
- list_clock_latencies

See Also
- 6.52. set_clock_latency
- 6.36. remove_clock_latency

6.19 list_clock_uncertainties (Ask a Question)

Description
Returns details about all of the clock uncertainties in the current timing constraint scenario.

```
list_clock_uncertainties
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Returns details about all of the clock uncertainties.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the clock uncertainties in the current timing constraint scenario.

```
puts [list_clock_uncertainties]
```

Related Examples on GitHub

- list_clock_uncertainties

See Also

- 6.54. set_clock_uncertainty
- 6.37. remove_clock_uncertainty

6.20 list_clocks (Ask a Question)

Description
Returns details about all of the clock constraints in the current timing constraint scenario.

```
list_clocks
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Returns details about all of the clock constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example displays the details about all of the clock constraints in the current timing constraint scenario.

```
puts [list_clocks]
```

Related Examples on GitHub
- list_clocks

See Also
- 6.6. create_clock
- 6.34. remove_clock

6.21 list_disable_timings (Ask a Question)

Description
Returns the list of disable timing constraints for the current scenario.

```
list_disable_timings
```

Arguments

Return Type	Description
string	Returns list of disable timing constraints for the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the disable timing constraints in the current timing constraint scenario.

```
puts [list_disable_timings]
```

Related Examples on GitHub
- list_disable_timings

See Also
- 6.56. set_disable_timing
- 6.38. remove_disable_timing

6.22 list_false_paths (Ask a Question)

Description
Returns details about all of the false paths in the current timing constraint scenario.

```
list_false_paths
```

Arguments

Return Type	Description
string	Returns details about all of the false paths in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the false paths in the current timing constraint scenario.

```
puts [list_false_paths]
```

Related Examples on GitHub
- list_false_paths

See Also
- 6.59. set_false_path
- 6.39. remove_false_path

6.23 list_generated_clocks (Ask a Question)

Description
Returns details about all of the generated clock constraints in the current timing constraint scenario.

```
list_generated_clocks
```

Arguments

Return Type	Description
string	Returns details about all of the generated clock constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example displays the details about all of the generated clock constraints in the current timing constraint scenario.

```
puts [list_generated_clocks]
```

Related Examples on GitHub

- list_generated_clocks

See Also

- 6.7. create_generated_clock
- 6.40. remove_generated_clock

6.24 list_input_delays (Ask a Question)

Description
Returns details about all of the input delay constraints in the current timing constraint scenario.

```
list_input_delays
```

Arguments

Return Type	Description
string	Details about all of the input delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the input delay constraints in the current timing constraint scenario.

```
puts [list_input_delays]
```

Related Examples on GitHub

- list_input_delays

See Also

- 6.41. remove_input_delay
- 6.60. set_input_delay

6.25 list_max_delays (Ask a Question)

Description

Returns details about all of the maximum delay constraints in the current timing constraint scenario.

```
list_max_delays
```

Arguments

Return Type	Description
string	Details about all of the max delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the maximum delay constraints in the current timing constraint scenario.

```
puts [list_max_delays]
```

Related Examples on GitHub

- list_max_delays

See Also

- 6.42. remove_max_delay
- 6.62. set_max_delay

6.26 list_min_delays (Ask a Question)

Description

Returns details about all of the minimum delay constraints in the current timing constraint scenario.

```
list_min_delays
```

Arguments

Return Type	Description
string	Details about all of the min delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the minimum delay constraints in the current timing constraint scenario.

```
puts [list_min_delays]
```

Related Examples on GitHub
- list_min_delays

See Also
- 6.43. remove_min_delay
- 6.63. set_min_delay

6.27 list_multicycle_paths (Ask a Question)

Description
Returns details about all of the multicycle paths in the current timing constraint scenario.

```
list_multicycle_paths
```

Arguments

Return Type	Description
string	Returns details about all of the multicycle paths in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the details about all of the multicycle paths constraints in the current timing constraint scenario.

```
puts [list_multicycle_paths]
```

Related Examples on GitHub
- list_multicycle_paths

See Also
- 6.44. remove_multicycle_path
- 6.64. set_multicycle_path

6.28 list_objects (Ask a Question)

Description
Returns a list of object matching the parameter. Objects can be nets, pins, ports, clocks, or instances.

```
list_objects <object>
```

Arguments

Parameter	Type	Description
objects	string	Any timing constraint parameter (object can be nets, pins, ports, clocks, or instances). This is mandatory.

Return Type	Description
list of objects	Returns a list of nets, pins, ports, clocks, or instances.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

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*]
```

Related Examples on GitHub

- list_objects

6.29 list_output_delays (Ask a Question)

Description

Returns details about all of the output delay constraints in the current timing constraint scenario.

```
list_output_delays
```

Arguments

Return Type	Description
string	Details about all of the output delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the output delay constraints in the current timing constraint scenario.

```
puts [list_output_delays]
```

Related Examples on GitHub

- list_output_delays

See Also

- 6.45. remove_output_delay
- 6.66. set_output_delay

6.30 list_paths (Ask a Question)

Description

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

Parameter	Type	Description
analysis	string	Specifies whether the timing analysis is done for max-delay (setup check) or min-delay (hold check). Valid values are: max or min.
format	string	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	string	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	string	Returns a list of paths from the specified clock domain. This option requires the -type option. You cannot use wildcards when specifying a clock name.
type	string	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 in the design. • async_to_reg -paths from asynchronous pins to registers. • reg_to_async -paths from registers to asynchronous pins of registers. • external_recovery -paths from input ports to asynchronous pins of registers. • external_removal -paths from input ports to asynchronous pins of registers. • external_setup -paths from input ports to data pins of registers. • external_hold -paths from input ports to data pins of registers. • clock_to_out -paths from registers to output ports.
from_clock	string	Used along with -to_clock to get the list of paths of the inter-clock domain between the two clocks.
to_clock	string	Used along with -from_clock to get the list of paths of the inter-clock domain between the two clocks.
in_to_out	None	Used to get the list of path between input and output ports.
from	string	Filter the list of paths to those starting from ports or pins matching the pattern.
to	string	Filter the list of paths to those ending at ports or pins matching the pattern.

Return Type	Description
list of strings	Returns a list of the n worst paths matching the arguments.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

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 ]
```

Related Examples on GitHub
- list_paths

6.31 list_scenario (Ask a Question)

Description
Returns a list of names of all of the available timing scenarios.

```
list_scenario name
```

Arguments

Return Type	Description
list of strings	List of all names of the available timing scenarios.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
With this command we get the list of available timing scenario names.

```
list_scenario
```

Related Examples on GitHub
- list_scenario

See Also

- 6.5. clone_scenario
- 6.55. set_current_scenario
- 6.13. get_current_scenario
- 6.46. remove_scenario
- 6.48. rename_scenario

6.32 read_sdc (Ask a Question)

Description

Evaluates 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 ( : | / ) \
-ignore_errors file_name
```

Table 6-3. Arguments

Parameter	Type	Description
add	None	Specifies that the constraints from the SDC file is 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	string	Specifies the scenario to add the constraints to. The scenario is created if none exists with this name.
netlist	string	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	char	Specify the pin separator used in the SDC file. It can be either ':' or '/'.
ignore_errors	None	Optional. Specifies whether to avoid reporting errors for derived constraints targeting the logic that becomes invalid due to logic optimization. It is an optional argument. Some IPs may have extra logic present depending on other IPs used in the design but the synthesis tool will remove this logic if fewer IPs were used. In such cases, the implementation flow will halt without -ignore_errors flag. **Note:** Do not use this flag outside similar use cases.
file	string	Specify the SDC file name.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2

RTG4™

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
```

6.33 remove_all_constraints (Ask a Question)

Description
Removes all timing constraints from analysis.

```
remove_all_constraints
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example removes all timing constraints from analysis.

```
remove_all_constraints
```

Related Examples on GitHub
- remove_all_constraints

See Also
- 6.4. check_constraints

6.34 remove_clock (Ask a Question)

Description
Removes the specified clock constraint from the current timing 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 clock and port names and the constraint ID.

```
remove_clock -name clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the clock constraint to remove from the current scenario. Specify either a clock name or an ID. **Note:** Specify clock name as {CLK}, not [get_clocks {CLK}].
id	integer	Specifies the ID of the clock constraint to remove from the current scenario. Specify either an ID or a clock name that exists in the current scenario.

Error Codes

Error Code	Description
None	Invalid clock name argument.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock names.

Example

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
```

Related Examples on GitHub

- remove_clock

See Also

- 6.6. create_clock
- 6.7. create_generated_clock

6.35 remove_clock_groups (Ask a Question)

Description
Removes a clock group by specifying its name or its group ID. If the arguments do not match, or if the ID does not refer to a clock group, the command fails.
Note: The exclusive flag is not needed when removing a clock group by ID. These flags are mutually exclusive. Only one can be specified.

```
remove_clock_groups [-id constraint_ID | -name groupname ] \
[-physically_exclusive | -logically_exclusive | -asynchronous]
```

Arguments

Parameter	Type	Description
id	integer	Specifies the clock group by the ID. You must specify either a clock group ID or a clock group name that exists in the current scenario.
name	string	Specifies the clock group by name (to be always followed by the exclusive flag).
physically_exclusive	None	Specifies that the clock groups are physically exclusive with respect to each other.
logically_exclusive	None	Specifies that the clocks groups are logically exclusive with respect to each other.
asynchronous	None	Specifies that the clock groups are asynchronous with respect to each other.

Error Codes

Error Code	Description
None	Only one argument is needed.
None	Invalid clock Groups name argument.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock groups name.

Example
The following commands removes clock groups with the `mygroup3` names and the clock groups with id 12.

```
remove_clock_groups -name mygroup3 -physically_exclusive
```

```
remove_clock_groups id 12
```

Related Examples on GitHub
- remove_clock_groups

See Also
- 6.51. set_clock_groups
- 6.17. list_clock_groups

6.36 remove_clock_latency (Ask a Question)

Description
Removes a clock source latency from the specified clock and from all edges of the clock. 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 clock and port names and the constraint ID.

```
remove_clock_latency -source clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
`source`	`string`	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 name or an ID.
`id`	`integer`	Specifies the ID of the clock constraint to remove the clock source latency from the current scenario. You must specify either an ID or a clock name that exists in the current scenario.

Error Codes

Error Code	Description
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying the clock names.

Example
The following example removes the clock source latency from the specified clock.

```
remove_clock_latency -source [get_clocks {my_clock} ]
```

Related Examples on GitHub
- remove_clock_latency

See Also
- 6.52. set_clock_latency

6.37 remove_clock_uncertainty (Ask a Question)

Description
Removes a clock-to-clock uncertainty from the current timing scenario by specifying either its exact arguments or its ID.

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.

```
remove_clock_uncertainty -from | -rise_from | -fall_from from_clock_list -to | -rise_to | \
-fall_to to_clock_list -setup {value} -hold {value} | -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	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	list of strings	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	list of strings	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	list of strings	Specifies the list of clock names as the uncertainty source.
to	list of strings	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.

.....continued

Parameter	Type	Description
rise_to	list of strings	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	list of strings	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	list of strings	Specifies the list of clock names as the uncertainty destination.
setup	None	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	None	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	integer	Specifies the ID of the clock constraint to remove the clock source uncertainty from the current scenario. You must specify either the exact parameters to set the constraint or its constraint ID.

Error Codes

Error Code	Description
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example removes uncertainties from Clk1 clock to Clk2 clock domains.

```
remove_clock_uncertainty -from [ get-clock {Clk1} ] -to [ get_clock {Clk2} ]
```

The following example removes uncertainties between Clk1, Clk2, Clk3, and Clk4 clock domains with specific edges.

```
remove_clock_uncertainty -from [ get_clocks {Clk1} ]  -fall_to [ get_clocks {Clk2 Clk3} ] -
setup
```

```
remove_clock_uncertainty 4.3 -fall_from [ get_clocks {Clk1 Clk2} ] -rise_to [ get_clocks {*} ]
```

```
remove_clock_uncertainty 0.1 -rise_from [ get_clocks {Clk1 Clk2} ] \
-fall_to [ get_clocks {Clk3 Clk4} ] -setup
```

```
remove_clock_uncertainty 5 -rise_from [ get_clocks {Clk1} ] -to [ get_clocks {*} ]
```

```
remove_clock_uncertainty -id $clockId
```

Related Examples on GitHub

- remove_clock_uncertainty

See Also

- 6.19. list_clock_uncertainties
- 6.54. set_clock_uncertainty

6.38 remove_disable_timing (Ask a Question)

Description

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 constraint_ID
```

Arguments

Parameter	Type	Description
from	string	Specifies the starting port. The -from and -to arguments must either both be present or both omitted for the constraint to be valid.
name	string	Specifies the cell(instance) 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 command.
to	string	Specifies the ending port. The -from arguments must either both be present or both omitted for the constraint to be valid.
id	string	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 command.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as `get_pins` or `get_ports`.

Example
The following example removes disable timing constraint between A and Y ports.

```
remove_disable_timing -from A -to Y -id new_constraint
```

Related Examples on GitHub
- remove_disable_timing

See Also
- 6.56. set_disable_timing
- 6.21. list_disable_timings

6.39 remove_false_path (Ask a Question)

Description
Removes a false path constraint from the current timing scenario by specifying either its exact arguments or its ID. 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.
Note: Do not specify both false path arguments and the constraint ID.

```
remove_false_path [-from from_list] [-to to_list] [-through through_list] | -id constraint_ID
```

Arguments

Parameter	Type	Description
`from`	`list of strings`	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`	`list of strings`	Specifies a list of pins, ports, nets, or instances (cells) through which the disabled paths must pass.

.....continued		
Parameter	**Type**	**Description**
`to`	`list of strings`	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`	`integer`	Specifies the ID of the false path constraint to remove from the current scenario. You must specify either the exact false path arguments to remove or the constraint ID that refers to the false path constraint to remove.

Error Codes

Error Code	Description
None	Invalid arguments -from/-to/-through.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as `get_pins` or `get_ports`.

Example

The following example specifies all false path 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
```

Related Examples on GitHub

- remove_false_path

See Also

- 6.59. set_false_path

6.40 remove_generated_clock (Ask a Question)

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 clock and port names and the constraint ID.

```
remove_generated_clock -name clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
name	string	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	integer	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.

Error Codes

Error Code	Description
None	Invalid clock name argument.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock name.

Example
The following example removes the generated clock constraint named `my_user_clock`

```
remove_generated_clock -name my_user_clock
```

Related Examples on GitHub
- remove_generated_clock

See Also
- 6.7. create_generated_clock

6.41 remove_input_delay (Ask a Question)

Description
Removes an input delay by specifying both the clocks and port names or the ID of the input delay constraint to remove. 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.

```
remove_input_delay -clock clock_name port_pin_list | -id constraint_ID
```

Arguments

Parameter	Type	Description
clock	string	Specifies the clock name to which the specified input delay value is assigned. **Note:** You must specify clock name as {CLK}, not [get_clocks {CLK}].
port_pin_list	list of strings	Specifies the port names to which the specified input delay value is assigned.
id	integer	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.

Error Codes

Error Code	Description
None	Parameter -clock has illegal value.
None	Invalid clock/port arguments.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock or port names, either alone or in an accessor command.

Example
The following example removes the input delay from CLK1 on port data1.

```
remove_input_delay -clock [get_clocks CLK1] [get_ports data1]
```

Related Examples on GitHub

- remove_input_delay

See Also

- 6.60. set_input_delay

6.42 remove_max_delay (Ask a Question)

Description

Removes a maximum delay constraint from the current timing scenario by specifying either its exact arguments or its ID. 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 maximum delay arguments and the constraint ID.

```
remove_max_delay [-from from_list] [-to to_list] [-through through_list]
```

```
remove_max_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	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	list of strings	Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.
to	list of strings	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	integer	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.

Error Codes

Error Code	Description
None	Invalid arguments -from/-to/-through.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock or port name, either alone or in an accessor command.

Example
The following example specifies a range of maximum delay constraints to remove.

```
remove_max_delay -through U0/U1:Y
```

Related Examples on GitHub
- remove_max_delay

See Also
- 6.62. set_max_delay
- 6.63. set_min_delay
- 6.43. remove_min_delay

6.43 remove_min_delay (Ask a Question)

Description
Removes a minimum delay constraint from the current timing scenario by specifying either its exact arguments or its ID. 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 minimum delay arguments and the constraint ID.

```
remove_min_delay [-from from_list] [-to to_list] [-through through_list]
```

```
remove_min_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
`from`	list of strings	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`	list of strings	Specifies a list of pins, ports, cells, or nets through, which the disabled paths must pass.
`to`	list of strings	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`	integer	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.

Error Codes

Error Code	Description
None	Invalid arguments `-from/-to/-through`.

.....continued	
Error Code	**Description**
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock or port name, either alone or in an accessor command.

Example
The following example specifies a range of minimum delay constraints to remove.

```
remove_min_delay -through U0/U1:Y
```

Related Examples on GitHub
- remove_min_delay

See Also
- 6.63. set_min_delay

6.44 remove_multicycle_path (Ask a Question)

Description
Removes a multicycle path constraint from the current timing scenario by specifying either its exact arguments or its ID. 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.
Note: Do not specify both multicycle path arguments and the constraint ID.

```
remove_multicycle_path [-from from_list] [-to to_list] [-through through_list]
```

```
remove_multicycle_path -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	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	list of strings	Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.

.....continued		
Parameter	**Type**	**Description**
to	list of strings	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	string	Specifies the ID of the multicycle path constraint to remove from the current scenario. Specify either the exact multicycle path arguments to remove or the constraint ID that refers to the multicycle path constraint to remove.

Error Codes

Error Code	Description
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as `get_pins` or `get_ports`.

Example
The following example specifies all the multicycle paths between reg1 and reg2.

```
remove_multicycle_path -from [get_pins {reg1}] -to [get_pins {reg2}]
```

Related Examples on GitHub
- remove_multicycle_path

See Also
- 6.64. set_multicycle_path

6.45 remove_output_delay (Ask a Question)

Description
Removes an output delay by specifying both the clocks and port names or the ID of the output_delay constraint to remove. 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.

Note: Do not specify both the clock and port names and the constraint ID.

```
remove_output_delay -clock clock_name port_pin_list
```

```
remove_output_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
`clock`	string	Specifies the clock name to which the specified output delay value is assigned. **Note:** You must specify clock name as `{CLK}`, not `[get_clocks {CLK}]`.
`port_pin_list`	list of strings	Specifies the port names to which the specified output delay value is assigned.
`id`	integer	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.

Error Codes

Error Code	Description
None	Parameter -clock has illegal value.
None	Invalid clock/port arguments.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock or port names, either alone or in an accessor command.

Example

The following example removes the output delay from `CLK1` on port `out1`.

```
remove_output_delay -clock {CLK1} [get_ports out1]
```

Related Examples on GitHub

- remove_output_delay

See Also

- 6.66. set_output_delay

6.46 remove_scenario (Ask a Question)

Description
Removes a scenario from the constraint database and removes it to the list of scenarios.

```
remove_scenario name
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the scenario to delete. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a set name to remove.

Example
The following command removes the scenario named my_scenario.

```
remove_scenario my_scenario
```

Related Examples on GitHub
- remove_scenario

See Also
- 6.8. create_scenario
- 6.5. clone_scenario
- 6.48. rename_scenario

6.47 remove_set (Ask a Question)

Description
Removes a set of paths from analysis. Only user-created sets can be deleted.

```
remove_set -name name
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the set of paths to delete.

Error Codes

Error Code	Description
None	Required parameter -name is missing.
None	Unable to find set.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a set name.

Example
The following command removes the set named my_set.

```
remove_set -name my_set
```

Related Examples on GitHub
- remove_set

See Also
- 6.9. create_set

6.48 rename_scenario (Ask a Question)

Description
Renames an existing timing scenario to a new name. The new name you provide must be unique and cannot be used by another timing scenario.

Note: It is recommended to use the organize_tool_files command instead of this command.

```
rename_scenario old_name new_name
```

Arguments

Parameter	Type	Description
old_name	string	Specifies the name of the existing timing scenario to be renamed.
new_name	string	Specifies the new name for the new scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following command renames the my_old_scenarioscenario name into a my_new_scenario new name.

```
rename_scenario my_old_scenario my_new_scenario
```

Related Examples on GitHub

- rename_scenario

See Also

- 6.8. create_scenario
- 6.5. clone_scenario
- 6.46. remove_scenario

6.49 report (Ask a Question)

Description
Specifies the type of reports to be generated and the type of Analysis (Max Delay or Min Delay) performed to generate the reports. Using this command, 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.
- Clock Domain Crossing report—This report analyzes timing paths that cross from one clock domain (the source clock) to another clock domain (the destination clock).

If the specified parameter/value is not correct, this command fails.

```
report -type (timing | violations | datasheet | bottleneck | constraints_coverage |
combinational_loops | cdc) \ -analysis <max|min> \ -format (csv|text) \ <filename> \ timing
options \ -max_parallel_paths <number> \ -max_paths <number> \ -print summary (yes|no)
\ -use_slack_threshold (yes|no) \ -slack threshold <double> \ -print_paths (yes|no) \ -
max_expanded_paths <number> \ -include_user_sets (yes|no) \ -include_clock_domains (yes|no)
\ -select_clock_domains <clock name list> \ -limit_max_paths (yes|no) \ -include_pin_to_pin
(yes|no) \ bottleneck options \ -cost_type (path_count|path_cost) \ -max_instances <number>
\ -from <port/pin pattern> \ -to <port/pin pattern> \ -set_type <set_type> \ -set_name <set
name> \ -clock <clock name> \ -from_clock <clock name> \ -to_clock <clock name> \ -in_to_out \
```

Arguments

Parameter	Type	Description
type	string	Specifies the type of the report to be generated. It is mandatory. • timing—Timing Report • violations—Timing Violation Report • datasheet—Datasheet Report • bottleneck—Bottleneck Report • constraints_coverage—Constraints Coverage Report • combinational_loops—Combinational Loops Report
analysis	string	Specifies the type of Analysis (Max Delay or Min Delay) Performed to generate the reports. It is optional. **Note:** This argument should not be used to generate datasheet reports. The command may fail if this argument is used to generate datasheet report. • max—Timing report considers maximum analysis (default). • min—Timing report considers minimum analysis.
format	string	Specifies the format in which the report is generated. It is optional. • text—Generates a text report (default). • csv—Generates the report in a comma-separated value format which you can import into a spreadsheet. **Note:** CDC type generates report in CSV format only.
filename	string	Specifies the file name of the generated report. It is mandatory.

Table 6-4. Timing Options and Values

Parameter/Value	Description
max_parallel_paths <number>	Specifies the max number of parallel paths. Parallel paths are timing paths with the same start and end points.
max_paths <number>	Specifies the max number of paths to display for each set. This value is a positive integer value greater than zero. Default is 100.
print_summary (yes|no)	Yes to include and No to exclude the summary section in the timing report.
use_slack_threshold (yes|no)	Yes to include slack threshold and no to exclude threshold in the timing report. The default is to exclude slack threshold.

.....continued	
Parameter/Value	**Description**
`slack_threshold <double>`	Specifies the threshold value to consider when reporting path slacks. This value is in nanoseconds (ns). By default, there is no threshold (all slacks reported).
`print_paths <yes|no>`	Specifies whether the path section (clock domains and in-to-out paths) will be printed in the timing report. Yes to include path sections (default) and no to exclude path sections from the timing report.
`max_expanded_paths <number>`	Specifies the max number of paths to expand per set. This value is a positive integer value greater than zero. Default is 100.
`include_user_sets (yes|no)`	If yes, the user set is included in the timing report. If no, the user set is excluded in the timing report.
`include_clock_domains (yes|no)`	Yes to include and no to exclude clock domains in the timing report.

`select_clock_domains <clock_name_list>`
or
`Select_clock_domains -yes -clock_domain <clock_name_list>`
	Defines the clock domain to be considered in the clock domain section. The domain list is a series of strings with domain names separated by spaces. Both the summary and the path sections in the timing report display only the listed clock domains in the `clock_name_list`.

`limit_max_paths (yes|no)`	Yes to limit the number of paths to report. No to specify that there is no limit to the number of paths to report (default).
`include_pin_to_pin (yes|no)`	Yes to include and no to exclude pin-to-pin paths in the timing report.

Table 6-5. Bottleneck Options and Values

`cost_type <path_count|path_cost>`	Specifies the cost_type as either path_count or path_cost. For path_count, instances with the greatest number of path violations will have the highest bottleneck cost. For path_cost, instances with the largest combined timing violations will have the highest bottleneck cost.
`max_instances <number>`	Specifies the maximum number of instances to be reported. Default is 10.
`from <port/pin pattern>`	Reports only instances that lie on violating paths that start at locations specified by this option.
`to <port/pin pattern>`	Reports only instances that lie on violating paths that end at locations specified by this option.
`clock <clock name>`	This option allows pruning based on a given clock domain. Only instances that lie on these violating paths are reported.
`set_name <set name>`	Displays the bottleneck information for the named set. You can either use this option or use both -clock and -type. This option allows pruning based on a given set. Only paths that lie within the named set will be considered towards bottleneck.

set_type <set_type>	This option can only be used in combination with the -clock option, and not by itself. The options allows you to filter which type of paths should be considered towards the bottleneck: • reg_to_reg—Paths between registers in the design • async_to_reg—Paths from asynchronous pins to registers • reg_to_async—Paths from registers to asynchronous pins • external_recovery—The set of paths from inputs to asynchronous pins • external_removal—The set of paths from inputs to asynchronous pins • external_setup—Paths from input ports to registers • external_hold—Paths from input ports to registers • clock_to_out—Paths from registers to output ports
from_clock <clock name>	Reports only bottleneck instances that lie on violating timing paths of the inter-clock domain that starts at the source clock specified by this option. This option can only be used in combination with to_clock.
to_clock <clock name>	Reports only instances that lie on violating paths that end at locations specified by this option.
in_to_out	Reports only instances that lie on violating paths that begin at input ports and end at output ports.

Return Type	Description
file	Generates SmartTime report file with the specified format.

Error Codes

Error Code	Description
None	Parameter -type has illegal value
None	Required parameter -type is missing
None	Required parameter _AtclParam0_ is missing

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example generates a timing violation report named `timing_viol.txt`. The report considers an analysis using maximum delays and does not filter paths based on slack threshold. It reports two paths per section and one expanded path per section:

```
report \ -type violations \ -analysis max \ -use_slack_threshold no \ -limit_max_paths yes \
-max_paths 2 \ -max_expanded_paths 1 \ timing_viol.txt
```

The following example generates a datasheet report named `datasheet.csv` in CSV format:

```
report -type datasheet -format csv datasheet.csv
```

You can use any one of the following examples to report inter-clock domain timing paths:

```
report -type timing -select_clock_domains yes -clock_domain { clk1 clk2 } report1.txt
```

or

```
report -type timing -select_clock_domains { clk1 clk2 } report2.txt
```

Related Examples on GitHub
- report

6.50 save (Ask a Question)

Description
Saves all changes made prior to this command. This includes changes made on constraints, options, and sets.

```
save
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

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
```

Related Examples on GitHub
- save

6.51 set_clock_groups (Ask a Question)

Description
Disables timing analysis between the specified clock groups. No paths are reported between the clock groups in both directions. Paths between clocks in the same group continue to be reported.

Note: If you use the same name and the same exclusive flag of a previously defined clock group to create a new clock group, the previous clock group is removed and a new one is created in its place. The exclusive flags for the arguments above are all mutually exclusive. Only one can be specified.

```
set_clock_groups [-name name ] \
[-physically exclusive | -logically exclusive | -asynchronous] \
[-comment comment_string ] \
-group clock_list
```

Arguments

Parameter	Type	Description
name	string	Name given to the clock group.
physically_exclusive	None	Specifies that the clock groups are physically exclusive with respect to each other. Examples are multiple clocks feeding a register clock pin. The exclusive flags are all mutually exclusive. Only one can be specified.
logically_exclusive	None	Specifies that the clocks groups are logically exclusive with respect to each other. Examples are clocks passing through a mux.
asynchronous	None	Specifies that the clock groups are asynchronous with respect to each other, as there is no phase relationship between them. **Note:** The exclusive flags are all mutually exclusive. Only one can be specified.
group	list of strings	Specifies a list of clocks. There can any number of groups specified in the set_clock_groups command.

Return Type	Description
integer	Returns the ID of the clock group.

Error Codes

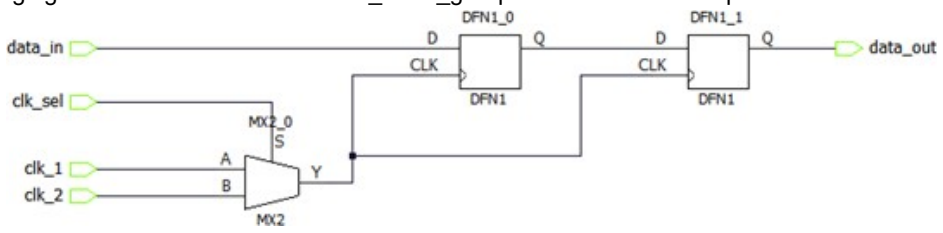
Error Code	Description
None	Invalid set_clock_groups constraint - only one of -physically_exclusive, -logically_exclusive or -asynchronous should be used.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

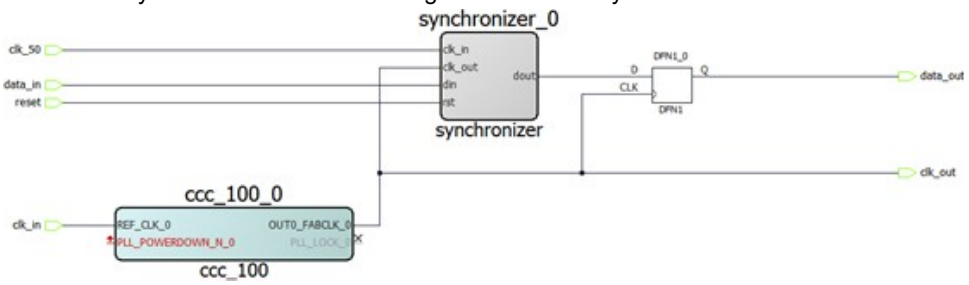
1. The following figure shows how to use the set_clock_groups constraint for multiplexed clocks..



SDC:

```
create_clock -name clk_1 -period 5 [ get_ports clk_1 ]
create_clock -name clk_2 -period 10 [ get_ports clk_2 ]
set_clock_groups -logically_exclusive -group clk_1 -group clk_2
```

2. Here, there are three synchronous clocks receiving data from an asynchronous clock.



SDC:

```
create_clock -name clk_in -period 10 [ get_ports clk_in ]
create_clock -name clk_50 -period 20 [ get_ports clk_50 ]
create_generated_clock -name ccc_100 -divide_by 2 \
-source [ get_pins ccc_100_0/ccc_100_0/pll_inst_0/REF_CLK_0 ] \
[ get_pins ccc_100_0/ccc_100_0/pll_inst_0/OUT0 ] \
create_generated_clock -name clk_out -divide_by 1 \
-source [ get_pins { ccc_100_0/ccc_100_0/pll_inst_0/OUT0 } ] \
[ get_ports clk_out ]
set_clock_groups -asynchronous -group { clk_in ccc_100 clk_out } -group clk_50
```

Related Examples on GitHub

- set_clock_groups

See Also

- 6.17. list_clock_groups
- 6.35. remove_clock_groups

6.52 set_clock_latency (Ask a Question)

Description

Defines the delay between an external clock source and the definition pin of a clock within SmartTime.

Clock source latency defines the delay between an external clock source and the definition pin of a clock within SmartTime. 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.

Clock latency is of two types:

- External latency (source) - From an external source to on-chip clock definition point using the `-source` argument.
- Internal latency (network) - From a clock generator to an end point (FF) due to clock tree synthesis (CTS).

In FPGA, internal clock latency is not used since the clock tree is already inserted and Libero SoC tool is already aware of the delay. Hence, `set_clock_latency` is primarily used to model external latency in FPGA.

When external clock latency is modeled using the `-source` arguments, hold calculations are never impacted since the hold checks occur for the same clock edge. Setup times are only impacted when the `-early` and `-late` arguments are used since otherwise the clock is uniformly delayed to all endpoints in the design. The following table summarizes the behavior:

`-source` used	`-early` used	`-late` used	Result
Yes	No	No	No setup or hold impact
Yes	No	Yes	No setup or hold impact
Yes	Yes	No	No setup or hold impact
Yes	Yes	Yes	Setup check only
No	N/A	N/A	Not recommended for FPGA designs.

Setup time is calculated for a scenario where the launch edge is delayed and the capture edge is early:

- The `-early` value is added from the required time.
- The `-late` value is added to the arrival time

```
set_clock_latency -source [-rise] [-fall] [-early] [-late] delay clock
```

Arguments

Parameter	Type	Description
`source`	None	Specifies the source latency on a clock pin, potentially only on certain edges of the clock.
`rise`	None	Specifies the edge for which this constraint will apply. If neither or both rise and fall are passed, the same latency is applied to both edges.
`fall`	None	Specifies the edge for which this constraint will apply. If neither or both fall and rise are passed, the same latency is applied to both edges.
`late`	None	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`	None	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.

.....continued		
Parameter	**Type**	**Description**
delay	floating point	Specifies the latency value for the constraint.
clock	string	Specifies the clock to which the constraint is applied. This clock must be constrained.

Error Codes

Error Code	Description
Error: SDC0061	Invalid clock latency constraint: Parameter has illegal value invoked from within command.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
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 }
```

Related Examples on GitHub
- set_clock_latency

See Also
- 6.6. create_clock
- 6.7. create_generated_clock

6.53 set_clock_to_output (Ask a Question)

Description
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

Parameter	Type	Description
delay_value	integer	Specifies the clock to output delay in nanoseconds. This time represents the amount of time available inside the FPGA between the launch clock edge and the data change at the output port.
clock	string	Specifies the reference clock to which the specified clock to output is related. This is a mandatory argument. **Note:** The clock parameter needs to be set to the clock driving the FF.
max	None	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 constraint values to be equal.
min	None	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 constraint values to be equal.
output_list	list of strings	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 ({}).

Error Codes

Error Code	Description
None	Required parameter -clock is missing

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example sets an output delay of 0.3 ns for port Q relative to the clk clock.

```
set_clock_to_output -max 0.3 -clock { clk } [ get_ports { Q } ]
```

Related Examples on GitHub

- set_clock_to_output

6.54 set_clock_uncertainty (Ask a Question)

Description
Specifies simple clock uncertainty for single clock and clock-to-clock uncertainty between two clocks (from and to).

The set_clock_uncertainty command sets the timing uncertainty of clock networks. It can be used to model clock jitter or add guard band in timing analysis.

SmartTime computes uncertainty values very similar to the set_clock_latency command (when used with the -early and -late arguments), but with one difference in the timing report. Uncertainty number is always applied to the required time calculation whereas latency numbers are applied to both arrival and required time. Uncertainty is added for both setup and hold checks.

To ensure that clock jitter is modeled accurately using the set_clock_uncertainty command, you must explicitly use -setup argument. If the -setup argument is not specified, SmartTime will apply the uncertainty value to both setup and hold calculations leading to incorrect jitter modeling.

-setup used	-hold used	Result
No	No	Setup and hold checks.
No	Yes	Hold checks only.
Yes	No	Setup checks only. For jitter modeling.
Yes	Yes	Setup and hold checks.

Either simple clock uncertainty or clock-to-clock uncertainty can be specified. Simple clock uncertainty can be set on a clock or on any pin in the clock network. It will then apply to any path with the capturing register in the forward cone of the uncertainty. If multiple simple uncertainty applies to a register, the last one (in the propagation order from the clock source to the register) is used. Clock-to-clock uncertainty applies to inter-clock paths. Both from clock and to clock must be specified. Clock-to-clock uncertainty has higher priority than simple uncertainty. If both are set (a clock-to-clock uncertainty and a simple clock uncertainty on the to clock), the simple clock uncertainty will be ignored for inter-clock paths, only the clock-to-clock uncertainty will be used.

```
set_clock_uncertainty [-setup] [-hold] uncertainty [object_list -from from_clock |
-rise_from rise_from_clock | -fall_from fall_from_clock -to to_clock | -rise_to rise_to_clock
|
-fall_to fall_to_clock ]
```

Arguments

Parameter	Type	Description
uncertainty	floating point	Specifies the time in nanoseconds that represents the amount of variation between two clock edges.
object_list	list of strings	Specifies a list of clocks, ports, or pins for simple uncertainty; the uncertainty is applied either to destination flops clocked by one of the clocks in the object list option, or destination flops whose clock pins are in the fanout of a port or a pin specified in the object_list option.
from	list of strings	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	list of strings	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	list of strings	Specifies that the clock-to-clock uncertainty applies only to falling edges of source clock list. Only one of the -from, -rise_from, or -fall_from arguments can be specified for the constraint to be valid.

.....continued

Parameter	Type	Description
from_clock/rise_from_clock/ fall_from_clock	list of strings	Specifies the list of clock names as the uncertainty source.
to	list of strings	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	list of strings	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	list of strings	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/rise_to_clock/ fall_to_clock	list of strings	Specifies the list of clock names as the uncertainty destination.
setup	None	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	None	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.

Return Type	Description
integer	Returns the ID of the clock uncertainty constraint.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

Simple Clock Uncertainty constraint examples.

The following example specifies uncertainty of 2 ns.

```
set_clock_uncertainty 2 [get_clocks clk]
```

The following example specifies setup uncertainty of 2 ns.

```
set_clock_uncertainty 2 -setup [get_clocks clk]
```

Clock to Clock Uncertainty constraint examples:

The following example specifies uncertainties of 10ns between Clk1 and Clk2 clock domains.

```
set_clock_uncertainty 10 -from [get_clocks { Clk1 }] -to [get_clocks { Clk2 }]
```

The following example specifies setup uncertainties between Clk1 and {Clk2 Clk3} clock domains with specific edges.

```
set_clock_uncertainty 0 -from [get_clocks { Clk1 }] -fall_to [get_clocks { Clk2 Clk3 }] -setup
```

```
set_clock_uncertainty 4.3 -fall_from [get_clocks { Clk1 Clk2 }] -rise_to *
```

```
set_clock_uncertainty 0.1 -rise_from [ get_clocks { Clk1 Clk2 }] \
-fall_to [get_clocks { Clk3 Clk4 }] -setup
```

```
set_clock_uncertainty 5 -rise_from [get_clocks {Clk1}]  -to [ get_clocks {*} ]
```

Related Examples on GitHub

- set_clock_uncertainty

See Also

- 6.19. list_clock_uncertainties
- 6.37. remove_clock_uncertainty

6.55 set_current_scenario (Ask a Question)

Description

Specifies the timing scenario for the Timing Analyzer to use. All commands that follow this command will apply to the specified timing scenario. 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.

Note: It is recommended to use the `organize_tool_files` command instead of this command.

```
set_current_scenario name
```

Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the timing scenario to which to apply all commands from this point on.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following command sets `scenario_A` as current scenario of the timing scenario.

```
set_current_scenario scenario_A
```

Related Examples on GitHub
- set_current_scenario

See Also
- 6.8. create_scenario
- 6.13. get_current_scenario
- 6.46. remove_scenario
- 6.48. rename_scenario

6.56 set_disable_timing (Ask a Question)

Description
Disables timing arcs within a cell (instance) and returns the ID of the created constraint if the command succeeded. To specify a Disable Timing constraint, open the Set Constraint to Disable Timing Arcs dialog box in the following way: From the **Constraints** menu, click **Disable Timing**.
Note: This constraint is for the Place and Route tool and the Verify Timing tool. It is ignored by the Synthesis tool.

```
set_disable_timing -from value -to value name
```

Arguments

Parameter	Type	Description
`from`	string	Specifies the starting point for the timing arc. The `-from` and `-to` arguments must either both be present or both omitted for the constraint to be valid.
`to`	string	Specifies the ending point for the timing arc. The `-from` and `-to` arguments must either both be present or both omitted for the constraint to be valid.
`name`	string	Specifies the instance(cell) name for which the disable timing arc constraint will be created.

Return Type	Description
`integer`	Returns the ID of created constraint.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC

SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example disables timing arcs within A to Y.

```
set_disable_timing -from A -to Y
```

Related Examples on GitHub

- set_disable_timing

See Also

- 6.38. remove_disable_timing

6.57 set_external_check (Ask a Question)

Description
Defines the external setup and hold delays for an input relative to a clock.

```
set_external_check delay_value -clock clock_ref [-setup] [-hold] input_list
```

Arguments

Parameter	Type	Description
`delay_value`	integer	Specifies the delay at the input port/pin inside the FPGA.
`clock`	string	Specifies the reference clock to which the specified external check is related. This is a mandatory argument.
`setup or hold`	None	Specifies that delay_value refers to the setup/hold check at the specified input. This is a mandatory argument if `-hold` is not used. You must specify either the `-setup` or `-hold` option.
`input_list`	list of strings	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 ({}).

Return Type	Description
`integer`	Returns the ID of the external setup and hold constraint.

Error Codes

Error Code	Description
None	Required parameter -clock is missing.

Supported Families

PolarFire®

PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example sets an external setup delay with 0.2 ns for A input port.

```
set_external_check -setup 0.2 -clock { clk } [ get_ports { A } ]
```

Related Examples on GitHub
- set_external_check

See Also
- 6.60. set_input_delay

6.58 set_external_delay (Ask a Question)

Description
Specifies the external delay between `-from` and `-to` ports (outside of chip). The delay is considered during Timing Analysis for PLL external feedback delay calculation when the PLL output goes outside of the chip through the `-from` pin, and re-enters the chip through the `-to` pin, which then connects to the PLL feedback clock input pin.

Important: This constraint is not supported by the Synplify Pro Synthesis software. In Libero flow, this constraint is skipped for Synplify Pro Synthesis software.

```
set_external_delay -from value -to value [-min] [-max] <delay_value>
```

Arguments

Parameter	Type	Description
`from`	string	Specifies the output port that is connected to PLL output. This argument is mandatory.
`to`	string	Specifies the input port that is connected to PLL feedback. This argument is mandatory.
`min`	flag	Specifies the external feedback delay for minimum analysis.
`max`	flag	Specifies the external feedback delay for maximum analysis.
`delay_value`	real	Specifies the external delay value between `-from` to `-to` ports in nanoseconds. This argument is mandatory. If neither the `-min` nor `-max` parameter value is specified, the same `delay_value` is used for both minimum and maximum analysis.

Return Type	Description
void	No return type

Error Codes

Error Code	Description
SDC0015	Invalid external delay constraint: port list <specified_port> is incorrect.
SDC0078	Invalid external delay constraint: shared -from or -to ports
SDC0061	Error in command set_external_delay: Parameter -from has illegal value
SDC0061	Error in command set_external_delay: Parameter -to has illegal value

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example sets the PLL external feedback off-chip delay between ports GL0_CLK_OUT and FB_CLK_IN to 3 ns.

```
set_external_delay -from [ get_ports { GL0_CLK_OUT} ] -to [ get_ports { FB_CLK_IN } ] 3.0
```

6.59 set_false_path (Ask a Question)

Description
Identifies paths that are considered false and excluded from the timing analysis in the current timing scenario. 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.

```
set_false_path [-ignore_errors] [-from from_list ] [-through through_list ] [-to to_list ]
```

Arguments

Parameter	Type	Description
`ignore_errors`	None	Specifies to avoid reporting errors for derived constraints targeting the logic that becomes invalid due to logic optimization. It is an optional argument. Some IPs may have extra logic present depending on other IPs used in the design but the synthesis tool will remove this logic if fewer IPs were used. In such cases, the implementation flow will halt without `-ignore_errors` flag. **Note:** It is not recommended to use this flag outside similar use cases.
`from`	list of strings	Specifies a list of timing paths 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`	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the disabled paths must pass.
`to`	list of strings	Specifies a list of timing paths ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Return Type	Description
None	None

Error Codes

Error Code	Description
Error: SDC0021	Invalid false path constraint: the `-from` value is incorrect.
Error: SDC0022	Invalid false path constraint: the `-from` is empty.
Error: SDC0024	Invalid false path constraint: the `-to` is empty.
Error: SDC0026	Invalid false path constraint: the `-through` is empty.
Warning:	cell (`get_cells`) is incorrect type; `-through` objects must be of type net (`get_nets`), or pin (`get_pins`). **Note:** Constraint will be disabled.
Warning:	port (`get_ports`) is incorrect type; `-through` objects must be of type net (`get_nets`), or pin (`get_pins`). **Note:** Constraint will be disabled.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example specifies all paths from clock pins of the registers in clock domain `clk1` to data pins of a specific register `reg_2` 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
```

The following example specifies a derived false path constraint through the `PCIe_Demo_0/SYSRESET_POR/POWER_ON_RESET_N` pin.

```
set_false_path -ignore_errors -through [ get_pins {PCIe_Demo_0/SYSRESET_POR/
POWER_ON_RESET_N } ]
```

Related Examples on GitHub
- set_false_path

See Also
- 6.39. remove_false_path

6.60 set_input_delay (Ask a Question)

Description
Creates an input delay on a port list by defining the arrival time of an input relative to a clock in the current scenario.

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}]`.

Notes:
- The behavior of the -add_delay option is identical to that of PrimeTime(TM).
- If, using the `-add_delay` mechanism, multiple constraints are otherwise identical, except they specify different `-max` or `-min` values
  - The surviving `-max` constraint will be the maximum of the `-max` values.
  - The surviving `-min` constraint will be the minimum of the `-min` values.

```
set_input_delay delay_value -clock clock_ref [-max] [-min] [-clock_fall] [-rise] [-fall] [-
add_delay] \
input_list
```

Arguments

Parameter	Type	Description
delay_value	real	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	string	Specifies the clock reference to which the specified input delay is related. This is a mandatory argument.
max	None	Specifies that the 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	None	Specifies that the 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	None	Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.
rise	None	Specifies that the delay is relative to a rising transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
fall	None	Specifies that the delay is relative to a falling transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
add_delay	None	Specifies that this input delay constraint should be added to an existing constraint on the same port(s). The -add_delay option is used to capture information on multiple paths with different clocks or clock edges leading to the same input port(s).
input_list	list of string	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 ({}).

Return Type	Description
integer	Returns the ID of the clock input delay constraint.

Error Codes

Error Code	Description
Error: SDC0004	clk does not match any clock name or source.
Error: SDC0015	port list [get_ports {CLK_0_D}] is incorrect.
Error: SDC0054	Invalid IO delay constraint: the min delay is greater than max delay.
Error: SDC0061	Parameter _AtclParam0_ has illegal value.

Supported Families

PolarFire®
PolarFire SoC

SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example sets an input delay of 1.2 ns 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}
```

The following example demonstrates an override condition of two constraints. The first constraint is overridden because the second constraint specifies a different clock for the same input.

```
set_input_delay 1.0 -clock CLK1 -max {IN1}
```

```
set_input_delay 1.4 -clock CLK2 -max {IN1}
```

The next example is almost the same as the previous one, however, in this case, the user has specified `-add_delay`, so both constraints will be honored.

```
set_input_delay 1.0 -clock CLK1 -max {IN1}
```

```
set_input_delay 1.4 -add_delay -clock CLK2 -max {IN1}
```

The following example is more complex:

- All constraints are for an input to port PAD1 relative to a rising edge clock `CLK2`. Each combination of `{-rise, -fall} x {-max, -min}` generates an independent constraint. But the max rise delay of 5 and the max rise delay of 7 interfere with each other.
- For a -max option, the maximum value overrides all lower values. Thus the first constraint will be overridden and the max rise delay of 7 will survive.

```
set_input_delay 5 -max -rise -add_delay [get_clocks CLK2] [get_ports PAD1] # will be
overridden
```

```
set_input_delay 3 -min -fall -add_delay [get_clocks CLK2] [get_ports PAD1]
```

```
set_input_delay 3 -max -fall -add_delay [get_clocks CLK2] [get_ports PAD1]
```

```
set_input_delay 7 -max -rise -add_delay [get_clocks CLK2] [get_ports PAD1]
```

Related Examples on GitHub
- set_input_delay

See Also
- 6.66. set_output_delay
- 6.41. remove_input_delay
- 6.45. remove_output_delay

6.61 set_input_jitter (Ask a Question)

Description

Sets the input jitter for a given clock.

```
set_input_jitter <input_jitter> <clock>
```

Important:

- This constraint is not supported as a Tcl command. Enter it as part of a timing constraint (`.sdc`) file.
- The SynplifyPro synthesis software ignores the `set_input_jitter` SDC constraint.

Arguments

Parameter	Type	Description
`<input_jitter>`	floating point	Specifies the input jitter value in nanoseconds. This value must be greater than zero. The input_jitter value allows the user to specify the clock jitter on the external clock. For RTG4™ FPGAs, the user must include the larger of the external clock jitter or the RTG4 Input Buffer jitter data sheet specifications using this constraint. Furthermore for RTG4 FPGAs, if the input port is a Strobe input to the RTG4 CCC SpaceWire clock and data recovery circuit, use this constraint to account for the CCC SpaceWire effective recovered clock jitter data sheet specification during STA.
`<input clock>`	string	Specifies the clock on which to apply the input clock jitter value. The user must ensure there is also a create_clock SDC constraint applied to the same clock input port to define that input as a clock for Static Timing Analysis (STA).

Example

The following example sets the input jitter to 0.1 nanosecond.

```
create_clock -name {clk} -period 8 -waveform {0 4 } [ get_ports { clk } ]
set_input_jitter 0.1 [ get_clocks { clk } ]
```

6.62 set_max_delay (Ask a Question)

Description

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 the `delay_value`. The timing engine 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 multi-cycle path constraint.

You must specify at least one of the `-from` , `-to`, or `-through` arguments for this constraint to be valid.

```
set_max_delay delay_value [-from from_list ] [-to to_list ] [-through through_list ]
```

Arguments

Parameter	Type	Description
delay_value	floating point	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	list of strings	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	list of strings	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	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the timing paths must pass.

Return Type	Description
integer	Returns the ID of the clock maximum delay constraint.

Error Codes

Error Code	Description
Error: SDC0021	Invalid max delay constraint: the -from value is incorrect.
Error: SDC0022	Invalid max delay constraint: the -from is empty.
Error: SDC0023	Invalid max delay constraint: the -to value is incorrect.
Error: SDC0024	Invalid max delay constraint: the -to is empty.
Error: SDC0026	Invalid max delay constraint: the -through is empty
Error: SDC0061	Invalid max delay constraint: Missing or Illegal parameter/value.
Warning	cell (get_cells) is incorrect type;"-through" objects must be of type net (get_nets), or pin (get_pins). **Note:** Constraint will be disabled.
Warning	port (get_ports) is incorrect type;"-through" objects must be of type net (get_nets), or pin (get_pins). **Note:** Constraint will be disabled.

Supported Families

PolarFire®
PolarFire SoC

SmartFusion® 2
IGLOO® 2
RTG4™

Example
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*]
```

Related Examples on GitHub
- set_max_delay

See Also
- 6.62. set_max_delay
- 6.42. remove_max_delay

6.63 set_min_delay (Ask a Question)

Description
Specifies the required minimum delay for timing paths in the current design. The path length should be such that the delay along the specified path should be more than the value mentioned for the `delay_value` switch. The timing engine 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 multi-cycle path constraint.

You must specify at least one of the `-from`, `-to`, or `-through` arguments for this constraint to be valid.

```
set_min_delay delay_value [-from from_list ] [-to to_list ] [-through through_list ]
```

Arguments

Parameter	Type	Description
`delay_value`	floating point	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.

.....continued

Parameter	Type	Description
from	list of strings	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	list of strings	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	list of string	Specifies a list of pins, ports, nets, or instances (cells) through which the timing paths must pass.

Return Type	Description
integer	Returns the ID of the clock minimum delay constraint.

Error Codes

Error Code	Description
Error: SDC0021	Invalid min delay constraint: the -from value is incorrect.
Error: SDC0022	Invalid min delay constraint: the -from is empty.
Error: SDC0023	Invalid min delay constraint: the -to value is incorrect.
Error: SDC0024	Invalid min delay constraint: the -to is empty.
Error: SDC0026	Invalid min delay constraint: the -through is empty.
Error: SDC0061	Invalid min delay constraint: Missing or Illegal parameter/value.
Warning	port (get_ports) is incorrect type;"-through" objects must be of type net (get_nets), or pin (get_pins).

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets a minimum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay more 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 more than 3.8 ns.

```
set_min_delay 3.8 -to [get_ports out*]
```

Related Examples on GitHub

- set_min_delay

See Also

- 6.63. set_min_delay
- 6.42. remove_max_delay

6.64 set_multicycle_path (Ask a Question)

Description

Defines a path that takes multiple clock cycles in the current scenario. 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.

```
set_multicycle_path ncycles [-setup] [-hold] [-setup_only] [-from from_list] \
[-through through_list ] [-to to_list] [ -start ] [ -end ]
```

Arguments

Parameter	Type	Description
`ncycles`	integer	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. Number of cycles must be greater than `1`. If you set ncycles as `2.2` or `4/2` or `"8a"` then it is being truncated as `2` or `4` or `8`, and no warning is reported.
`setup`	None	Optional. Applies the cycle value for the setup check only. The default hold check will be applied unless you have specified another set_multicycle_path command for the hold value.
`hold`	None	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 0 not ncycles -1 .
`setup_only`	None	Optional. Specifies that the path multiplier is applied to setup paths only.
`from`	list of strings	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`	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the multiple cycle paths must pass.
`to`	list of strings	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.

.....continued		
Parameter	**Type**	**Description**
`start`	None	Changes the clock edges used to launch and capture the data. By default, setup multicycle holds the multicycle shifts launching edge backwards (i.e. -start is the default for hold). -start allows you to change those defaults and hold multicycle by shifting the launching clock forward.
`end`	None	Changes the clock edges used to launch and capture the data. By default, setup multicycle shifts the capturing edge forward (i.e. -end is the default for setup). -end allows you to change the defaults and specify a setup multicycle by shifting the capturing clock backward.

Error Codes

Error Code	Description
Error: SDC0004	clk does not match any clock name or source.
Error: SDC0015	port list [get_ports { CLK_0_D }] is incorrect.
Error: SDC0054	Invalid IO delay constraint: the min delay is greater than max delay.
Error: SDC0061	Parameter _AtclParam0_ has illegal value.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
Multiple priority management is not supported in Microchip SoC designs. All multiple cycle path constraints are handled with the same priority.

Example
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}]
```

The following example specifies that four cycles are needed for setup only check on all paths starting at the registers in the clock domain `REF_CLK_0`.

```
set_multicycle_path -setup_only 4 -from [ get_clocks { REF_CLK_0 } ]
```

The following are few more examples of the `set_multicycle_path -setup_only` command.

```
set_multicycle_path -setup_only 4 -from [ all_registers ]
set_multicycle_path -setup_only 4 -through [ get_cells { comb_0/XOR2_0 } ]
set_multicycle_path -setup_only 4 -from [ get_clocks { PF_CCC_C0_0/PF_CCC_C0_0/pll_inst_0/
OUT0 } ] -to [ get_clocks { PF_CCC_C0_0/PF_CCC_C0_0/pll_inst_0/OUT1 } ]
```

The following example shifts lauching edge 1 clk1 cycle backward

```
set_multicycle_path -setup -start 2 -from [ get_clocks clk1 ] -to [ get_clocks clk2 ]
```

The following example shifts capturing edge 1 clk2 cycle forward

```
set_multicycle_path -setup -end 2 -from [ get_clocks clk1 ] -to [ get_clocks clk2 ]
```

Related Examples on GitHub

- set_multicycle_path

See Also

- 6.44. remove_multicycle_path

6.65 set_options (Ask a Question)

Description

Sets options for timing analysis which can be changed in the **SmartTime Options** dialog box in the SmartTime GUI. All of the options from SmartTime are passed on to place-and-route tool, and some affect timing-driven place-and-route.

```
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

Parameter	Type	Description
max_opcond	string	Sets the operating condition to use for Maximum Delay Analysis. The acceptable values for max_opcond for PolarFire can be the following: • slow_lv_ht - use slow_lv_ht conditions for maximum delay analysis • slow_lv_lt - use slow_lv_lt conditions for maximum delay analysis • fast_hv_lt - use fast_hv_lt conditions for maximum delay analysis Default is slow_lv_lt. max_opcond for SmartFusion® 2, IGLOO® 2, and RTG4™ can be as following: • 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 Default is worst.
min_opcond	string	Sets the operating condition to use for Minimum Delay Analysis. The acceptable values for min_opcond for PolarFire can be the following: • slow_lv_ht - use slow_lv_ht conditions for minimum delay analysis • slow_lv_lt - use slow_lv_lt conditions for minimum delay analysis • fast_hv_lt - use fast_hv_lt conditions for minimum delay analysis Default is fast_hv_lt. min_opcond for SmartFusion® 2, IGLOO® 2, and RTG4™ can be as following: • worst - use worst case conditions for minimum delay analysis • typical - use typical conditions for minimum delay analysis • best - use best case conditions for minimum delay analysis Default is best.
interclockdomain_analysis	string	Enables or disables inter-clock domain analysis. Value can be the following: • yes - enables inter-clock domain analysis • no - disables inter-clock domain analysis Default is no. Timing-driven place-and-route is affected by this option.

.....continued

Parameter	Type	Description
use_bibuf_loopbacks	string	Instructs the timing analysis whether to consider loopback path in bidirectional buffers (D->Y, E->Y) as false-path {no}. Default is no; i.e., loopbacks are false paths. Values can be the following: • yes - enables loopback in bibufs • no - disables loopback in bibufs
enable_recovery_removal_checks	string	Enables recovery checks to be included in max-delay analysis and removal checks in min-delay analysis. Default is no. Values can be the following: • yes - enables recovery an removal checks • no - disables recovery and removal checks
break_at_async	string	Specifies whether or not timing analysis is allowed to cross asynchronous pins (clear, reset of sequential elements). Default is yes. Values can be the following: • yes - enables breaking paths at asynchronous ports • no - disables breaking paths at asynchronous ports. Timing-driven place-and-route is affected by this option.
filter_when_slack_below	floating point	Specifies a minimum slack value for paths reported by list_paths. Not set by default.
filter_when_slack_above	floating point	Specifies a maximum slack value for paths reported by list_paths. Not set by default.
remove_slack_filters	None	Removes the slack minimum and maximum set using -filter_when_slack_below and -filter_when_slack_above.
limit_max_paths	integer	Specifies the maximum number of paths reported by list_paths. Default is 20. Number must be greater than 0.
expand_clock_network	string	Specify whether or not clock network details are reported in expand_path. Default is yes. Values can be the following: • yes - enables expanded clock network information in paths • no - disables expanded clock network information in paths.
expand_parallel_paths	integer	Specify the number of parallel paths {paths with the same ends} to include in expand_path. Default is 1. Number must be greater than 0.
analysis_scenario	string	Specify the constraint scenario to be used for timing analysis. Default scenario is Primary.
tdpr_scenario	string	Specify the constraint scenario to be used for timing-driven place-and-route. Default scenario is Primary. Timing-driven place-and-route is affected by this option.
reset	None	Reset all options to the default values, except those for analysis and TDPR scenarios, which remain unchanged.

Supported Families

PolarFire®
PolarFire SoC

SmartFusion® 2
IGLOO® 2
RTG4™

Example
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}
```

```
set_options -max_opcond {fast_hv_lt} -min_opcond {fast_hv_lt}
```

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}
```

Related Examples on GitHub

- set_options

6.66 set_output_delay (Ask a Question)

Description
Defines the output delay of an output relative to a clock in the current scenario.

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.

Notes:

- The behavior of the `-add_delay` option is identical to that of PrimeTime(TM).
- If, using the `-add_delay` mechanism, multiple constraints are otherwise identical, except they specify different `-max` or `-min` values.
  - the surviving `-max` constraint will be the maximum of the `-max` values.
  - the surviving `-min` constraint will be the minimum of the `-min` values.

```
set_output_delay [-max] [-min] delay_value -clock clock_ref [-clock_fall] [-rise] [-fall] \
[-add_delay] output_list
```

Arguments

Parameter	Type	Description
delay_value	float	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	string	Specifies the clock reference to which the specified output delay is related. This is a mandatory argument.

.....continued

Parameter	Type	Description
max	None	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	None	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	None	Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.
rise	None	Specifies that the delay is relative to a rising transition on the specified port(s). If `-rise` or `-fall` is not specified, then rising and falling delays are assumed to be equal.
fall	None	Specifies that the delay is relative to a falling transition on the specified port(s). If `-rise` or `-fall` is not specified, then rising and falling delays are assumed to be equal.
add_delay	None	Specifies that this output delay constraint should be added to an existing constraint on the same port(s). The `-add_delay` option is used to capture information on multiple paths with different clocks or clock edges leading to the same output port(s). **Notes:** • The behavior of the `-add_delay` option is identical to that of PrimeTime(TM). • If, using the `-add_delay` mechanism, multiple commands are otherwise identical, except they specify different `-max` or `-min` values. • the surviving `-max` constraint will be the maximum of the `-max` values. • the surviving `-min` constraint will be the minimum of the `-min` values.
output_list	list of string	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 ({}).

Return Type	Description
integer	Returns the ID of the clock output delay constraint.

Error Codes

Error Code	Description
Error: SDC0004	Invalid output delay constraint: clk does not match any clock name or source.
Error: SDC0015	Invalid output delay constraint: port list is incorrect.
Error: SDC0054	Invalid I/O delay constraint: the min delay is greater than max delay.
Error: SDC0061	Invalid output delay constraint: Missing or Illegal parameter/value.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets an output delay of 1.2 ns 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
```

The following example demonstrates an override condition of two constraints. The first constraint is overridden because the second constraint specifies a different clock for the same output.

```
set_output_delay 1.0 {OUT1} -clock CLK1 -max
set_output_delay 1.4 {OUT1} -clock CLK2 -max
```

The next example is almost the same as the previous one, however, in this case, the user has specified `-add_delay`, so both constraints will be honored.

```
set_output_delay 1.0 {OUT1} -clock CLK1 -max
set_output_delay 1.4 {OUT1} -add_delay -clock CLK2 -max
```

The following example is more complex:

- All constraints are for an output to port PAD1 relative to a rising edge clock CLK2. Each combination of {-rise, -fall} x {-max, -min} generates an independent constraint. But the max rise delay of 5 and the max rise delay of 7 interfere with each other.
- For a `-max` option, the maximum value overrides all lower values. Thus the first constraint will be overridden and the max rise delay of 7 will survive.

```
set_output_delay 5 [get_clocks CLK2] [get_ports PAD1] -max -rise -add_delay # will be
overridden
set_output_delay 3 [get_clocks CLK2] [get_ports PAD1] -min -fall -add_delay
set_output_delay 3 [get_clocks CLK2] [get_ports PAD1] -max -fall -add_delay
set_output_delay 7 [get_clocks CLK2] [get_ports PAD1] -max -rise -add_delay
```

Related Examples on GitHub

- set_output_delay

See Also

- 6.60. set_input_delay
- 6.41. remove_input_delay

6.67 set_system_jitter (Ask a Question)

Description

Sets the system jitter and overrides the automatically computed value.

```
set_system_jitter <system_jitter>
```

Important: This constraint is not supported as a Tcl command. Enter it as part of a timing constraint (`.sdc`) file.

Arguments

Parameter	Type	Description
`system_jitter`	floating point	Specifies the system jitter value in nanoseconds. This value must be greater than zero.

6.68 write_sdc (Ask a Question)

Description

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

Parameter	Type	Description
`scenario`	string	Specifies the scenario to write. By default the current scenario is used.
`pin_separator`	char	Specify the pin separator used in the SDC file. It can be either ':' or '/'.
`file name`	string	Specify the SDC file name.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

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 margin.sdc
```

See Also

- 6.32. read_sdc

7. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
K	04/2023	Added new sections 5.5. Set External Delay Constraint and 6.58. set_external_delay.
J	12/2022	The following changes are made in this revision: • In section 6.61. set_input_jitter: – Rearranged the order of the arguments in the syntax and arguments table. – Added a note that the Synplify Pro synthesis software ignores the `set_input_jitter` SDC constraint. – Added an example. • In section 4.1. Set a Clock Input Jitter Value, revised the cross-reference.

.....continued

Revision	Date	Description
H	08/2022	The following changes are made in this revision: • In section 2.2. Set a Generated Clock Constraint, revised the descriptions of the following options: **Phase** and **PLL Output**. • In section 2.3. Set an Input Delay Constraint, revised the description for the following options: **Input Port**, **Available Pins**; **Max Rise and Max Fall**; and **Min Rise and Min Fall**. Corrected the option name of **Add this input delay to existing one with same source**. • In section 2.4. Set an Output Delay Constraint, revised the description for the following options: **Output Port**, **Available Pins**; **Max Rise and Max Fall**; and **Min Rise and Min Fall**. • In section 3.1. Set a Maximum Delay Constraint, revised the description for the following options: **Source/From Pins**, **Available Pins**; **Through Pins**, and **Destination/To Pins**. • In section 3.2. Set a Minimum Delay Constraint, revised the description for the following options: **Source Pins/From**, **Through Pins**, and **Destination/To Pins**. • In section 3.4. Set a Multicycle Constraint, revised the description for the following options: **Source Pins/From**; **Through Pins**, **Available Pins**; and **Destination/To Pins**. • In section 3.6. Set a False Path Constraint, revised the description for the following options: **Source/From Pins**; **Through Pins**, **Available Pins**; and **Destination/To Pins**. • In section 4.2.1. Set Simple Clock Uncertainty Constraint, revised the description for the **Source** option. • In section 5.1. Set a Disable Timing Constraint, revised the description for the **Instance Name** option. • In section 6.35. remove_clock_groups, revised the example to reflect id 12. • In sections 6.36. remove_clock_latency and 6.37. remove_clock_uncertainty, revised the description for the **id** parameter. • In section 6.39. remove_false_path, revised the description for the **through** parameter. • In section 6.51. set_clock_groups, revised the description for the **name** parameter and edited the description for example 1. • In section 6.52. set_clock_latency, revised the description and removed the **latency** parameter from the arguments table. • In section 6.54. set_clock_uncertainty, revised the setup and hold calculations in the description. • In section 6.56. set_disable_timing, revised the description and example. • In section 6.57. set_external_check, revised the description for the **delay_value** and **clock** parameters. • In section 6.59. set_false_path, revised the description for the **through** parameter and example 1. • In section 6.60. set_input_delay, revised the description for the **clock** parameter. • In section 6.6. create_clock, revised the **Waveform** parameter description to read: So in edge list, falling edge value must be greater than rising edge value. • In sections 6.62. set_max_delay and 6.64. set_multicycle_path, revised the description for the **through** parameter. • In section 6.63. set_min_delay, revised the description for the **through** parameter and the descriptions for the two examples.

.....continued

Revision	Date	Description
G	06/2022	The following changes are made in this revision: • Added new chapter 4. Adjusting Clock Information. • In chapter 6. SmartTime Tcl Command Reference, added section 6.61. set_input_jitter and section 6.67. set_system_jitter. • In section 6.7. create_generated_clock, removed examples that showed CCC output maximum peak-to-peak period jitter.
F	04/2022	This document is released with Libero SoC Design Suite v2022.1 without changes from v2021.3.
E	12/2021	Editorial updates only. No technical content updates.
D	08/2021	The following changes are made in this revision: • In set_multicycle_path, added start and end parameters along with two examples for start and end respectively.
C	08/2021	The following changes are made in this revision: • In section 6.6. create_clock, added an example for an SDC constraint that must be added for 050 devices, with 4% max accuracy and 52 MHz (clock period 19.230 ns). • In section 6.7. create_generated_clock, added two examples: – One that shows an SDC constraint for a generated clock of 50 MHz reference clock and 100 MHz output clocks with a 90° phase shift. – One that shows the CCC output maximum peak-to-peak period jitter.
B	04/2021	Updated Tcl commands with link to GitHub examples.
A	12/2020	Initial Revision

Microchip FPGA Support (Ask a Question)

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information (Ask a Question)

The Microchip Website (Ask a Question)

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service (Ask a Question)

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support (Ask a Question)

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature (Ask a Question)

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice (Ask a Question)

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks (Ask a Question)

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-

ISBN: 978-1-6683-2123-2

Quality Management System (Ask a Question)

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820